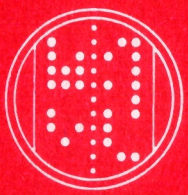
РЕШЕНИЕ МАССОВЫХ ГЕОДЕЗИЧЕСКИХ ЗАДАЧ НА МИКРОЭВМ

СПРАВОЧНОЕ ПОСОБИЕ



РЕШЕНИЕ МАССОВЫХ ГЕОДЕЗИЧЕСКИХ ЗАДАЧ НА МИКРОЭВМ

СПРАВОЧНОЕ ПОСОБИЕ

МОСКВА "НЕДРА" 1991

ББК 26.1
Р 47
УДК 528.1

Авторы: *М. И. Коробочкин*, д-р техн. наук, *В. С. Бережнов*, канд. техн. наук, *Н. С. Зайцева*, канд. техн. наук, *В. С. Красницкий*, канд. техн. наук

Решение массовых геодезических задач на микроЭВМ: Р 47 Справочное пособие/М. И. Коробочкин, В. С. Бережнов, Н. С. Зайцева, В. С. Красницкий.— М.: Недра, 1991.— 144 с.: ил.

ISBN 5-247-00660-7

Рассмотрены рациональные методы решения массовых геодезических задач на микроЭВМ с необходимой точностью и надежностью. Значительное внимание уделено построению эффективных программ для автоматического решения задач, выбору алгоритмов и организации данных. Рассмотрены также рациональные приемы программирования на языке Бейсик.

Для специалистов предприятий, учреждений и организаций всех ведомств, выполняющих геодезические вычисления. Может быть полезно студентам вузов и учащимся техникумов.

Р $\frac{1802020000-157}{043(01)-91}$ 21—91 **ББК 26.1**

ISBN 5-247-00660-7

ПРЕДИСЛОВИЕ

Важнейшим фактором повышения производительности труда инженера-геодезиста является умение эффективно использовать в своей работе средства вычислительной техники. Наиболее популярными из таких средств в настоящее время являются персональные ЭВМ (ПЭВМ). Наиболее дорогие ПЭВМ по своим основным параметрам (емкости запоминающих устройств и быстродействию) не уступают большим вычислительным машинам. Параметры ПЭВМ, рассчитанных на массового потребителя, вполне достаточны для решения большинства задач инженерной геодезии.

Эффективность использования средств вычислительной техники зависит, по крайней мере, от двух факторов: уровня подготовки лица, работающего с ЭВМ (пользователя), и качества программного обеспечения.

Программное обеспечение — это совокупность программ, обеспечивающих пользователю решение его задач на ЭВМ. Программное обеспечение принято делить на системное и прикладное. Системное программное обеспечение автоматически и по командам пользователя управляет всеми программными и аппаратными компонентами ЭВМ. Прикладное программное обеспечение состоит из программ, предназначенных для решения конкретных задач из той или иной производственной, научной, управленческой, социальной и других сфер деятельности.

Если системное программное обеспечение предоставляет пользователю объем услуг, вполне достаточный для ввода, трансляции, отладки и реализации его программ, то прикладными программами персональные ЭВМ обеспечены пока слабо. Инженерная геодезия относится к той группе дисциплин, в которой этот недостаток ощущается особенно остро. Программы решения многих задач, встречающихся в геодезической практике, часто приходится писать самим пользователям. Поэтому проблема овладения приемами программирования является для инженера-геодезиста достаточно актуальной.

Для овладения профессией программиста, умеющего создавать так называемый программный продукт, как и для овладения любой другой профессией, необходима длительная учеба, длительная практика и определенные способности. Но научиться писать простые и, вместе с тем, необходимые в профессиональной деятельности программы может любой грамотный человек. Авторы надеются, что данная книга в какой-то степени поможет читателю получить навыки, необходимые для подготовки таких программ, их отладки и реализации на ПЭВМ.

Книга содержит описание исходной версии языка Бейсик, реализуемой на ПЭВМ с минимальной оперативной памятью и минимальным набором внешних устройств, а также расширения возможностей языка в распространенных версиях. Описание языка сопровождается изложением методики программирования.

В книге приведены алгоритмы решения ряда распространенных задач инженерной геодезии, диалог, сопровождающий работу, с программами решения этих задач, форма вывода результатов и тестовые примеры для проверки работоспособности программ. Этот материал может оказаться полезным при самостоятельной разработке и отладке аналогичных программ.

Все описанные в книге программы разработаны и внедрены авторами, а также доцентом кафедры геодезии Московского института инженеров землеустройства Е. Г. Парамоновой и старшим преподавателем кафедры высшей геодезии этого же института В. Ю. Давыдовым.

1. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ БЕЙСИК

1.1. ЭВМ. Память ЭВМ. Программа

Электронная вычислительная машина (ЭВМ) — это автомат с программным управлением, предназначенный для хранения, обработки и выдачи данных. Это значит, что для решения той или иной задачи в машину прежде всего необходимо ввести программу (рис. 1) и инициировать ее выполнение. В процессе выполнения программы могут понадобиться исходные данные, которые также надо ввести. Ввод осуществляется либо вручную с клавиатуры, либо автоматически, например, с магнитного диска или магнитной ленты. В последнем случае вводимые данные должны быть предварительно записаны на диске или ленте вручную либо автоматически с помощью другой, ранее выполнявшейся программы.

В соответствии с программой происходит обработка данных (вычисление, отбор по указанным признакам, сортировка по указанным правилам и т. п.). Результаты обработки подлежат выводу на экран дисплея, если их используют немедленно, или на печать, если их необходимо сохранить в качестве документа или использовать в дальнейшем. Если полученные результаты предполагается использовать в качестве исходных данных при последующем выполнении этой или другой программы, то они выводятся на внешние запоминающие устройства (магнитные диски или ленты).

Для хранения программы, обрабатываемых данных и результатов ЭВМ снабжается памятью. Принято различать два вида памяти: оперативную и внешнюю. Оперативная память позволяет хранить данные и программу в процессе решения задачи, грубо говоря, пока машина включена. Эта память обладает очень высоким быстродействием и меньшей, по сравнению с внешней памятью, емкостью, которая колеблется от 64 000 байт (у наиболее дешевых персональных ЭВМ) до нескольких миллионов байт (у развитых моделей ЭВМ того же типа). Байт — это единица информации. Один байт позволяет хранить один символ (букву, цифру, знак и т. п.).

Внешняя или долговременная память позволяет хранить данные практически неограниченное время. По сравнению с оперативной памятью она имеет значительно меньшее быстродействие и во много раз большую емкость. Распространенными видами внешней памяти у персональных ЭВМ являются гибкие магнитные диски (ГМД), магнитные ленты и жесткие магнитные диски.

В определенном смысле можно утверждать, что емкость внешней памяти неограничена, так как всегда имеется возможность хранить данные на нескольких дисках или кассетах с магнитной лентой.

Программа — это последовательность указаний, выполняя которые машина осуществляет обработку данных. Непосредственно на устрой-

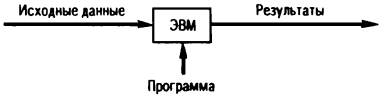


Рис. 1. Схема движения информации при решении задачи на ЭВМ

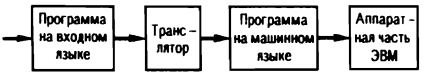


Рис. 2. Схема реализации программы в ЭВМ

ства машины воздействует программа, состоящая из так называемых «машинных команд». Подготовка такой программы осуществляется автоматически с помощью специальных программ, называемых трансляторами. Исходной информацией для транслятора является программа, написанная на одном из входных языков (Фортран, РЛ/1, Бейсик, Паскаль и др.). Схематически прохождение программы в ЭВМ изображено на рис. 2.

Транслятор либо вводится в оперативную память ЭВМ перед вводом программы, либо реализуется в виде интегральной схемы и является составной частью оборудования машины. В последнем случае ЭВМ отличается большей мобильностью и меньшей универсальностью.

Программа на входном языке представляет из себя последовательность операторов. Оператор — это указание выполнить то или иное действие: вычислить, прочитать, вывести на экран и т. д. В процессе трансляции каждый оператор преобразуется в одну или несколько машинных команд. Кроме операторов программа на входном языке содержит некоторую информацию, необходимую транслятору, например, сведения о характере данных, подлежащих обработке при выполнении программы. Объем этой вспомогательной информации в разных языках различен. В языке Бейсик он сведен к минимуму.

По принципу действия трансляторы делятся на интерпретирующие и компилирующие. В трансляторах интерпретирующего типа (интерпретаторах) очередной оператор после его перевода в последовательность машинных команд немедленно исполняется. В трансляторах компилирующего типа (компиляторах) программа на входном языке (исходный модуль) вначале полностью переводится на язык машинных команд (преобразуется в объектный модуль), затем с помощью редактора связей подготавливается к выполнению (преобразуется в загрузочный модуль). Загрузочный модуль представляет собой самостоятельную программную единицу, которая не требует для своего выполнения вспомогательных программ. Он может быть записан на магнитный диск или ленту и вызываться оттуда для исполнения.

Выполнение программы, представленной в виде загрузочного модуля, происходит быстрее, чем выполнение такой же программы на входном языке с помощью транслятора интерпретирующего типа. С другой стороны, создание загрузочного модуля требует бо́льшего количества программных средств и предъявляет более высокие требования к оперативной памяти ЭВМ.

1.2. Данные и правила их представления в программах на языке Бейсик

В исходной версии языка существует три формы представления данных: числовые константы (числа), числовые переменные и текс-

товые константы. В более развитых версиях языка существует также текстовая переменная.

Числовые константы. В языке Бейсик существует две формы представления чисел: естественная и показательная (экспоненциальная). Естественная форма отличается от общепринятой тем, что дробная часть числа отделяется от целой части не запятой, а точкой, например, число 35,475 по правилам языка Бейсик будет записано: 35.475. Нуль в целой части числа можно не писать, например, 0.542 или .542, —0.03085 или —.03085 и т. п.

В показательной форме число имеет вид MEP, где M — мантисса (любое число в естественной форме), P — порядок (целое число), например: 2.54E—5, —.0.73E12, .456E—3. Для перевода числа из показательной формы в естественную необходимо мантиссу умножить на десять в степени, равной порядку. Фактические значения приведенных выше чисел: 0,0000254, —730000000000, 0.000456.

При вводе чисел в машину можно пользоваться любой формой их представления. При этом число может иметь любую длину, а в экспоненциальной форме мантисса может иметь любой вид, например, 0.00125, 125E—5, 0.125E—2, .125E—2, 1.25E—3, 35000000000000, 0.00000000030765. В процессе ввода число переводится в показательную форму, мантисса округляется до семи значащих цифр и с этой точностью число хранится в памяти ЭВМ и используется для вычислений.

В исходной версии языка Бейсик вывод чисел осуществляется по следующему правилу: если для размещения значащих цифр числа требуется не более семи позиций, то число выводится в естественной форме, в противном случае используется показательная форма. При этом мантисса нормализуется, т. е. принимает значение, удовлетворяющее условию $0{,}1 \leqslant M < 1$ (табл. 1). В других версиях языка встречаются отклонения от приведенного выше правила, например, в показательной форме мантисса удовлетворяет условию $1 \leqslant M < 10$.

Числовая переменная — это данное, которое при выполнении программы может принимать различные числовые значения. В языке Бейсик, как и в большинстве других языков, существует два вида переменных: простые переменные и переменные с индексами.

В языке Бейсик простая переменная может обозначаться либо одной буквой, либо одной буквой и одной цифрой, причем, буква должна стоять на первом месте. Примеры переменных: X, A, Z, C4, D3, AØ. Примеры обозначения простых переменных, недопустимые в языке Бейсик: AP (две буквы), C14 (две цифры), 5C (цифра на первом месте).

Переменная с индексом является элементом массива. Массив — это совокупность однородных переменных, имеющих общее имя (имя массива). Переменные, принадлежащие одному массиву, отличаются друг от друга значением индекса, которое указывается в скобках после имени массива. Например, X(2), X(5). В языке Бейсик значение индекса может быть указано с помощью числа, простой переменной, арифметического выражения или другой переменной с индексом, например A(4), A(P), A(I+4), A(B/2), A(C(1)) и т. п.

Имя массива пишется по тем же правилам, что и имя простой

Таблица 1
Примеры отображения чисел в языке Бейсик

Возможная форма числа при вводе	Обязательная форма числа при выводе
0.1	.1
−0.543	−.543
2	2
002	2
0	0
000	0
−9.9 E−3	−.0099
0.570000	.57
350000000	.35 E 9
0.0000000054206	.54206 E−8
12000000978	.12 E 11
12.3456789	.1234568E2

переменной. Так, например, в языке Бейсик можно использовать переменные Z(4), P1(7), X4(K+1) и нельзя использовать переменные XA(4), P15(I), 2Q(L+1) и т. п.

В языке Бейсик допустимо использование переменных с одним или двумя индексами. В последнем случае индексы отделяются друг от друга запятой, например: X(3,2), R3(3,7), Z(K,K+2), C3(L,N).

Индексы в языке Бейсик могут принимать значения от 0 до 255, так, например, переменная A(300) недопустима.

Текстовая константа — это последовательность символов, заключенная в кавычки. Часто это какой-либо текст, содержащий информацию о вводимых или выводимых данных, например: «ВВЕСТИ КООРДИНАТЫ ИСХОДНОГО ПУНКТА», «ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ» и т. п.

Текстовые переменные. Понятие текстовой переменной отсутствует в исходной версии языка. Однако это понятие введено в состав большинства более поздних версий.

Текстовые переменные записываются по тем же правилам, что и числовые, с добавлением в конце имени символа ⊗. Например, A⊗, X⊗, D4⊗. Существует понятие текстовой переменной с индексом, например: F⊗(5), R3⊗(2), Z⊗(1), N⊗(Q+2), W⊗(K(1)).

В процессе выполнения программы текстовые переменные могут принимать значения последовательностей символов. Например, переменная D⊗ может принимать значение слова «ФЕВРАЛЬ». При выводе значения переменной D⊗ на экране дисплея появится это слово.

1.3. Структура программы на языке Бейсик

Все операторы языка Бейсик начинаются с ключевого слова. Например, INPUT — ввести, PRINT — отпечатать, вывести, LET — допустим, вычислим, END — конец и т. п. Исключением является слово LET, которое во многих версиях языка не является обязательным.

Операторы размещаются в строках. Каждая строка имеет номер и может содержать один или несколько операторов. Если одна строка содержит несколько операторов, то они в исходной версии языка

разделяются двоеточиями. В более поздних версиях точка с запятой или знак \. Использование разных разделителей в различных версиях языка часто является единственным препятствием для реализации одной и той же программы в разных организациях. Поэтому лучше размещать операторы по одному в каждой строке.

Строки нумеруются в порядке возрастания в той последовательности, в которой должны выполняться содержащиеся в них операторы. Для нумерации строк используются целые числа. Пример программы:

```
1 INPUT A, B, C, X
2 LET Z=A+B
3 LET D=C/X: LET R=Z+D
4 PRINT D, R, Z
5 END
```

В строке 1 находится оператор ввода. С его помощью в машину вводятся числовые значения переменных A, B, C и X. В строках 2 и 3 находятся три оператора присваивания. С их помощью вычисляются числовые значения переменных Z, D и R. В строке 4 расположен оператор вывода, с помощью которого на экран дисплея выводятся числовые значения переменных D, R и Z.

Строки не обязательно нумеровать натуральным рядом чисел, как это сделано в приведенном выше примере. Более удобной является нумерация с некоторым шагом, например:

```
10 INPUT A, B, C, X
20 LET Z=A+B
30 LET D=C/X: LET R=Z+D
40 PRINT D, R, Z
50 END
```

Такая нумерация позволяет легко поместить в программу недостающий оператор. Например, если возникает необходимость дополнить приведенную выше программу операторами LET Q=D/R и PRINT Q, то их можно будет записать ниже, например, под номерами 34 и 37. Независимо от последовательности ввода операторы выполняются в порядке возрастания номеров строк, в которых они размещены.

1.4. Алфавит языка Бейсик. Операторы вывода, ввода, присваивания

Алфавит языка Бейсик содержит следующие символы:

заглавные буквы латинского алфавита (от A до Z);

цифры от Ø до 9, цифру «нуль» перечеркивают, чтобы не спутать с буквой O;

специальные символы: + − / * :; , . ∧ ( ) " < > = ;

пробел (при записи программы на бланке пропускается одна позиция, например,

"	Р	А	З	Д	Е	Л		5	"

,

при записи на неразграфленной бумаге пишется символ ␣ ', например, «РАЗДЕЛ␣5», при вводе в машину нажимается клавиша «интервал»).

При записи программы можно использовать только символы, входящие в состав алфавита языка. Исключение составляют лишь текстовые константы, значения текстовых переменных и оператора REM, в которых можно использовать любые символы, имеющиеся на устройствах ввода и вывода данной ЭВМ.

Оператор вывода. Общий вид (формат) оператора вывода:

PRINT список вывода.

В стандартной версии языка Бейсик оператор служит для вывода данных на экран дисплея. Список вывода состоит из одного или нескольких элементов, разделенных запятыми или точками с запятой. Элементами списка вывода могут быть переменные, числа, арифметические выражения или текстовые константы, например:

```
110 PRINT X
120 PRINT 2.7
130 PRINT A, B2, C
140 PRINT (X-Z)/2
150 PRINT Z; P-R; D/2+K
160 PRINT "СПИСОК ГРУППЫ"
170 PRINT "X=" X
180 PRINT "R=" (A+B)/(C-D)
190 PRINT L "␣" M⊗; G"␣ГОДА, ␣" F⊗
200 PRINT "РАССТОЯНИЕ="S"␣КМ"
```

В случае числа, переменной или арифметического выражения оператор PRINT выводит их значения. Текстовая константа выводится в том виде, в каком она записана в программе, но без кавычек.

Если до выполнения приведенной выше группы операторов числовые переменные будут иметь значения: X=5, A=7.4, B2=−3.2, C=7, S=53.2, Z=12.8, P=20, R=25.5, D=16, K=4, L=12, B=10.6, G=1988, а значения текстовых переменных — M⊗=ФЕВРАЛЯ, F⊗= =ПЯТНИЦА, то в результате выполнения операторов 110—190 на экране появится следующий текст:

```
 5
 2.7
 7.4          −3.2          7
−3.9
 12.8 −5.5  12
 СПИСОК ГРУППЫ
 X=5
 R=−2
 12 ФЕВРАЛЯ 1988 ГОДА, ПЯТНИЦА
 РАССТОЯНИЕ=53.2 КМ
```

Если в списке вывода оператора PRINT перед текстовой константой или после нее располагается переменная или выражение, как в операторах 170—200, то разделитель (точку с запятой) между ними можно не ставить. Независимо от того, стоит после кавычек точка

с запятой или нет, значение следующего данного выводится непосредственно после значения предшествующей ему текстовой константы. При выводе положительных чисел появляется пробел на позиции знака (см. результаты выполнения операторов 170 и 200).

При подготовке операторов вывода не следует забывать о пробелах, необходимых перед текстом, который выводится после числовых значений (см. операторы 190 и 200).

Использование разделителей (запятой или точки с запятой) позволяет по-разному располагать результаты на экране. При выводе строка экрана делится условно на 5 зон по 14 позиций. Если элементы списка вывода разделены запятыми, то значение очередного элемента выводится в следующей зоне. Например, при выполнении оператора 130 знак «—» в числе —3.2 появится на 15-й позиции, а цифра 7 — на 30-й позиции (подразумевается, что 29-я позиция занята знаком «+», который в языке Бейсик не выводится).

Если между элементами списка вывода поместить подряд две запятые, то значение очередного элемента списка выводится через одну зону после значения предыдущего элемента.

Если между элементами списка вывода стоит точка с запятой, то значение очередного элемента выводится через одну позицию после предыдущего элемента. Например, при выполнении оператора 150 знак "␣" в числе —5.5 появится на 7-й позиции, а цифра 1 в числе 12 — на 13-й позиции. (При проверке этого утверждения не забудьте учесть позиции, отводимые под знаки чисел).

Если в конце списка вывода стоит запятая или точка с запятой, то следующий оператор PRINT осуществляет вывод в ту же строку, что и предыдущий. Если в конце списка вывода разделитель отсутствует, то следующий оператор PRINT осуществляет вывод с новой строки. Если количество элементов списка вывода таково, что их значения не помещаются на одной строке, то автоматически происходит переход на следующую строку. Оператор PRINT, не содержащий списка вывода, осуществляет перевод на новую строку.

Рассмотрим пример:

```
 10 LET A=1:LET B=2:LET C=3
 20 LET D=4:LET E=5:LET F=6
 30 PRINT "␣␣␣␣ПРИМЕР 1:"
 40 PRINT
 50 PRINT "A", "B", "C", "D"
 60 PRINT A, B,
 70 PRINT C, D
 80 PRINT
 90 PRINT "E=" E, "F=" F
100 PRINT
110 PRINT "ПРИМЕР␣2:", A; B; C; D; E; F
120 PRINT
130 PRINT "ПРИМЕР␣3:", A, B, C, D, E,, F
```

Здесь операторы LET, размещенные в строках 10 и 20, присваивают числовые значения переменным A, B, C, D, E, F.

В результате выполнения программы будет выведен следующий текст (курсивом приведены пояснения):

```
␣␣␣␣␣ПРИМЕР 1: (␣ обозначает пустую позицию)
(пустая строка)
A␣␣␣␣␣␣␣␣␣␣B␣␣␣␣␣␣␣␣␣C␣␣␣␣␣␣␣␣␣␣␣␣D
 1                2                3                4
(пустая строка)
E= —␣5·          F=␣6
(пустая строка)
ПРИМЕР␣2:␣␣␣␣␣␣1␣␣2␣␣3␣␣4␣␣5␣␣6
(пустая строка)
ПРИМЕР␣3:        1          .     2                3  4
5                                  6
```

Вывод в заданном формате. Вывод числовых данных с помощью оператора PRINT осуществляется по правилам, описанным выше. В некоторых версиях языка существует директива USING, которая позволяет задавать формат вывода чисел. Формат задается с помощью последовательности символов #, между которыми можно помещать точку. Например, формат # # #, # # предписывает разместить число, включая знак и десятичную точку, на шести позициях, из которых две последние позиции отводятся под дробную часть. Если длина дробной части числа превосходит отведенное для нее количество позиций, то число округляется. Если не хватает позиций, предусмотренных для размещения целой части, то число выводится без ошибки, но перед ним помещается символ %, свидетельствующий о том, что полученный результат превосходит предусмотренный для него формат. Формат заключают в кавычки и помещают его перед выводимым данным, используя в качестве разделителя точку с запятой.

Примеры использования формата:

Пусть A=35.2471. Тогда в результате выполнения оператора

```
PRINT USING "# # #.# #"; A
```

будет выведено число 35.25; в результате выполнения оператора PRINT USING "РАССТОЯНИЕ="; "# # #.# #"; A"␣КМ" будет выведен текст: РАССТОЯНИЕ=35.25 КМ; в результате выполнения оператора PRINT USING "#.# #"; A на экране появится текст: %35.25.

При выводе нескольких чисел перед ними необходимо поместить список форматов, разделенных пробелами, например,

```
PRINT USING "# #.# #  #.# # #"; A; B
```

В некоторых версиях языка допустимо использование одного формата перед несколькими переменными, которые в этом случае будут выводиться в одном и том же формате, например,

```
PRINT USING "# # # #.# #"; X; Y; Z
```

Отрицательные числа выводятся со знаком "—", положительные — без знака. Если необходимо положительные числа выводить со знаком

"+", то этот знак помещают в начале формата. Например, если A= =45.24, B= −15.45, C=2.392, то в результате выполнения оператора

PRINT USING "+ # # #. # + # # # # #.# + # # # #.# #"; A; B; C

будут выведены числа +45.2 −15.5 +2.39.

Если необходимо свободные позиции слева от числа заполнить символами *, то в левой части формата помещают **. Одновременно эти два символа играют ту же роль, что и # #, т. е. указывают дополнительное количество позиций под целую часть числа. Например, если A=1.5, B=256.35, то при выполнении оператора

PRINT USING "** # #.# # ** # #.# #"; A; B

будет выведен текст ***1.50*256.35.

Если в начале формата поместить ⊗⊗ (знаки денежной единицы), то один из указанных символов будет выведен перед числом. Например, если A=23.682, то в результате выполнения оператора

PRINT USING "⊗⊗ # # #.# #"; A

будет выведен текст ⊗ 23.69.

В различных версиях языка существуют некоторые отступления от перечисленных правил работы с директивой USING. Все эти отступления легко выяснить опытным путем.

Вывод на печать. Во многих версиях языка Бейсик определено понятие файла, что дает возможность организовать вывод данных на печатающее устройство. Для этого в начальной части программы необходимо с помощью оператора OPEN открыть для вывода файл с физическим именем "LP:" и произвольно выбранным номером. Структура оператора OPEN для организации вывода:

OPEN физическое имя FOR OUTPUT AS FILE # номер .

Например:

10 OPEN "LP:" FOR OUTPUT AS FILE #1 .

В операторах PRINT, предназначенных для вывода на печать, в качестве первого элемента списка вывода необходимо помещать номер, указанный в операторе OPEN. Например:

140 PRINT #1, A; B; C
200 PRINT #1, "ПЛОЩАДЬ=" S"␣КВ.М" .

Номер отделяется запятой от остальных элементов списка вывода.

В конце программы после всех операторов PRINT, предназначенных для вывода на печать, открытый ранее файл необходимо закрыть с помощью оператора «CLOSE # номер», например, CLOSE #1. В противном случае подлежащие выводу данные могут остаться в буфере печатающего устройства.

Можно организовать в программе вывод на печать или экран по желанию пользователя, например:

```
10 PRINT "ВЫВОД НА ПЕЧАТЬ ИЛИ ЭКРАН?"
20 PRINT "НА ПЕЧАТЬ — ВВЕДИТЕ LP:"
30 PRINT "НА ЭКРАН — ВВЕДИТЕ TT:";
```

```
40 INPUT P⊗
50 OPEN P⊗FOR OUTPUT AS FILE #1
.....
100 PRINT #1, ...
.....
300 CLOSE #1
.....
```

Здесь при выполнении оператора 40 символьная переменная P⊗ принимает значение физического имени печатающего устройства (LP:) или экрана (TT:), а оператор 50 открывает соответствующий файл и присваивает ему номер 1.

Оператор ввода. Формат оператора ввода: INPUT список ввода. Список ввода состоит из одной или нескольких переменных, разделенных запятыми, например:

```
110 INPUT X
120 INPUT A, B1, Y(3)
130 INPUT Z(1)
140 INPUT Q⊗
150 INPUT Z(5): D4⊗, F⊗(7)
```

Оператор предназначен для присвоения перечисленным в списке ввода переменных числовых или текстовых значений (текстовые значения допустимы не во всех версиях языка). Предварительно эти значения должны быть помещены на экран.

При выполнении оператора INPUT машина выдает на экран знак вопроса (?) и останавливается. В ответ необходимо поместить на экране разделенные запятыми значения вводимых данных в количестве, равном числу переменных в списке ввода, и нажать клавишу ВК. В результате машина присвоит переменным из списка ввода значения данных, которые были занесены на экран.

Примеры:

1. При выполнении оператора 110 INPUT X на экране появится знак вопроса. Если после этого занести на экран число 3.4 (? 3.4) и нажать клавишу ВК, то переменная X получит значение 3.4.

2. Если при выполнении оператора 120 INPUT A, B1, Y(3) в ответ на знак вопроса набрать числа 7.15, —.35E10,4, то переменная A получит значение 7.15, переменная B1 — значение -0.35×10^{10}, переменная Y(3) — значение 4.

Если количество помещенных на экран чисел больше или меньше, чем количество переменных в списке ввода, то в исходной версии языка Бейсик машина выдает сообщение об ошибке и знак вопроса. Ввод нужно повторить. Например, если при выполнении оператора 40 INPUT A, B, C, P ввести ?5, 6.3, 7, 1.2, 4.5, то машина выведет сообщение:

```
ОШИБКА 122 СТРОКЕ 40
?
```

В ответ необходимо ввести четыре числа, а не пять, как это было ошибочно сделано.

Описанный контроль ввода представляется достаточно эффективным. К сожалению во многих более поздних версиях этому вопросу уделено меньше внимания. В этих версиях в случае, когда количество введенных на экран чисел превосходит количество элементов списка ввода, «лишние» числа не читаются и выполнение оператора считается законченным без ошибок. Если количество чисел, помещенных на экран, недостаточно, то начальные элементы списка получают значения и на экране вновь загорается знак вопроса. Например, если при выполнении оператора 40 INPUT A, B, C, P ввести ? 23, 14, то переменные A и B получат соответственно значения 23 и 14, а на экране мы будем наблюдать изображение:

```
? 23, 14
?
```

После этого ввод необходимо продолжить.

В тех версиях языка, которые допускают работу с текстовыми переменными, необходимо иметь в виду, что запятая является разделителем между вводимыми значениями и поэтому ее нельзя использовать в качестве элемента вводимого текста. Например, при выполнении оператора 140 INPUT Q⊗ нельзя ввести значение ПЕТРОВ, СИДОРОВ. Переменная Q⊗ получит значение ПЕТРОВ. Но можно ввести значение ПЕТРОВ СИДОРОВ без запятой.

Перед оператором ввода полезно помещать оператор PRINT с информацией о данных, подлежащих вводу. Это обеспечит ввод в режиме диалога. Например, в некоторой программе необходимо осуществить ввод координат трех точек: X1, Y1, X2, Y2, X3, Y3 с помощью трех операторов ввода. Сделать это можно следующим образом:

```
10 PRINT "ВВЕСТИ КООРДИНАТЫ (X, Y):"
20 PRINT "ТОЧКИ 1";
30 INPUT X1, Y1
40 PRINT "ТОЧКИ 2";
50 INPUT X2, Y2
60 PRINT "ТОЧКИ 3";
70 INPUT X3, Y3
```

Пусть координаты имеют следующие значения для точек:

1 — 5124.7 3425.0;
2 — 1560.5 2540.8;
3 — 7440.8 6666.6.

В этом случае диалог на экране будет иметь следующий вид:

```
ВВЕСТИ КООРДИНАТЫ (X, Y):
ТОЧКИ 1? 5124.7, 3425.0 ВК
ТОЧКИ 2? 1560.5, 2540.8 ВК
ТОЧКИ 3? 7440.8. 6666.6 ВК
```

Обратите внимание на точку с запятой в конце операторов 20, 40, 60. Она обеспечивает вывод операторами 30, 50, 70 вопросительных знаков в ту же строку, в которую предшествующие им операторы 20, 40, 60 выводят помещенные в них запросы. При отсутствии точки

с запятой вопросительный знак выводится в следующую строку. Например, после выполнения операторов

```
20 PRINT "ТОЧКИ 1"
30 INPUT X1, Y1
```

текст на экране будет иметь следующий вид:

```
ТОЧКИ 1
?
```

При составлении программы для персональных ЭВМ следует уделять организации диалога большое внимание. При выполнении программы машина должна выдавать вразумительные запросы, а результаты должны выводиться в понятной и удобочитаемой форме.

Оператор присваивания. Формат оператора присваивания:

LET переменная = арифметическое выражение.

Примеры операторов присваивания:

```
10 LET A=3
20 LET C=A+1
30 LET B=C
40 LET D=2*A+C-B/2
```

Оператор LET выполняется в следующей последовательности: вначале вычисляется арифметическое выражение, расположенное справа от знака "=", затем, полученное числовое значение присваивается переменной, расположенной слева от знака "=".

В приведенном выше примере переменная A получает значение 3, затем переменная C получает значение 4, затем переменная B получает значение 4 и, наконец, переменная D получает значение 8.

Знак "=" в операторе LET обозначает не "равно", а "присвоить". Оператор LET может содержать по обе стороны от знака "присвоить" имя одной и той же переменной, например:

```
50 LET I=I+1
60 LET P=P*3
70 LET R=R*2+R/4
```

и т. п.

В операторе 50 переменная I получает значение на единицу больше того, которое имела, в операторе 60 переменная P получает значение втрое больше предыдущего, в операторе 70 новое значение R вычисляется по более сложной формуле, но тоже с использованием предыдущего значения R.

В большинстве версий языка Бейсик слово LET в операторе присваивания не является обязательным.

Правой частью оператора присваивания в общем случае является арифметическое выражение. Рассмотрим те правила записи арифметических выражений, которые не являются очевидными.

В арифметических выражениях используются специальные знаки операций (табл. 2). Арифметическое выражение

$$R+\frac{A}{B}-C^{2.1}$$

в программе будет записано следующим образом:

Таблица 2

Арифметические операции, допустимые в языке Бейсик

Символ языка Бейсик	Пример записи	Комментарий	Приоритет операций
$\wedge$	A$\wedge$B	Возведение в степень (A^B)	Первый
*	A*B	Умножение	Второй
/	A/B	Деление	Второй
+	A+B	Сложение	Третий
−	A−B	Вычитание	Третий

R+A/B−C$\wedge$2.1 .

Арифметическое выражение с использованием скобок

$$\{[(A+B)\cdot 2{,}1+C]\cdot 3{,}5\}+\frac{D}{R}$$

в машинной записи будет выглядеть так:

(((A+B)*2.1+C)*3.5)+D/*R* ,

т. е. алфавит языка Бейсик располагает только круглыми скобками. При выполнении программы арифметическое выражение вычисляется в последовательности, соответствующей приоритету операций (см. табл. 2) и с учетом скобок.

При записи арифметических выражений, представляющих собою дроби с многочленами в числителе и (или) знаменателе, следует заключить многочлены в скобки. Например, выражение $\frac{A+B}{C+D}$ в программе должно иметь вид (A+B)/(C+D). Выражение $A+B/C+D$ будет вычисляться, как $A+\frac{B}{C}+D$.

Скобки могут понадобиться и в том случае, когда в знаменателе находится произведение. Например, дробь $\frac{A\cdot B}{C\cdot D}$ следует записать в виде A*B/(C*D) или без скобок в виде A*B/C/D. Выбирать формулу следует таким образом, чтобы уменьшить погрешность вычислений.

Возведение в степень A$\wedge$B в языке Бейсик осуществляется по формуле $e^{B\ln A}$. Поскольку не существует логарифмов отрицательных чисел и нуля, выполнение операций A$\wedge$B при $A\leqslant 0$ недопустимо даже в случае, когда B целое число. Например, нельзя программировать операцию A$\wedge$3, если A может оказаться отрицательным числом или нулем. В указанной ситуации возведение в степень следует заменить многократным умножением, например, A^3 программировать, как A*A*A.

Аналогично следует поступать, когда в целую степень возводится арифметическое выражение, которое может оказаться отрицательным числом или нулем, например, при возведении в квадрат приращения координат некоторой точки: $(X2-X1)^2$. При этом лучше не перемножать непосредственно арифметическое выражение (X2−X1)* *(X2−X1), а воспользоваться вспомогательной переменной, например, вначале вычислить D=X2−X1, а затем перемножить D*D. Это

Таблица 3
Встроенные функции языка Бейсик

Обозначение функции в программе	Вычисление функции
SIN (X)	$\sin x$
COS (X)	$\cos x$
ATN (X)	$\operatorname{arctg} x$
SQR (X)	$\sqrt{x}$
EXP (X)	e^x
LOG (X)	$\ln x$
ABS (X)	$\lvert x \rvert$
INT (X)	Образуется целое число, ближайшее к x, но не превосходящее его
SGN (X)	Равна -1, если $X<0$; 0, если $X=0$; $+1$, если $X>0$
RND (X)	Вырабатывается случайное число в диапазоне от 0 до 1 (распределение равномерное)

избавит машину от необходимости дважды вычислять одно и то же выражение.

В арифметических выражениях языка Бейсик можно использовать функции, перечисленные в табл. 3.

Некоторые особенности использования функций в программе на языке Бейсик:

1. При использовании функций обязательно заключать аргумент в скобки. Нельзя, например, писать SIN X, SQR 5; необходимо писать SIN (X), SQR (5) и т. п.

2. Аргументом функции может быть число, переменная или выражение, например SIN (A), SIN (R+1.57), SQR (A*A+B*B).

3. В соответствии с описанием действие функции INT (X) неэквивалентно округлению и эквивалентно выделению целой части только при положительном аргументе, например: INT (6.8) =6, INT (6.3) = =6, но INT (−6.8) = −7, INT (−6.3) = −7.

Эту функцию можно использовать для округления до целых с помощью выражения INT (X+0.5), где X — округляемое число, переменная или выражение. Если данное нужно округлить до десятых, сотых и т. п. (или до десятков, сотен и т. д.), то его нужно предварительно увеличить (или уменьшить) в десять, сто раз и т. д., округлить до целых, а затем уменьшить (или увеличить) во столько же раз. Например, для округления значения переменной X до сотых можно воспользоваться оператором

```
LET X1 = INT (X*100 + 0.5) / 100,
```

где X1 — округленное значение переменной X. Оператор

```
LET X1 = INT (X/10 + 0.5) * 10
```

округляет величину X до десятков.

4. Случайное число, вырабатываемое функцией RND (X), не зависит от аргумента, поэтому можно всегда писать RND (Ø).

5. Для вычисления корня любой степени из положительного числа можно воспользоваться соотношением

$$\sqrt[n]{x} = x^{1/N},$$

правой части которого в языке Бейсик соответствует выражение: X∧(1/N).

6. Аргументы функций SIN, COS и результат вычисления функции ATN выражаются в радианной мере. Поскольку в геодезической практике углы обычно измеряются в градусах, минутах и секундах, то при использовании перечисленных функций требуется перевод углов из градусной меры в радианную и наоборот. Например, если положительный угол равен G градусов, M минут, S секунд, то синус этого угла можно вычислять с помощью выражения

```
SIN (((S/60+M)/60+G)*3.141593/180).
```

Здесь вначале секунды, минуты и градусы преобразуются в градусы с десятичной дробной частью:

$$\alpha_{\text{дес}} = (S/60 + M)/60 + G,$$

а затем $\alpha_{\text{дес}}$ умножается на коэффициент перевода угла из градусов в радианы: 3.141593/180.

Чтобы сделать запись менее громоздкой, можно воспользоваться вспомогательной переменной:

```
100 LET R=((S/60+M)/60+G)*3.141593/180
110 LET P=SIN (R)
```

Вспомогательная переменная удобна и в тех случаях, когда необходимо вычислить различные функции одного и того же угла, например:

```
100 LET R=((S/60+M)/60+G)*3.141593/180
110 LET X=SIN (R)
120 LET Y=COS (R)
```

Если известно значение тангенса угла (T), и необходимо определить угол в градусах (G), минутах (M) и секундах (S), то в случае положительного угла в диапазоне от 0 до 90 это можно сделать с помощью следующей группы операторов:

```
100 LET R=ATN (T)  (R — угол в радианах)
110 LET G1=R*180/3.141593  (G1 — угол с десятичной дробной частью)
120 LET G=INT (G1)  (G — градусы)
130 LET M1=(G1-G)*60  (M1 — минуты с десятичной дробной частью)
140 LET M=INT (M1)  (M — минуты)
150 LET S=INT((M1-M)*60+0.5)  (S — секунды, округленные до целых).
```

Недостатком приведенной выше программы является возможность получения углов, в которых число секунд равно 60, например, 27°16′60″ вместо 27°17′; 48°59′60″ вместо 49° и т. п. Способ корректировки по-

добных результатов будет рассмотрен при изучении условных операторов.

Приведенная выше программа работоспособна лишь в случае положительных углов. Поэтому в задачах, где возможно появление отрицательного угла, нужно изменить оператор 110:

```
110 LET G1=ABS (R*180/3.141593),
```

а при выводе результата перед значением угла выводить знак, используя оператор SGN (R).

Необходимо также иметь в виду, что арктангенс, как и другие обратные тригонометрические функции, является многозначной функцией (одному и тому же значению тангенса соответствует бесконечное множество углов). Функция языка Бейсик ATN (X) выдает угол в диапазоне от $-\pi/2$ до $+\pi/2$. До перевода в градусы, минуты, секунды (до действий, предусмотренных операторами 120—150) этот угол следует преобразовать в соответствии с конкретной задачей. Например, при решении обратной геодезической задачи можно вначале вычислить острый угол (обозначим его A):

```
100 X=X2-X1
110 Y=Y2-Y1
120 A=ATN (Y/X)*180/3.141593
```

Этот угол будет отрицательным ($-90° < A \leqslant 0°$), если приращения координат имеют разные знаки, и положительным ($0° < A \leqslant 90°$), если они имеют одинаковые знаки. Обратите также внимание на то, что в строке 120 угол A вычисляется в градусах с десятичной дробной частью. Дирекционный угол является положительным углом в диапазоне от 0 до 360°. Преобразовать острый угол A в дирекционный можно по следующим правилам:

$\alpha_{дир} = A$, если $X > 0$, $Y \geqslant 0$;
$\alpha_{дир} = 180° + A$, если $X < 0$, независимо от знака Y;
$\alpha_{дир} = 360° + A$, если $X > 0$, $Y < 0$.

Реализация этих правил с помощью операторов языка Бейсик будет рассмотрена в разделе, посвященном разветвляющимся алгоритмическим структурам. Там же будет рассмотрена ситуация, когда $X = 0$ и нельзя воспользоваться оператором из строки 120 (деление на 0).

Чтобы определить угол по заданному значению котангенса, синуса или косинуса, необходимо предварительно вычислить тангенс, а затем с помощью арктангенса вычислить угол. Например, если известно значение $C = \cos \alpha$, то вначале необходимо вычислить тангенс:

```
90 LET T=SQR (1-C*C)/C,
```

а затем с помощью операторов 100—150 (см. пример на стр. 19) определить угол.

В заключение рассмотрим примеры использования операторов INPUT, PRINT, LET, END.

Пример 1.

Написать программу вычисления расстояния между двумя точками с координатами X1, Y2 и X2, Y2 по формуле

$$S=\sqrt{\Delta X^2+\Delta Y^2},$$

где $\Delta X=X2-X1$, $\Delta Y=Y2-Y1$. Результат округлить до 0,01.

До составления программы необходимо выбрать наименования переменных. Их выбирают произвольно, но с соблюдением правил языка. Координаты точек и расстояние между ними обозначим так же, как они обозначены в условии задачи. Поскольку имена ΔX и ΔY в языке Бейсик недопустимы, обозначим приращения переменными X, Y. Поскольку приращения ΔX и ΔY (в программе X и Y) могут оказаться меньшими или равными нулю, воспользоваться для этих данных операцией возведения в степень (↑) нельзя. Заменим ее умножением.

Рассмотрим один из вариантов программы:

```
10 PRINT "ВВЕСТИ КООРДИНАТЫ"
20 PRINT "ТОЧКИ 1"
30 INPUT X1, Y1
40 PRINT "ТОЧКИ 2";
50 INPUT X2, Y2
60 LET X=X2-X1
70 LET Y=Y2-Y1
80 LET S=SQR (X*X+Y*Y)
90 PRINT "РАССТОЯНИЕ=" INT (S*100+0.5)/100" M"
100 END
```

Пример 2.

Написать программу вычисления площади прямоугольного участка по значению его основания A и угла α между диагональю и основанием. Формула для вычисления площади (в га):

$$P=\frac{A^2*\operatorname{tg}\alpha}{10\,000}.$$

Угол α вводить в градусах, минутах, секундах. Площадь округлить до 0,001 га.

Поскольку в языке Бейсик отсутствует функция $\operatorname{tg}\alpha$, то ее придется вычислять, как $\sin\alpha/\cos\alpha$. При этом дважды понадобится угол α в радианах. Его удобно вычислить заранее, используя вспомогательную переменную.

Вариант программы:

```
10 PRINT "ОСНОВАНИЕ В МЕТРАХ=";
20 INPUT A
30 PRINT "УГОЛ В ГРАДУСАХ, МИН, С=";
40 INPUT G,M,S
50 LET R=((S/60+M)/60+G)*3.141593/180
60 LET P=A*A* SIN (R)/COS (R)/10000
70 LET P=INT (P*1000+0.5)/1000
80 PRINT "ПЛОЩАДЬ=" P" ГА"
90 END
```

1.5. Понятие алгоритма. Линейные и разветвляющиеся алгоритмические структуры, средства их реализации в программах на языке Бейсик

Одним из основных понятий теории программирования является понятие алгоритма. Алгоритм — это последовательность предписаний, выполнение которых приводит к получению необходимого результата. Однако не всякая последовательность предписаний является алгоритмом. Алгоритм должен обладать следующими свойствами:

1. *Однозначность предписаний.* Каждое предписание должно допускать одно единственное толкование. Свойство однозначности позволяет поручить выполнение алгоритма автомату.

2. *Результативность.* Алгоритм должен обязательно приводить либо к искомому результату, либо к сообщению о том, что в данной ситуации алгоритм не применим.

3. *Массовость.* Алгоритм должен приводить к искомому результату при различных наборах исходных данных, допустимых для решения задач данного класса.

Последовательность предписаний, из которых состоит алгоритм, должна быть так или иначе записана. Одним из средств записи алгоритмов являются языки программирования.

Алгоритм, представленный в форме программы, написанной на том или ином языке программирования, может быть введен в ЭВМ, ею прочитан и выполнен. Однако на стадии разработки алгоритма языки программирования не всегда удобны, так как они не обладают достаточной наглядностью и требуют описания таких деталей, которые необходимы лишь на стадии реализации алгоритма машиной.

Более наглядным способом описания алгоритма на стадии его разработки является блок-схема. Блок-схема представляет собой совокупность блоков, каждый из которых содержит то или иное предписание. Блоки соединяются между собой стрелками, показывающими последовательность выполнения предписания. Обозначения блоков в зависимости от содержащихся в них предписаний приведены на рис. 3. Блок-схема алгоритма решения задачи из примера 1 (с. 21) приведена на рис. 4. При разработке простых программ и при наличии некоторого опыта в программировании блок-схемы, как правило, не нужны.

Для решения одной и той же задачи почти всегда можно предложить несколько алгоритмов. Лучшим из них будет тот, который обладает бóльшей ясностью, не содержит запутанных связей между блоками, легко допускает внесение изменений и исправлений. Перечисленным требованиям отвечают так называемые структурированные алгоритмы. Структурированный алгоритм представляет собою комбинацию из основных алгоритмических структур. Обычно различают пять типов основных алгоритмических структур:

линейную;

разветвляющуюся типа "ЕСЛИ — ТО";

разветвляющуюся типа "ЕСЛИ — ТО — ИНАЧЕ";

циклическую типа "ПОКА — ВЫПОЛНЯТЬ" (структура с предпроверкой);

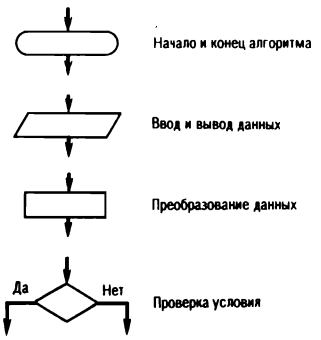


Рис. 3. Условные обозначения в блок-схемах

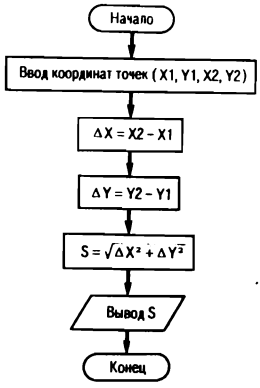


Рис. 4. Пример блок-схемы алгоритма решения задачи из примера 1

циклическую типа "ВЫПОЛНЯТЬ — ПОКА" (структура с постпроверкой).

Линейные алгоритмические структуры представляют собою последовательности предписаний, которые всегда выполняются в одном и том же порядке. Рассмотренный выше алгоритм является примером линейной алгоритмической структуры. Для описания линейных алгоритмических структур в языке Бейсик используются уже известные нам операторы INPUT, LET, PRINT, END. Соответствующие примеры были рассмотрены в разделе 1.4.

Разветвляющиеся алгоритмические структуры (рис. 5) содержат предписания, которые могут быть выполнены или пропущены, в зависимости от результатов проверки некоторого условия. Выбор той или иной структуры зависит от характера программируемой задачи и от возможностей языка программирования.

Структура "ЕСЛИ — ТО". Для описания разветвляющейся алгоритмической структуры типа "ЕСЛИ — ТО" в языке Бейсик используют условный оператор. Формат оператора:

IF условие THEN оператор, где IF означает "если", THEN — "то".

Условие записывается в виде двух арифметических выражений, разделенных одним из знаков отношения: $>$, $<$, $=$, $>=$ (вместо $\geqslant$), $<=$ (вместо $\leqslant$), $<>$ (вместо $\neq$). Например, A$>$10, B*C $<$Ø, T+K$>$=D/2, P$<>$ Ø .

В случае выполнения условия работает оператор, расположенный после слова THEN. При невыполнении условия оператор, расположенный после слова THEN, пропускается. Например, структура, приведенная на рис. 6, реализуется с помощью условного оператора

IF R $<$15 THEN LET P=R/2+D .

После слова THEN можно располагать любой оператор, в том числе оператор перехода, имеющий формат

GOTO номер строки.

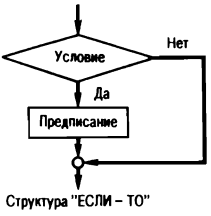


Рис. 5. Типы разветвляющихся алгоритмических структур

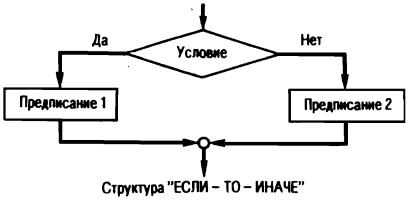
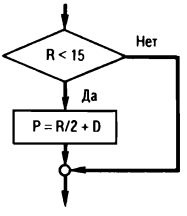


Рис. 6. Пример алгоритмической структуры «ЕСЛИ—ТО»



GOTO означает «идти к». При выполнении этого оператора происходит переход к указанной строке. Например, оператор 100 GOTO 250 осуществляет переход из строки 100 в строку 250, а оператор

```
100 IF A>ØTHEN GOTO 250
```

осуществляет переход в строку 250 только в том случае, когда A>Ø.

Условный оператор, в состав которого входит оператор перехода, иногда называют оператором условного перехода.

В большинстве реализаций языка Бейсик условие влияет не на один оператор, расположенный справа от слова THEN, а на все операторы, расположенные справа от THEN в данной строке. Например, структуру, изображенную на рис. 7, можно реализовать одним оператором

```
100 IF R<15 THEN LET P=R/2+D:PRINT P, R, D
```

Схему, изображенную на рис. 7, нельзя реализовать средствами языка Бейсик в том случае, когда группа операторов, зависящих от условия, не помещается в одной строке. Кроме того, размещение нескольких операторов в одной строке уменьшает инвариантность программы по отношению к различным версиям языка. Более универсальной

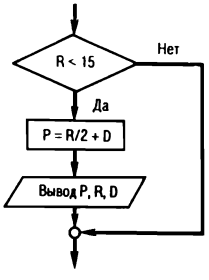


Рис. 7. Структура «ЕСЛИ—ТО», в которой предписание требует выполнения двух действий

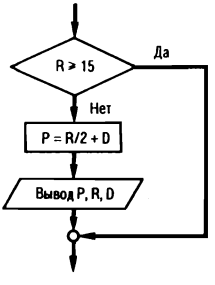


Рис. 8. Структура, полученная в результате замены исходного условия на альтернативное

является схема, изображенная на рис. 8, в которой исходное условие R<15 заменено на альтернативное R>=15 и, соответственно, изменены наименования выходов логического блока. Программная реализация этой схемы может иметь следующий вид:

```
100 IF R>=15 THEN GOTO 130
100 LET P=R/2+D
120 PRINT P,R,D
```

Непосредственная реализация структуры, изображенной на рис. 7 (без замены условия на его альтернативу и без размещения нескольких операторов в одной строке), получается более громоздкой:

```
100 IF R<15 THEN GOTO 120
110 GOTO 140
120 LET P=P/2+D
130 PRINT P, R, D
```

В исходной версии языка Бейсик оператор условного перехода разрешается писать без слов GOTO, например:

```
100 IF R>=15 THEN 40
```

В дальнейшем иногда будем использовать эту сокращенную форму записи.

В некоторых версиях языка опускают слово THEN, например:

```
100 IF R>=15 GOTO 40
```

Рассмотрим фрагмент программы, содержащей структуру «ЕСЛИ — ТО»: определение угла по заданному значению тангенса с корректировкой результата в случае S=60. Начальная часть фрагмента приведена на с. 19.

Продолжим его:

```
160 IF S<60 THEN 220      |
170 LET S=0               |
180 LET M=M+1             | A
190 IF M<60 THEN 220      |
200 LET M=0       } B     |
210 LET G=G+1     }       |
220 PRINT "УГОЛ В ГРАДУСАХ, МИН, С=" G; M; S
```

Данный фрагмент содержит две структуры «ЕСЛИ — ТО»; вложенные одна в другую. Внешняя структура требует выполнения предписания А при условии M=60, внутренняя — выполнения предписания В при условии M=60. Операторы 160 и 190 проверяют альтернативные условия: S<60 и M<60 (случаи S>60 и M>60 в данной задаче не могут иметь место).

Структура «ЕСЛИ — ТО — ИНАЧЕ». Схема реализации структуры «ЕСЛИ — ТО — ИНАЧЕ» на языке Бейсик приведена на рис. 9. В схеме, помимо оператора условного перехода, использован оператор

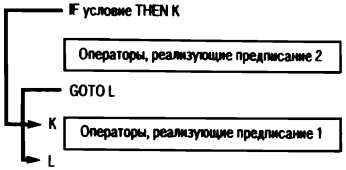


Рис. 9. Схема реализации структуры «ЕСЛИ—ТО—ИНАЧЕ» средствами языка Бейсик

GOTO, необходимый для того, чтобы пропустить предписание 1 в том случае, когда выполняется предписание 2. Ниже приведен пример программной реализации структуры «ЕСЛИ — ТО — ИНАЧЕ», в которой по заданным значениям координаты X для двух точек необходимо определить, где находится вторая точка по отношению к первой:

```
100 LET D=X2-X1
110 IF D>=0 THEN 140
120 PRINT "ЮГО-ВОСТОК, ЮГ, ЮГО-ЗАПАД"
130 GOTO 150
140 PRINT "ЗАПАД, СЕВЕРО-ЗАПАД, СЕВЕР, СЕВЕРО-
    ВОСТОК, ВОСТОК"
150
```

Структуру «ЕСЛИ — ТО — ИНАЧЕ» можно реализовать также путем проверки условия, альтернативного исходному. Никакими достоинствами или недостатками этот способ не обладает. Просто в программе операторы, реализующие предписание 1 и предписание 2, поменяются местами:

```
110 IF D<0 THEN 140
120 PRINT «ЗАПАД, СЕВЕРО-ЗАПАД, СЕВЕР, СЕВЕРО-
    ВОСТОК, ВОСТОК"
130 GOTO 150
140 PRINT "ЮГО-ВОСТОК, ЮГ, ЮГО-ЗАПАД"
150
```

Рассмотрим еще два примера.

Пример 4. Определение дирекционного угла направления из точки с координатами X1, Y1 в точку с координатами X2, Y2. Это пример ситуации, описанной на с. 20, когда острый угол, вычисленный с помощью функции ATN, необходимо преобразовать в угол в диапазоне от 0 до 360° с тем же значением арктангенса. Необходимо также учесть ситуацию, когда приращение $\Delta X=0$ (дирекционный угол равен 90 или 270°) и воспользоваться формулой $\alpha=\text{arctg}\,(\Delta Y/\Delta X)$ нельзя.

Ниже приведен фрагмент программы (приращения ΔX и ΔY обозначены, соответственно, переменными X и Y, острый угол — переменной A, дирекционный угол — переменной D):

```
100  }
...  }   Ввод X1, Y1, X2, Y2
200 X=X2-X1
210 Y=Y2-Y1
220 IF X<>0 THEN 310
```

```
230 IF Y<>0 THEN 260
240 PRINT "ТОЧКИ СОВПАЛИ"
250 GOTO 100
260 IF Y>0 THEN 290
270 D=270
280 GOTO 400
290 D=90
300 GOTO 400
310 A=ATN (Y/X)*180/3.141593
320 IF X>0 THEN 350
330 D=180+A
340 GOTO 400
350 IF Y>=0 THEN 380
360 D=360+A
370 GOTO 400
380 D=A
400 } Перевод дирекционного угла в градусы, минуты, секунды
... } и вывод результата.
```

Приведенный выше фрагмент содержит пять вложенных друг в друга структур «ЕСЛИ — ТО — ИНАЧЕ»:

1-я структура (внешняя). Условие проверяется в строке 220. При выполнении условия выполняется предписание B (строки 310—380), в противном случае выполняется предписание A (операторы 230—290).

2-я структура (предписание A). Условие проверяется в строке 230. При выполнении условия выполняется предписание C (строки 260—290), в противном случае выполняется предписание, расположенное в строке 240.

3-я структура (предписание C). Условие проверяется в строке 260. При выполнении условия выполняется предписание из строки 290, в противном случае — выполняется предписание, расположенное в строке 270.

4-я структура (часть предписания B). Условие проверяется в строке 320. При выполнении условия выполняется предписание D (строки 350—380), в противном случае — предписание из строки 330.

5-я структура (предписание D). Условие проверяется в строке 350. При выполнении условия выполняется предписание, расположенное в строке 380, в противном случае — предписание из строки 360.

В рассмотренном примере каждая разветвляющаяся структура представляет собою самостоятельную программную единицу.

Пример 5. Рассмотрим фрагмент программы для определения вхождения точки *2* (рис. 10) в сектор $\pm 10°$ по отношению к линии, проведенной из точки *1* на восток:

```
100 Y=Y2-Y1
110 IF Y<=0 THEN 180
120 X=X2-X1
130 T=ABS (X/Y)
140 A=ATN (T)*180/3.141593
150 IF A>10 THEN 180
```

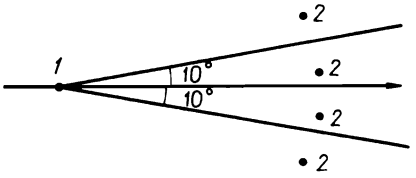


Рис. 10. Иллюстрация к примеру 5

```
160 PRINT "ТОЧКА В ЗАДАННОМ СЕКТОРЕ"
170 GOTO 190
180 PRINT "ТОЧКА ЗА ПРЕДЕЛАМИ СЕКТОРА"
190
```

Здесь мы имеем дело с двумя структурами «ЕСЛИ — ТО — ИНАЧЕ». В первой из них условие проверяется в строке 110, а предписания размещены в строке 180 и в строках 120—160. Второе из этих предписаний, в свою очередь, содержит структуру «ЕСЛИ — ТО — ИНАЧЕ», расположенную в строках 150—180: условие — в строке 150, предписания — в строках 180 и 160. В приведенном примере одно и то же предписание (строка 180) является одновременно элементом двух разветвляющихся структур. Это нарушает структурность программы (входящие в ее состав структуры не являются самостоятельными единицами). Данную задачу можно запрограммировать структурно, но для этого оператор

```
PRINT "ТОЧКА ЗА ПРЕДЕЛАМИ СЕКТОРА"
```

придется повторить дважды.

Проверка сложных условий. Часто необходимость выполнения того или иного предписания зависит от нескольких условий. Например, необходимо выполнить оператор M=M+1 в том случае, когда одновременно выполняются два условия: X>0 и Y>0. (Если X и Y — приращения координат заданной точки по отношению к некоторой исходной точке, то выполнение этих условий означает, что заданная точка находится на северо-восток от исходной).

Поскольку в условном операторе после слова THEN может находиться любой оператор, в том числе и условный, поставленное требование можно записать одним оператором:

```
IF X>0 THEN IF Y>0 THEN M=M+1.
```

Такой условный оператор часто называют вложенным. Существуют версии языка Бейсик, которые позволяют обходиться без вложенных условных операторов. В этих версиях при записи условий допускается использование большего количества средств из числа тех, которые в математике используются при записи логических выражений.

Строго говоря, условие, которое мы до сих пор писали после слова IF (X>0, A>10, Y<=0 и т. п.), является простейшей разновидностью логического выражения — отношением. Приведем еще несколько примеров отношений: A<>0, B+C<2*D, Q/3<10, X>=Y, Z>2*S-3.

Логическое выражение так же, как и арифметическое, это конструкция, с помощью которой вычисляется некоторый результат — значение выражения. Результатом вычисления арифметического выражения является число, результат вычисления логического выражения — это одно из двух логических значений: ИСТИНА (TPUE) или ЛОЖЬ (FALSE). Например, логическое выражение A> 10 при A=5 — ложно, при A=10 — ложно, при A=15 — истинно.

Используя понятие логического выражения, еще раз рассмотрим структуру и правила выполнения условного оператора:

структура: IF логическое выражение THEN оператор;

правила выполнения: оператор, расположенный после слова THEN, выполняется в том случае, когда логическое выражение истинно, и пропускается, если это выражение ложно.

Помимо отношений, логическое выражение может содержать знаки логических операций. В тех версиях языка Бейсик, в которых использование этих знаков допустимо, их изображают с помощью слов NOT (операция отрицания), AND (операция логического умножения) и OR (операция логического сложения). Русский перевод: NOT — не, AND — и, OR — или. Здесь операций перечислены в порядке приоритета их выполнения: в первую очередь выполняется операция NOT, затем AND и в последнюю очередь OR. Для изменения приоритета, а также для придания логическому выражению большей наглядности можно использовать круглые скобки.

С помощью знаков AND или OR можно соединить два или больше логических выражений в одно логическое выражение:

(лог. выр. 1) AND (лог. выр. 2) AND (лог. выр. 3) (1)
(лог. выр. 1) OR (лог. выр. 2) OR (лог. выр. 3) (2)
(лог. выр. 1) AND (лог. выр. 2) OR (лог. выр. 3) (3)
(лог. выр. 1) AND ((лог. выр. 2) OR (лог. выр. 3)). (4)

Логическое выражение, состоящее из нескольких, соединенных знаками AND, является истинным в том случае, когда истинны все входящие в его состав выражения, и будет ложным, если, по крайней мере, одно из этих выражений ложно. Логическое выражение, состоящее из нескольких соединенных знаками OR, является истинным в том случае, когда истинно, по крайней мере, одно из входящих в его состав выражений, и ложно только тогда, когда ложны все эти выражения. В более сложных ситуациях истинность логических выражений вычисляется исходя из приоритета операций и расположения скобок. Значения истинности логических выражений (1)—(4) в зависимости от истинности компонентов, которые входят в состав этих выражений, приведены в табл. 4.

Используя операцию AND, можно переписать рассмотренный раньше оператор

IF X > 0 THEN IF Y > 0 THEN M=M+1

иначе: IF (X > 0) AND (Y > 0) THEN M=M+1 .

Приоритет вычисления отношений выше приоритета операций NOT, AND и OR, поэтому скобки в приведенном примере необязательны.

Таблица 4
Таблица истинности логических выражений (1) — (4)

Лог. выр. 1	Лог. выр. 2	Лог. выр. 3	(1)	(2)	(3)	(4)
Истинно	Истинно	Истинно	Истинно	Истинно	Истинно	Истинно
Истинно	Истинно	Ложно	Ложно	Истинно	Истинно	Истинно
Истинно	Ложно	Истинно	Ложно	Истинно	Истинно	Истинно
Истинно	Ложно	Ложно	Ложно	Истинно	Ложно	Ложно
Ложно	Истинно	Истинно	Ложно	Истинно	Истинно	Ложно
Ложно	Истинно	Ложно	Ложно	Истинно	Ложно	Ложно
Ложно	Ложно	Истинно	Ложно	Истинно	Истинно	Ложно
Ложно	Ложно	Ложно	Ложно	Ложно	Ложно	Ложно

Однако многие трансляторы воспринимают отсутствие скобок в данной ситуации, как синтаксическую ошибку. Помимо прочего, скобки делают текст логического выражения более наглядным.

Усложним пример, приведенный в начале параграфа. Будем считать, что в случае, когда $X > 0$ и $Y > 0$, необходимо выполнить три оператора:

```
M=M+1
X2 (M)=X1 (I)
Y2 (M)=Y1 (I)
```

Рассмотрим несколько вариантов программной реализации этого примера.

Вариант 1.

```
IF (X>0) AND (Y>0) THEN M=M+1:X2 (M)=X1 (I):
:Y2 (M)=Y1 (I)
```

Здесь все три оператора, реализующие предписание, расположены в одной строке. Такая структура условного оператора нежелательна по причинам, изложенным на с. 24.

Вариант 2. Используем условие, альтернативное условию $(X > 0)$ AND $(Y > 0)$, в случае его выполнения будем пропускать операторы, которые реализуют предписание. Аналогичный пример приведен на с. 25. Любое логическое выражение можно превратить в альтернативное, поместив перед ним знак логической операции NOT:

```
NOT ((X>0) AND (Y>0)).
```

Вместо этого можно заменить операцию AND на операцию OR, а логические выражения (в данном случае отношения), стоящие по обе стороны от этой операции, заменить на их альтернативы:

```
(X<=0) OR (Y<=0).
```

Действительно, последнее логическое выражение будет ложным в случае, когда выражение $(X > 0)$ AND $(Y > 0)$ — истинно, и будет

истинным, когда выражение (X > 0) AND (Y > 0) — ложно. Запишем теперь соответствующий фрагмент программы:

```
100 IF (X<=0) OR (Y<=0) THEN 140
110 M=M+1
120 X2 (M)=X1 (1)
130 Y2 (M)=Y1 (1)
140
```

Вариант 3. Если версия языка Бейсик, с которым мы работаем, не позволяет использовать логические операции NOT, AND, OR, то в варианте 2 оператор из строки 100 можно заменить двумя операторами:

```
100 IF X<=0 THEN 140
105 IF Y<=0 THEN 140
```

Эта пара операторов будет осуществлять переход в строку 140 в случае, когда истинно отношение X<=0, или истинно отношение Y<=0, или истинны оба эти отношения, т. е. эта пара операторов будет работать так же, как и оператор 100 в предыдущем варианте.

Из приведенных в данном параграфе примеров можно вывести два общих правила:

1. Условный оператор, в котором после слова IF находится логическое произведение, можно заменить группой вложенных друг в друга условных операторов.

2. Условный оператор, в котором после слова IF находится логическая сумма, можно заменить группой условных операторов, расположенных в последовательно идущих строках.

В заключение рассмотрим еще один пример.

Пример 6. Написать программу для вычисления площади треугольника (в га) по формуле Герона

$$S=\sqrt{P(P-A)(P-B)(P-C)}/10\,000,$$

где A, B, C — стороны треугольника (в м); P=(A+B+C)/2. В программе предусмотреть проверку: можно ли из отрезков длиной A, B и C построить треугольник. Построить треугольник можно в том случае, когда сумма любой пары сторон больше третьей стороны. Если треугольник построить нельзя, программа должна выдать соответствующее сообщение и завершить работу.

Рассмотрим несколько вариантов программы.

Вариант 1. Условие существования треугольника записываем в виде одного логического выражения:

```
10 PRINT "ВВОДИТЕ СТОРОНЫ ТРЕУГОЛЬНИКА В МЕТРАХ:"
20 PRINT "A=";
30 INPUT A
40 PRINT "B=";
50 INPUT B
60 PRINT "C=";
70 INPUT C
```

```
80 IF (A+B>C) AND (A+C>B) AND (B+C>A) THEN 110
90 PRINT "ИЗ ОТРЕЗКОВ" A; B; C "ТРЕУГОЛЬНИК ПОСТ-
   РОИТЬ НЕЛЬЗЯ"
100 GOTO 140
110 P=(A+B+C)/2
120 S=SQR (P*(P-A)*(P-B)*(P-C))/10000
130 PRINT "ПЛОЩАДЬ=" S "ГА"
140 END
```

Вариант 2. Условие существования треугольника проверяется с помощью трех вложенных условных операторов. Для этого изменяем строку 80:

```
80 IF A+B>C THEN IF A+C>B THEN IF B+C>ATHEN 110.
```

Вариант 3. Проверяем условие невозможности построить треугольник из данных отрезков. Это условие альтернативно условию существования треугольника. Воспользуемся сформулированным в данном параграфе правилом преобразования логического выражения в его альтернативу: заменим операции AND на операции OR, а соединенные этими операциями отношения заменим на их альтернативы. В результате строка 80 приобретет вид:

```
80 IF ((A+B<=C) OR (A+C<=B) OR (B+C<=A) THEN
   130
```

а предписания, расположенные в строках 90 и 100—130 поменяются местами:

```
 90 P=(A+B+C)/2
100 S=SQP (P*(P-A)*(P-B)*(P-C))/10000
110 PRINT "ПЛОЩАДЬ=" S "ГА"
120 GOTO 140
130 PRINT "ИЗ ОТРЕЗКОВ" A; B; C "ТРЕУГОЛЬНИК ПОСТ-
    РОИТЬ НЕЛЬЗЯ"
140 END
```

Вариант 4. Условие невозможности построить треугольник проверяем с помощью трех условных операторов, расположенных в последовательно идущих строках. Для этого строку 80 заменяем тремя строками:

```
80 IF A+B<=C THEN 130
81 IF A+C<=B THEN 130
82 IF B+C<=A THEN 130
```

1.6. Циклические структуры

Циклической называется такая алгоритмическая структура, которая содержит в себе многократно повторяемый участок — тело цикла. Остальные элементы циклической структуры предназначены для обеспечения нужного числа повторений тела цикла. Типовые циклические структуры были перечислены в разделе 1.5. Блок-схемы этих структур приведены на рис. 11 и 12. Из схем видно, что в структуре

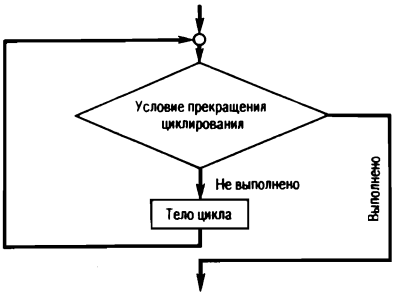


Рис. 11. Циклическая структура с предпроверкой

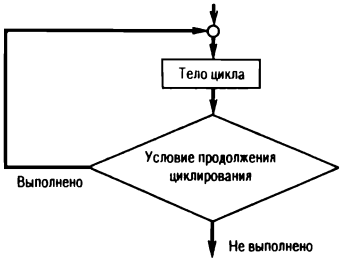


Рис. 12. Циклическая структура с постпроверкой

с предпроверкой может сложиться ситуация, когда тело цикла не будет выполнено ни разу, в структуре с постпроверкой обязательно, по крайней мере, одно выполнение тела цикла.

В программе на языке Бейсик циклическую структуру можно создать с помощью операторов условного перехода.

Пример 7. В машину вначале вводятся координаты исходной точки (X_0, Y_0), затем последовательно вводятся координаты периферийных точек (X, Y). После появления точки, расположенной на северо-восток от исходной, необходимо прекратить ввод и сообщить номер точки. Предполагается, что по крайней мере одна периферийная точка находится на северо-востоке от исходной.

Программа:

```
 10 LET K=Ø
 20 PRINT "ВВОДИТЕ КООРДИНАТЫ ИСХОДНОЙ ТОЧКИ";
 30 INPUT XØ, YØ
 40 PRINT "ВВОДИТЕ КООРДИНАТЫ ОЧЕРЕДНОЙ ПЕРИФЕ-
    РИЙНОЙ ТОЧКИ"
 50 INPUT X, Y
 60 LET K=K+1
 70 IF X<=XØ THEN 40
 80 IF Y<=YØ THEN 40
 90 PRINT "ТОЧКА"K"—НА СЕВЕРО-ВОСТОКЕ"
100 END
```

В программе реализован цикл с постпроверкой. Тело цикла представлено операторами 40—60. Условие продолжения циклирования проверяется операторами 70, 80. Эти операторы проверяют истинность логического выражения (X<XØ) OR (Y≤YØ). Если это выражение истинно, значит очередная периферийная точка не находится на северо-востоке и тело цикла следует повторить. В теле цикла после ввода координат очередной периферийной точки (операторы 40, 50) оператор 60 увеличивает значение переменной K на 1. До начала цикла оператор 10 присваивает этой переменной нулевое значение. В результате K всегда равно номеру только что введенной периферийной точки.

Общая схема реализации цикла с постпроверкой:

```
100
...   } Тело цикла
200
210  IF условие продолжения цикла THEN 100
```

Для организации цикла с предпроверкой, кроме оператора условного перехода, потребуется еще оператор GOTO:

```
 90  IF условие прекращения цикла THEN 220
100
...   } Тело цикла
200
210  GOTO 90
220
```

В приведенных выше схемах условия можно заменить на альтернативные: в строке 210 (цикл с постпроверкой) можно проверять условие прекращения цикла, а в строке 90 (цикл с предпроверкой) — проверять условие продолжения цикла. Однако такая замена усложнит программу.

Цикл с постпроверкой:

```
100
...   } Тело цикла
200
210  IF условие прекращения цикла THEN 230
220  GOTO 100
230
```

Цикл с предпроверкой:

```
 90  IF условие продолжения цикла THEN 100
 95  GOTO 220
100
...   } Тело цикла
200
210  GOTO 90
220
```

Очевидно, что до наступления очередной проверки на продолжение или прекращение циклирования должны измениться значения некоторых переменных, участвующих в этой проверке. Иначе циклирование либо никогда не наступит, либо никогда не завершится. В примере 7 такими переменными были X и Y. Эти переменные обычно называют параметрами цикла. В приведенном примере нельзя сформулировать правило, по которому изменяются параметры X и Y. Их значения зависят от размещения на карте периферийных точек.

Для организации циклов, в которых параметр изменяется по закону арифметической прогрессии, в языке Бейсик существуют операторы FOR и NEXT. Ниже приведена схема использования этих операторов:

```
FOR параметр цикла =E1 TO E2  STEP E3
```

Тело цикла
NEXT параметр цикла

В качестве параметра цикла можно использовать любую простую переменную; E1, E2, E3 — арифметические выражения (в простейшем случае — числа или переменные). С помощью E1 вычисляется начальное значение параметра цикла, с помощью E2 — конечное значение, с помощью E3 — шаг изменения цикла (если этот шаг равен единице, то STEP1 можно не писать).

Тело цикла выполняется при каждом новом значении параметра цикла до тех пор, пока это значение не окажется больше (при положительном шаге) или меньше (при отрицательном шаге), чем значение выражения E2.

Примеры оператора FOR:

```
FOR Z=5 TO 105 STEP 10
```

(параметр Z принимает значения от 5 до 105 с шагом 10).

```
FOR K=3/L TO P*8 STEP D
```

(начальное значение параметра K вычисляется с помощью выражения 3/L, конечное — по формуле P*8, шаг изменения параметра K равен значению переменной D).

```
FOR I=3 TO 40 STEP 1
```

или

```
FOR I=3 TO 40
```

(параметр I изменяется от 3 до 40 с шагом 1).

```
FOR P=4 TO -10 STEP -2
```

(параметр P последовательно принимает значения 4, 2, 0, −2, −4, −6, −8, −10).

Обратите внимание на некоторые особенности использования и выполнения операторов FOR и NEXT:

1. В операторах FOR и NEXT, организующих некоторый цикл, указывается один и тот же параметр цикла. Например,

```
100 FOR L=5 TO 37 STEP 2
...  Тело цикла
200 NEXT L
```

Здесь L — параметр цикла.

2. Операторы FOR и NEXT организуют цикл с предпроверкой. Следовательно, возможно возникновение ситуаций, в которых тело цикла не будет выполнено ни разу. Например, ни разу не будет выполнено тело цикла, если при выполнении участка программы

```
100 INPUT A, B, C
110 FOR I=A TO B STEP C
...  Тело цикла
200 NEXT I
```

ввести A=20, B=8, C>0.

3. В описании языка Бейсик не оговорено значение параметра после завершения цикла. Обычно параметр сохраняет то значение, которое он имел во время последнего выполнения тела цикла.

Рассмотрим примеры нескольких задач, которые на языке Бейсик удобно программировать с использованием операторов FOR — NEXT.

Пример 8. Вычислить значения синусов углов от 6 до 87° через каждые 3°. Здесь параметром цикла является угол (обозначим его переменной A).

Вариант программы:

```
10 FOR A=6 TO 87 STEP 3
20 LET S=SIN (A*3.141593/180)
30 PRINT "SIN("A")=" S
40 NEXT A
50 END
```

Пример 9. Вычислить среднее значение 17 чисел по формуле

$$x=\frac{1}{17}\sum_{i=1}^{17} x_i.$$

Программа решения этой задачи может иметь вид:

```
10 LET S=0
20 FOR I=1 TO 17
30 PRINT "ВВЕСТИ X" I;
40 INPUT X
50 LET S=S+X
60 NEXT I
70 LET P=S/17
80 PRINT "СРЕДНЕЕ=" P
90 END
```

В программе сумма обозначена переменной S. Вначале оператор 10 присваивает этой переменной значение 0. Для передачи значений x_i в сумму используется переменная X. В операторе 40 эта переменная получает значение очередного x_i. Затем оператор 50 прибавляет это значение к сумме S. После этого тело цикла (операторы 30—50) повторяется: переменная X получает значение следующего x_i и передает его в сумму. Оператор 20 обеспечивает семнадцатикратное повторение тела цикла. Параметр цикла (переменная I) служит счетчиком повторений и используется в теле цикла для организации диалога: оператор 30 запрашивает значение очередного x_i и указывает его номер (I). Операторы 70 и 80 вычисляют и выводят на экран среднее значение, которое в программе получает переменная P.

Пример 10. Вычислить среднее значение произвольного количества чисел:

$$\bar{x}=\frac{1}{N}\sum_{i=1}^{N} x_i.$$

В отличие от предыдущей задачи здесь заранее неизвестно количество суммируемых чисел. Программа решения задачи может иметь следующий вид:

```
10 PRINT "КОЛИЧЕСТВО СЛАГАЕМЫХ=";
20 INPUT N
```

```
30 LET S=0
40 FOR I=1 TO N
50 PRINT "X" I "=";
60 INPUT X
70 LET S=S+X
80 NEXT I
90 LET P=S/N
100 PRINT "СРЕДНЕЕ=" P
110 END
```

Здесь оператор 10 запрашивает, а оператор 20 читает количество усредняемых чисел (N). Затем N используется в операторе 40 для ограничения количества повторений тела цикла и в операторе 90 для вычисления среднего.

Бесконечный цикл. Помимо рассмотренных выше типовых циклических структур, в языке Бейсик иногда полезно использовать так называемый бесконечный цикл. Бесконечный цикл допустим при наличии операторов, осуществляющих приостановку выполнения программы в ожидании ответных действий пользователя. В языке Бейсик таким оператором является оператор INPUT (см. с. 14).

Рассмотрим пример программы, содержащей бесконечный цикл.

Пример 11.

```
10 PRINT "УГОЛ В ГРАДУСАХ, МИН С=";
20 INPUT G, M, S
30 LET A=((S/60+M)/60+G)*3.141593/180.
40 LET S1=SIN (A)
50 PRINT "СИНУС УГЛА"=" S1
60 GOTO 10
```

Программа предназначена для вычисления синусов произвольного числа любых углов, заданных в градусах, минутах и секундах. Тело цикла составляют операторы с 10 по 50. Оператор 60 обеспечивает повторение тела цикла. В момент выполнения оператора 20 INPUT G, M, S происходит проиостановка выполнения программы, во время которой пользователь может либо ввести значение нового угла, либо прекратить выполнение программы, а значит и повторение тела цикла.

Таким образом, в случае бесконечного цикла контроль числа повторений тела цикла осуществляет пользователь в то время, как в других циклических структурах этот контроль осуществляется автоматически с помощью операторов IF или FOR.

Вложенные циклы. Тело цикла может, в свою очередь, содержать цикл. Такую конструкцию называют вложенными циклами. Максимально допустимое количество вложений зависит от конкретной реализации языка. Рассмотрим пример задачи, программа решения которой содержит вложенные циклы.

Пример 12. Написать программу вычисления и вывода таблицы значений функции

$$H = A + B \cdot \sin\alpha \cdot \cos\beta,$$

где угол α принимает значения от 10 до 40° с шагом 10°, а угол β

принимает значения от 10 до 70° с шагом 10°. Значения *A* и *B* могут быть любыми и вводятся во время выполнения программы.

Программа решения данной задачи может иметь следующий вид:

```
10 C=3.141593/180
20 PRINT "A=";
30 INPUT A
40 PRINT "B=";
50 INPUT B
60 PRINT "ВЫВОД НА LP: ИЛИ НА TT: ";
70 INPUT U⊗
80 OPEN U⊗FOR OUTPUT AS FILE #1
90 K=70
100 GOSUB 400
110 PRINT#1,, "A="A,, "B="B
120 GOSUB 400
130 PRINT #1,, "I",, "АЛЬФА"
140 PRINT #1," БЕТА ","I";
150 K=56
160 GOSUB 400
170 PRINT#1,,
180 FOR I=10 TO 40 STEP 10
190 PRINT #1, "I "I,
200 NEXT I
210 PRINT#1
220 K=70
230 GOSUB 400
240 FOR B1=10 TO 70 STEP 10
250 R=COS.(B1*C)
260 PRINT #1," "B1,
270 FOR A1=10 TO 40 STEP 10
280 H=A+B*SIN (A1*C)*R
290 PRINT #1, "I "H,
300 NEXT A1
310 PRINT #1
320 GOSUB 400
330 NEXT B1
340 GOTO 500
400 FOR L=1 TO K
410 PRINT #1, "—";
420 NEXT L
430 PRINT # 1
440 RETURN
500 END
```

Здесь оператор 10 вычисляет коэффициент перехода от градусов к радианам. Этот коэффициент используется в операторах 250 и 280. Операторы 60—80 осуществляют выбор устройства вывода (печать или экран). Операторы 90—230 печатают «шапку» таблицы. При этом для вывода горизонтальной линии используется подпрограмма, располо-

Таблица 5
Результаты решения задачи из примера 12

$A=100$			$B=200$	
БЕТА	АЛЬФА			
	10	20	30	40
10	134.202	167.365	198.481	226.604
20	132.635	164.279	193.969	220.805
30	130.077	159.24	186.603	211.334
40	126.604	152.401	176.604	198.481
50	122.324	143.969	164.279	182.635
60	117.365	134.202	150	164.279
70	111.878	123.396	134.202	143.969

женная в строках 400—440 (см. 1.9). Операторы 260 и 280 печатают остальные строки таблицы. Для обозначения углов α, β используются соответственно переменные A1, B1. Наружный цикл (операторы 240—330) осуществляет изменение угла β и вывод каждого нового значения β на экран или печать (оператор 260). Внутренний цикл (операторы 270—300) при постоянном β изменяет значения угла α, при каждом новом α вычисляет величину H и выводит ее на экран. Другими словами, наружный цикл осуществляет переход от строки к строке, а внутренний — внутри строки осуществляет переход от столбца к столбцу. После вывода очередной строки с помощью подпрограммы на экран или на печать выводится горизонтальная линия.

Пример печати результатов, полученных с помощью данной программы при $A=100$ и $B=200$, приведен в табл. 5.

1.7. Массив и переменная с индексом. Использование циклических структур для работы с массивами

При программировании многих задач бывает удобно обозначить группу переменных одним и тем же именем, а отличать эти одноименные переменные друг от друга по их номерам (индексам). Такие переменные называют переменными с индексами (см. с. 7), а совокупность однородных переменных, имеющих общее имя, называют массивом.

Вспомним правила записи переменных с индексами: P(2), R(I), D1(4) — переменные с одним индексом; A(3, 4), C(I, K), X(S, R) — переменные с двумя индексами.

В качестве индекса может быть использовано любое выражение, например:

S(I+2), X(SQR (P)), Y(2*R+D/2) и т. п.

Выражение, используемое в качестве индекса, называется индексным выражением. При вычислении индексного выражения оно округляется до ближайшего целого. Например, если в момент использования переменной Y(2*R+D/2) переменная R имела значение 0,75, а переменная D — значение 4,5, то фактически будет использована переменная Y(4).

В языке Бейсик индекс может принимать значения от 0 до 255. Не следует путать простые переменные и переменные с индексом: P2 — простая переменная, P(2) — переменная с индексом. Простая переменная — самостоятельное данное, переменная с индексом — элемент массива.

Совокупность одноименных переменных, каждая из которых имеет один индекс, образует одномерный массив. Совокупность одноименных переменных, каждая из которых имеет два индекса, образует двумерный массив. В языке Бейсик допустимо использование только одномерных и двумерных массивов.

Если в программе на языке Бейсик используется массив, то предварительно необходимо объявить его имя и максимальные значения индексов. Делается это с помощью оператора DIM (от английского слова dimension — размер). Формат этого оператора:

DIM имя массива (максимальные значения индексов).

Минимальное значение индекса всегда равно нулю. Примеры объявления массивов: DIM X(15) — объявлен одномерный массив, состоящий из 16 переменных от X(0) до X(15); DIM P(4, 3) — объявлен двумерный массив P, состоящий из 20 переменных:

P(0, 0)	P(0, 1)	P(0, 2)	P(0, 3)
P(1, 0)	P(1, 1)	P(1, 2)	P(1, 3)
P(2, 0)	P(2, 1)	P(2, 2)	P(2, 3)
P(3, 0)	P(3, 1)	P(3, 2)	P(3, 3)
P(4, 0)	P(4, 1)	P(4, 2)	P(4, 3)

Не все переменные объявленного массива обязательно использовать в данной программе. Например, можно объявить DIM P(10), а использовать в программе переменные с P(1) по P(10), не используя P(0). Это бывает удобно в тех случаях, когда в задаче нумерация переменных начинается с 1. Если программа предназначена для работы с массивом, длина которого заранее не оговорена, следует объявить массив с максимально возможным в данной задаче числом элементов. Например, объявив DIM(200), мы получаем возможность работать с любым массивом S, в котором номер последнего элемента не превосходит 200.

Использование переменных с индексами существенно расширяет возможности программирования. Рассмотрим два примера.

Пример 13.

```
100 PRINT A5*A6
110 PRINT A7*A8
120 PRINT A9*A10
130 PRINT A11*A12
```

Пример 14.

```
 10 DIM A(100)
100 PRINT "НАЧАЛЬНЫЙ ИНДЕКС=";
110 INPUT K
120 PRINT "КОНЕЧНЫЙ ИНДЕКС=";
130 INPUT L
140 FOR I=K TO L STEP 2
150 PRINT A(I)*A(I+1)
160 NEXT I
```

В примере 13 использованы простые переменные от A5 до A12, в примере 14 — переменные с индексом A(I) и A(I+1), где I может принимать значения от 0 до 99. При выполнении фрагмента программы из примера 13 вычисляются и печатаются значения произведений A5*A6, A7*A8, A9*A10, A11*A12. Те же произведения можно вычислить по программе из примера 14, если при выполнении операторов 100—130 ввести значения K=5 и L=11. При этом, если программа из примера 13 работает всегда с одними и теми же переменными от A5 до A12 и всегда вычисляет и печатает ровно четыре произведения, то в примере 14, при вводе соответствующих значений K и L, можно вычислить и отпечатать попарные произведения любой группы переменных от A(K) до A(L) в пределах $0 \leqslant K,\ L \leqslant 99$. Если пример 14 дополнить еще одной парой операторов

```
135 PRINT "ШАГ=";
136 INPUT S
```

а оператор 140 изменить:

```
140 FOR I=K TO L STEP S
```

то возможности программы станут еще шире.

Еще одна ситуация, в которой используется возможность отбора из массива указанной группы данных, рассмотрена в примере 15.

Пример 15.

```
100 PRINT "УКАЖИТЕ НАЧАЛЬНЫЙ И КОНЕЧНЫЙ НОМЕРА
    УЧАСТКОВ";
110 INPUT N1, N2
120 FOR I=N1 TO N2
130 PRINT "УЧАСТОК" I "ПЛОЩАДЬ=" P(I) "␣ГА"
140 NEXT I
```

Приведенный фрагмент программы выводит на экран площади земельных участков, начиная с участка номер N1 и кончая участком номер N2. При повторном использовании той же программы можно ввести другие значения N1 и N2 и вывести на экран площади другой группы участков.

Элементы массива (переменные с индексом) можно использовать в программе так же, как простые переменные, например:

```
100 INPUT A(5), A(7), A(2)
110 LET X(I)=5.2
```

```
120 LET Z(4)=P
130 LET K(3)=N(6)
140 LET R=(A(4)—B)/(D(3)*P(2))
```

Значение индексов можно вычислять или вводить, например:

```
100 PRINT "УКАЖИТЕ НОМЕРА ИСХОДНОЙ И КОНЕЧНОЙ
    ТОЧЕК";
110 INPUT I, K
120 LET D1=X(K)—X(I)
130 LET D2=Y(K)—Y(I)
140 LET R=SQP(D1*D1+D2*D2)
150 PRINT "РАССТОЯНИЕ=" R " M"
```

Поскольку обработка массивов обычно осуществляется в теле цикла, рассмотрим некоторые распространенные методы такой обработки.

Присваивание элементам массива числовых значений в процессе ввода. Рассмотрим несколько примеров.

Пример 16. Присваивание вводимых значений всем элементам массива заранее известной длины:

```
10 DIM R(16)
20 FOR I=Ø TO 16
30 PRINT "R" I "=";
40 INPUT R(I)
50 NEXT I
```

Пример 17. Присваивание элементам одномерного массива некоторого, заранее не указанного в программе, количества вводимых значений. В этом случае объявляется массив с максимально возможным в данной задаче числом элементов (например, 100), а фактическое число вводимых элементов обозначается переменной (например, N) и вводится в процессе выполнения программы:

```
10 DIM F(100)
20 PRINT "N=";
30 INPUT N
40 FOR I=Ø TO N
50 PRINT "F" I "=";
60 INPUT F(I)
70 NEXT I
```

Пример 18. Присваивание вводимых значений элементам двумерного массива. Осуществляется с помощью вложенных циклов. Максимальные значения индексов так же, как и в случае одномерных массивов, могут быть известны заранее или вводиться в процессе выполнения программы. Например, ввод значений элементов матрицы:

$$\begin{matrix} A_{10} & A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{20} & A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ A_{N0} & A_{N1} & A_{N2} & A_{N3} & A_{N4} & A_{N5}, \end{matrix}$$

содержащей 6 столбцов (от Ø до 5) и заранее не указанное число строк N ($N \leqslant 20$). Нумерация строк начинается с 1.

Указанная матрица может быть представлена в программе двумерным массивом, объявляемым как DIM A(20, 5). Ввод можно осуществить с помощью операторов:

```
10 DIM A(20, 5)
20 PRINT "КОЛИЧЕСТВО СТРОК=";
30 INPUT N
40 FOR I=1 TO N
50 FOR J=Ø TO 5
60 PRINT "A" I; J "=";
70 INPUT A(I, J)
80 NEXT J
90 NEXT I
```

Пример 19. Присваивание вводимых значений последовательно расположенным элементам некоторой части массива, например, элементам одномерного массива P, начиная с P(5) и кончая P(12):

```
100 FOR I=5 TO 12
110 PRINT "P" I "="
120 INPUT P(I)
130 NEXT I
```

Аналогично осуществляется ввод и в тех случаях, когда участок массива заранее неизвестен, например:

```
100 PRINT "УКАЖИТЕ НОМЕРА НАЧАЛЬНОГО И КОНЕЧНОГО
    ЭЛЕМЕНТОВ"
110 INPUT N, M
120 FOR I=N TO M
130 PRINT "X" I "=";
140 INPUT X(I)
150 NEXT I
```

Пример 20. Присваивание вводимых значений элементам массива, индексы которых меняются по тому или иному правилу. Например, присвоить вводимые значения всем элементам массива с четными значениями индекса. Длина массива заранее неизвестна, но не более 200 элементов, начиная с Z (1):

```
10 DIM Z(200)
20 PRINT "N=";
30 INPUT N
40 FOR I=2 TO N STEP 2
50 PRINT "Z"I"=";
60 INPUT Z (I)
70 NEXT I
```

Ввод данных с магнитного диска. В тех версиях языка Бейсик, которые оперируют понятием файла, можно осуществить ввод данных с магнитного диска. Подлежащие вводу данные должны быть предва-

рительно подготовлены на диске, например, с помощью редактора текстов. Файл на диске, на котором размещены подлежащие вводу данные, необходимо в программе открыть с помощью оператора OPEN. Формат оператора:

OPEN «имя файла на диске» FOR INPUT AS FILE номер файла. Полностью имя файла на диске состоит из трех компонент: имени дисковода (DXØ:, DX1:, MXØ:, A:, B: и т. п.), имени файла и расширения имени файла. Перед расширением имени ставится точка. Пример имени файла, применительно к микроЭВМ семейства «Электроника 60»: MX1:MATR6. DAT. Пример оператора открытия файла на магнитном диске перед началом чтения данных:

OPEN "MX1:MATR6. DAT" FOR INPUT AS FILE#2.

Используя текстовую переменную, можно запрограммировать открытие файла с заранее не обусловленным именем:

```
10 PRINT "УКАЖИТЕ ИМЯ ФАЙЛА С ИСХОДНЫМИ ДАННЫМИ — ";
20 INPUT Q⊗
30 OPEN Q⊗ FOR INPUT AS FILE #1
```

Номер файла задается произвольно. Если в программе должно быть открыто одновременно несколько файлов, то их номера не должны совпадать.

Чтение данных из файла осуществляет оператор INPUT, в котором перед списком переменных указывается номер файла, например,

```
100 INPUT #2, X, Y, Z
```

Если в списке оператора INPUT содержится несколько переменных, то на магнитном диске должно быть такое же число данных, размещенных в одной строке через запятую, например: 2.7, 1.3, 5.6, или размещенных в столбик без запятой, например,

2.7
1.3
5.6 .

Следует иметь в виду, что при наличии в программе нескольких операторов INPUT или при многократном выполнении одного и того же оператора INPUT поиск данных при каждом выполнении этого оператора начинается с новой строки. Например, при наличии в программе участка

```
100 INPUT #3, X, Y
110 INPUT #3, K
120 INPUT #3, A, B, C
```

где переменные должны получить значения

X ←— 19.45, Y ←— 3.8, K ←— 12,
A ←— 2425, B ←— 347.8, C ←— 498.13,

исходные данные могут быть размещены следующим образом:

19.45, 3.8
12
2425, 347.8, 498.13 .

Они могут быть размещены и в столбик:

19.45
3.8
12
2425
и т. д.,

но не могут быть размещены, например, так:

19.45, 3.8, 12
2425, 347.8, 498.13 ,

потому что в этом случае X получит значение 19.45, Y — значение 3.8, K — значение 2425, так как оператор 110 INPUT #3, K начнет читать данное из новой строки. Оператор 120 INPUT #3, A, B, C будет искать несуществующую третью строку данных, что вызовет сообщение об ошибке и прерывание работы программы.

В силу указанного свойства оператора INPUT при вводе одномерного массива

```
100 FOR I=1 TO N
110 INPUT #5, X(I)
120 NEXT I
```

или двумерного массива (матрицы)

```
100 FOR I=1 TO N
110 FOR J=1 TO M
120 INPUT #4, X(I, J)
130 NEXT J
140 NEXT I
```

исходные данные должны быть расположены на магнитном диске в столбик.

Понятия о размещении данных на диске «в строчку» или «в столбик» не отражают реальную физическую картину. Но именно так выглядят данные на экране дисплея при их подготовке к записи на магнитный диск.

Рассмотрим два примера ввода массивов с магнитного диска. Как правило, программа предусматривает ввод массива произвольных размеров в пределах, допустимых данной версией языка и типом ЭВМ. В этом случае размеры массива (число элементов для одномерного или число строк и столбцов для двумерного) можно поместить на магнитном диске непосредственно перед данными.

Пример 21. Переписать с диска одномерный массив из N чисел. Количество чисел заранее не обусловлено и указано в начале файла, например,

5
9.4
8.3
5.6
0.95
3.33 .

Вариант участка программы, осуществляющего ввод:

```
10 DIM X(255)
20 PRINT "ИМЯ ФАЙЛА"
30 INPUT A⊗
40 OPEN A⊗ FOR INPUT AS FILE #5
50 INPUT #5, N
60 FOR I=1 TO N
70 INPUT #5, X(I)
80 NEXT I
90 CLOSE #5
```

Пример 22. Переписать с диска двумерный массив. Число строк и столбцов в массиве заранее не обусловлено и указано в начале файла, например, матрица

25.3	4.67	9.15	32.5
13.4	6.8	5.34	9.87
22.6	7.2	14.6	17.3

размещена на магнитном диске следующим образом:

```
3, 4
25.3
4.67
9.15
32.5
13.4
6.8
и т. д.
```

Вариант участка программы, осуществляющего ввод:

```
10 DIM X(100, 10)
20 PRINT "УКАЖИТЕ ИМЯ ФАЙЛА С ИСХОДНЫМИ ДАН-
   НЫМИ — ";
30 INPUT A⊗
40 OPEN A⊗ FOR INPUT AS FILE #8
50 INPUT #8, N, M
60 FOR I=1 TO N
70 FOR J=I TO M
80 INPUT #8, X(I, J)
90 NEXT J
100 NEXT I
110 CLOSE #8
```

Обратите внимание, что здесь число строк и столбцов (3, 4) размещено в файле на одной строке. Связано это с тем, что эти данные читаются одним оператором 50 INPUT #8, N, M.

Отмеченная выше особенность оператора INPUT особенно неудобна при подготовке к вводу матриц. Чтобы обойти указанный недостаток, можно каждую строку матрицы читать, как текстовую переменную, а затем программным путем посимвольно преобразовывать эту перемен-

ную в последовательность чисел. Запятая и пробел в строке символов могут быть использованы как признаки конца очередного числа. Для реализации описанного метода потребуются программные средства, которые в данной книге не рассматриваются.

Вывод числовых значений элементов массива организуется аналогично вводу. В приведенных выше программах оператор INPUT следует заменить оператором PRINT. При этом необходимо иметь в виду ограниченные размеры экрана и особенности использования разделителей — запятой и точки с запятой (см. с. 11).

Пример 23. Вывод всех значений элементов одномерного массива по одному числу в строке с указанием имени элемента:

```
100 FOR I=0 TO N
110 PRINT "Z"I"="Z(I)
120 NEXT I
```

Пример 24. Вывод части значений элементов одномерного массива в указанном диапазоне изменения индексов:

```
100 FOR I=M TO N
110 PRINT "Q"I"="Q(I),
120 NEXT I
130 PRINT
```

Здесь M и N соответственно начальное и конечное значения индексов, подлежащих выводу элементов. Эти значения могут вводиться в процессе выполнения программы, например:

```
80 PRINT "УКАЖИТЕ НОМЕРА НАЧАЛЬНОГО И КОНЕЧНОГО
   ЭЛЕМЕНТОВ"
90 INPUT M, N
```

или могут быть заранее известны, например:

```
100 FOR I=3 TO 25
```

Запятая в конце оператора 110 обеспечивает вывод очередного значения в следующую зону той же строки. После вывода пяти чисел в одну строку происходит автоматический переход к следующей строке. Если количество выводимых чисел не кратно 5, то последняя строка окажется заполненной не до конца. Оператор 130 необходим для того, чтобы в любом случае по окончании вывода массива происходил переход на новую строку. Если убрать запятую в конце оператора 110, то значения переменной Q(I) будут выводиться в столбик.

Вывод значений двумерного массива осуществляется с помощью вложенных циклов, аналогично вводу (см. пример 18). Рассмотрим несколько примеров вывода двумерных массивов.

Пример 25. Если массив отображает матрицу, состоящую не более чем из 20 строк и пяти столбцов, то при выводе эту матрицу можно полностью разместить на экране:

```
100 FOR I=1 TO N
110 FOR J=1 TO M
120 PRINT B(I, J),
```

```
130 NEXT J
140 IF M<5 THEN PRINT
150 NEXT I
```

Здесь $N \leqslant 20$ и $M \leqslant 5$. Запятая в конце оператора 120 обеспечивает вывод всех элементов одной строки матрицы на одну строку экрана. Оператор 140 обеспечивает переход на новую строку экрана после окончания вывода очередной строки матрицы.

Пример 26. Если матрица состоит более чем из пяти столбцов, то ее строка может не поместиться на одной строке экрана. В этом случае вывод можно осуществить построчно, указывая номер строки и приостанавливая выполнение программы, чтобы дать возможность строку прочесть или переписать:

```
100 FOR I=Ø TO M
110 PRINT "СТРОКА" I
120 FOR J=Ø TO N
130 PRINT S(I, J),
140 NEXT J
150 PRINT
160 PRINT "ПЕРЕПИШИТЕ И НАЖМИТЕ КЛАВИШУ ВК"
170 INPUT Z⊗
180 NEXT I
```

Приостановка программы происходит в момент выполнения оператора 170. При нажатии на клавишу ВК переменная Z⊗ получит значение пробела и выполнение программы продолжится. (В некоторых версиях языка перед нажатием на клавишу ВК необходимо ввести на экран пробел или другой символ). Если площади экрана хватает для размещения матрицы, а также в том случае, когда вывод осуществляется на печать, операторы 160 и 170 не нужны. Можно увеличить плотность размещения чисел на экране, заменив в конце оператора 130 запятую на точку с запятой.

Пример 27. Можно выводить на экран участок матрицы («окно»), указав координаты «северо-западного» угла этого участка:

```
 90 PRINT "УКАЖИТЕ КООРДИНАТЫ СЕВ-ЗАП УГЛА:"
100 PRINT "НОМЕР ЛЕВОГО СТОЛБЦА=";
110 INPUT L
120 PRINT "НОМЕР ВЕРХНЕЙ СТРОКИ=";
130 INPUT V
140 LET L1=L+4
150 IF L1>M THEN LET L1=M
160 LET V1=V+17
170 IF V1>N THEN LET V1=N
180 FOR I=V TO V1
190 FOR J=L TO L1
200 PRINT X(I, J),
210 NEXT J
220 IF L1-L<4 THEN PRINT
230 NEXT I
```

Здесь на экран выводится участок матрицы, содержащий 5 столбцов и 18 строк. Полные размеры матрицы: M столбцов и N строк. Если от указанного северо-западного угла до границы матрицы остается меньше 5 столбцов или 18 строк, то размеры окна, соответственно, уменьшаются (операторы 150 и 170).

Пример 28. На печать матрицу M×N можно вывести по частям (по N строк и 5 столбцов в каждой части):

```
100 OPEN "P:" FOR OUTPUT AS FILE #1
110 FOR L=1 TO M STEP 5
120 K=L+4
130 IF K>M THEN K=M
140 FOR I=1 TO N
150 FOR J=L TO K
160 PRINT #1, X(I, J),
170 NEXT J
180 IF K-L<4 THEN PRINT #1
190 NEXT I
200 PRINT #1
220 NEXT L
230 CLOSE #1.
```

Здесь операторы 130 и 180 обеспечивают работоспособность программы, когда общее число столбцов не кратно 5.

Присваивание элементам массива заданных или вычисленных значений. Техника такого присваивания не отличается от техники ввода или вывода значений элементов массива. Различие заключается лишь в том, что в теле цикла вместо операторов PRINT и INPUT используются операторы присваивания. Типичным случаем является присваивание всем элементам массива нулевых значений.

Пример 29. Присвоить нулевые значения всем элементам двумерного массива размером M×N:

```
100 FOR I=1 TO N
110 FOR J=1 TO M
120 LET X(I, J)=Ø
130 NEXT J
140 NEXT I
```

Присваивание всем элементам массива нулевых значений иногда называют «очисткой» массива. Очистка массива бывает необходима в том случае, когда не обязательно все его элементы получают числовые значения при выполнении программы, а использованными в дальнейших вычислениях или при выводе могут оказаться все элементы. Кроме того, массив следует очищать в тех случаях, когда в дальнейшем в элементах массива накапливаются суммы или произведения.

Пример 30. Можно создать в программе модель экрана в виде двумерного массива текстовых переменных:

```
10 DIM E⊗(18, 70)
```

присвоить всем элементам этого массива значения пробелов:

```
20 FOR I=1 TO 18
30 FOR J=1 TO 70
40 E⊗(I, J)="␣"
50 NEXT J
60 NEXT I
```

присвоить некоторым элементам массива значения заданного символа (+ * и т. п.) и вывести на экран рисунок:

```
200 FOR I=1 TO 18
210 FOR J=1 TO 70
220 PRINT E⊗(I, J);
230 NEXT J
240 PRINT
250 NEXT I
```

Характер рисунка определяется алгоритмом выбора элементов массива, которым присваиваются значения символов. Например, поместив между строками 60 и 200 группу операторов

```
 70 FOR I=1 TO 18
 80 LET E⊗(I, 19−I)="+"
 90 NEXT I
100 FOR I=2 TO 18
110 E⊗(I, 18)="*"
120 NEXT I
130 FOR J=2 TO 17
140 E⊗(18, J)="−"
150 NEXT J
```

мы получим программу, которая выводит на экран прямоугольный треугольник (рис. 13).

В приведенной выше программе операторы 70—90 заносят в матрицу экрана символы "+", которые при выводе на экран образуют наклонную линию (гипотенузу треугольника). Операторы 10—120 и 130—150 заносят в матрицу экрана символы * и "−", которые при выводе на экран образуют, соответственно, вертикальную и горизон-

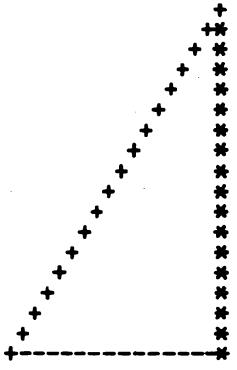


Рис. 13. Контур, выведенный на экран алфавитно-цифрового дисплея или на печать

тальную линии (катеты треугольника). Здесь мы имеем дело с программой, которая рисует всегда один и тот же контур.

Если группу операторов 70—150 заменить блоком, который по тем или иным формулам вычисляет координаты заполняемых символами элементов матрицы экрана, то мы получим программу, которая будет выводить на экран различные варианты фигур одного и того же класса. Например, используя уравнение прямой, проведенной через две точки с заданными координатами, можно создать программу, которая будет выводить на экран контур многоугольника. Разработка подобных программ является достаточно трудоемкой и в данной книге не рассматривается.

В настоящее время персональные ЭВМ обеспечивают программный доступ к отдельной точке растра экрана (имеют графический дисплей). Принципы получения изображения на экране графического дисплея мало отличаются от описанного в данном примере, но разрешающая способность экрана значительно выше. Существующие системы программирования для персональных ЭВМ располагают достаточно широким набором средств для вывода на экран графической информации.

Вычисление сумм и произведений элементов массива. Методика суммирования последовательности вводимых чисел была рассмотрена в предыдущих разделах (см. примеры 9 и 10). Аналогично можно вычислить произведение последовательности вводимых чисел, но в этом случае переменная, накапливающая произведения, должна предварительно получить значение 1, например:

```
100 LET P=1
110 FOR I=1 TO N
120 INPUT X
130 LET P=P*X
140 NEXT I
```

В рассмотренных ситуациях введенное число подключается к сумме или произведению, после чего значение этого числа теряется, так как та же переменная X получает значение следующего вводимого числа.

Значительно чаще приходится суммировать или перемножать числа не непосредственно в процессе ввода, а после того, как значения этих чисел присвоены элементам массива. Рассмотрим примеры.

Пример 31. Вычисление среднего значения

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i,$$

где x_i — элементы массива X:

```
100 LET S=0
110 FOR I=1 TO N
120 LET S=S+X(I)
130 NEXT I
140 LET C=S/N
```

Здесь $\bar{x}$ обозначен переменной C.

Пример 32. Вычисление произведения значений элементов массива A:

```
100 LET D=1
110 FOR I=1 TO N
120 LET D=D*A(I)
130 NEXT I
```

Пример 33. Написать программу вычисления и вывода величин

$$d_i = \bar{x} - x_i,$$

где

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \quad (N \leqslant 255).$$

Здесь значения x_i понадобятся дважды: вначале для вычисления $\bar{x}$, а затем для вычисления d_i. Отказ от использования массива и суммирование x_i непосредственно в процессе ввода приведет в этом случае к необходимости дважды вводить одни и те же исходные данные. Вариант программы с использованием массива:

```
 10 DIM X(255)
 20 PRINT "ВВЕСТИ N"
 30 INPUT N
 40 FOR I=1 TO N
 50 PRINT "ВВЕСТИ X" I;
 60 INPUT X(I)
 70 NEXT I
 80 LET S=0
 90 FOR I=1 TO N
100 LET S=S+X(I)
110 NEXT I
120 LET A=S/N
130 FOR I=1 TO N
140 LET D=A-X(I)
150 PRINT "D"I"="D
160 NEXT I
170 END
```

Здесь вычисленные значения d_i выводятся сразу по мере счета и в дальнейшем в программе не используются. Поэтому нет надобности в массиве D. Значение очередного результата присваивается простой переменной D (оператор 140) и тут же выводится (оператор 150). После этого та же переменная D получает значение следующего результата и т. д.

Отбор и сортировка данных в массиве. Широко распространены задачи, в которых характер использования переменной зависит либо от значения ее индекса, либо от значения самой переменной. При программировании подобных задач иногда приходится выбирать шаг из-

менения индекса, отличный от 1, выбирать индексные выражения в соответствии с условием задачи, включать в тело цикла разветвляющиеся структуры. Рассмотрим несколько примеров.

Пример 34. Вычислить отдельно средние значения переменных с нечетными и четными значениями индексов:

$$\bar{x}_{\text{нч}} = \frac{2}{N} \sum_{i=1}^{N-1} x_i \quad (i \text{ — нечетные});$$

$$\bar{x}_{\text{чт}} = \frac{2}{N} \sum_{i=2}^{N} x_i \quad (i \text{ — четные}),$$

$N \leqslant 240$ (обязательно четное).

В программе придется вычислять две суммы. Обозначим их:

S1 — для переменных с нечетными индексами и S2 — для переменных с четными индексами. Средние значения обозначим соответственно C1 и C2.

Вариант программы:

```
10 DIM X (240)
20 PRINT "ВВЕСТИ N"
30 INPUT N
40 FOR I=1 TO N
50 PRINT "ВВЕСТИ X" I;
60 INPUT X(I)
70 NEXT I
80 LET S1=0
90 LET S2=0
100 FOR I=1 TO N-1 STEP 2
110 LET S1=S1+X(I)
120 LET S2=S2+X(I+1)
130 NEXT I
140 LET C1=2/N*S1
150 LET C2=2/N*S2
160 PRINT "СРЕДНЕЕ НЕЧЕТНЫХ="C1
170 PRINT "СРЕДНЕЕ ЧЕТНЫХ=" C2
180 END
```

Здесь при вычислении сумм индекс получает с помощью оператора 100 только нечетные значения (1, 3, 5 и т. д.). Оператор 110 накапливает сумму значений переменных с этим значением индекса. В операторе 120 переменная X(I+1) содержит индексное выражение (I+1), принимающее только четные значения, благодаря чему накапливается сумма значений переменных с четным значением индекса.

Пример 35. Вычислить средние значения положительных и отрицательных элементов массива из N чисел ($N \leqslant 120$). В программе предусмотреть возможность наличия в массиве нулевых значений.

Дадим массиву имя X. Искомые средние обозначим C1 (для $X > 0$) и C2 (для $X < 0$). Формулы для вычисления средних значений:

C1 = S1/N1 и C2 = S2/N2,

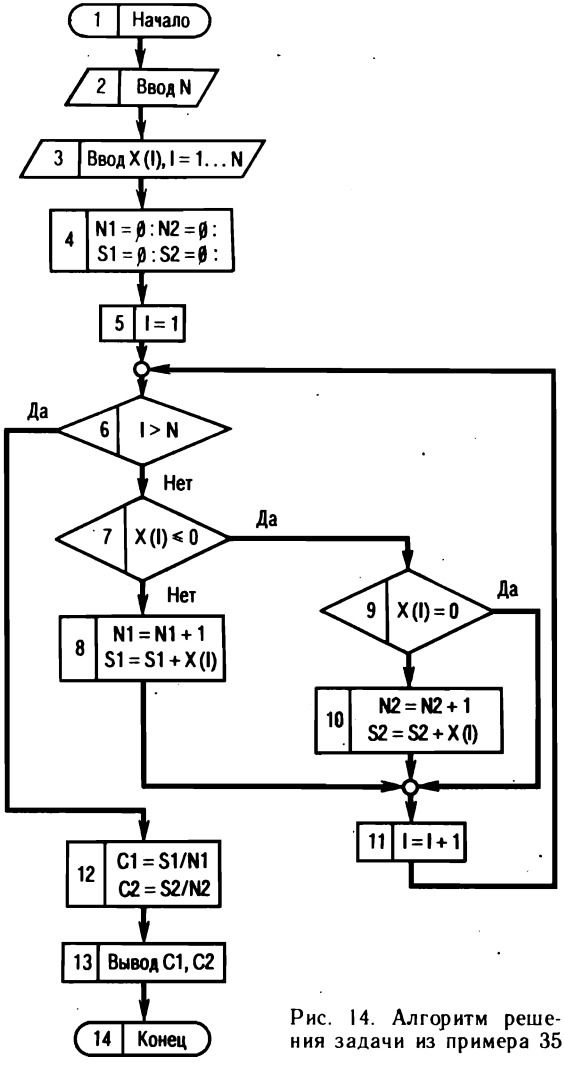


Рис. 14. Алгоритм решения задачи из примера 35

где S1, N1 — сумма и количество положительных X; S2, N2 — сумма и количество отрицательных X.

В отличие от задач из предыдущих примеров алгоритм решения данной задачи будет содержать разветвляющиеся структуры, помещенные в тело циклической структуры. Циклическая структура должна обеспечить просмотр всех X от X_1 до X_N, а разветвляющиеся структуры должны обеспечить раздельное накопление S1, N1 и S2, N2 по мере просмотра массива.

Накопление сумм можно осуществить с помощью операторов LET S1 = S1 + X(I) для X(I) > 0 и LET S2 = S2 + X(I) для X(I) < 0.

Количество положительных и отрицательных X можно накапливать с помощью операторов LET N1=N1+1 для X(I)>0 и LET N2=N2+1 для X(I)<0. Алгоритм решения задачи приведен на рис. 14. Он может быть реализован с помощью следующей программы.

Программа:

```
 10 DIM X(120)                                          Бл. 1
 20 PRINT "N="
 30 INPUT N                                             Бл. 2
 40 FOR I=1 TO N
 50 PRINT "X" I "=";
 60 INPUT X(I)                                          Бл. 3
 70 NEXT I
 80 LET N1=0: LET N2=0
 90 LET S1=0: LET S2=0                                  Бл. 4
100 FOR I=1 TO N                                        Бл. 5, 6
110 IF X(I)<=0 THEN 150                                 Бл. 7
120 LET N1=N1+1
130 LET S1=S1+X(I)                                      Бл. 8
140 GOTO 180
150 IF X(I)=0 THEN 180                                  Бл. 9
160 LET N2=N2+1
170 LET S2=S2+X(I)                                      Бл. 10
180 NEXT I                                              Бл. 11
190 LET C1=S1/N1
200 LET C2=S2/N2                                        Бл. 12
210 PRINT "СРЕДНЕЕ ПОЛОЖИТ. ЧИСЕЛ=" C1
220 PRINT "СРЕДНЕЕ ОТРИЦ. ЧИСЕЛ=" C2                   Бл. 13
230 END                                                 Бл. 14
```

В программе указано соответствие между ее операторами и блоками алгоритма.

Приведенный алгоритм применим только к таким массивам, которые обязательно содержат и отрицательные и положительные числа ($N1 \neq 0$ и $N2 \neq 0$). Если положительных или отрицательных чисел в массиве нет ($N1=0$ или $N2=0$), то при выполнении операторов 190 или 200 будет выдано сообщение об ошибке. Сделать алгоритм результативным при работе с любыми массивами можно, заменив блок 12 (см. рис. 14) последовательностью из двух структур «ЕСЛИ — ТО — ИНАЧЕ» (рис. 15). В программе эти структуры можно реализовать следующей группой операторов:

```
185 IF N1<>0 THEN 190                                   Бл. 12—1
186 PRINT "ПОЛОЖИТЕЛЬНЫХ ЧИСЕЛ
    В МАССИВЕ НЕТ"                                      Бл. 13—1
187 GOTO 210
190 LET C1=S1/N1                                        Бл. 12—2
200 PRINT "СРЕДНЕЕ ПОЛОЖИТЕЛЬНЫХ
    ЧИСЕЛ=" C1                                          Бл. 13—2
```

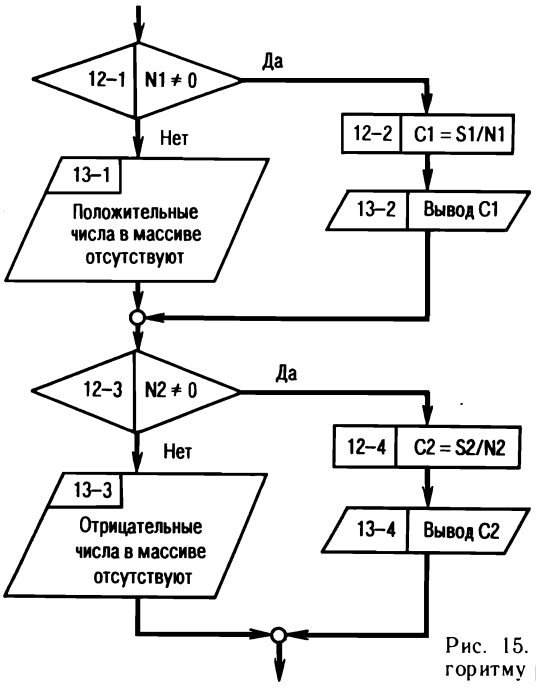


Рис. 15. Дополнительные блоки к алгоритму решения задачи из примера 35

```
210 IF N2< >0 THEN 215                          Бл. 12—3
211 PRINT "ОТРИЦАТЕЛЬНЫХ ЧИСЕЛ
    В МАССИВЕ НЕТ"                              Бл. 13—3
212 GOTO 230
215 LET S2=S2/N2                                Бл. 12—4
220 PRINT "СРЕДНЕЕ ОТРИЦАТЕЛЬНЫХ
    ЧИСЕЛ=" C2                                  Бл. 13—4
```

Пример 36. Поиск максимального или минимального значения в массиве. Такой поиск осуществляется обычно методом перебора. Алгоритм этого метода в случае поиска максимального значения приведен на рис. 16. Здесь переменная M вначале получает значение первого элемента массива, а затем поочередно сравнивается с остальными элементами. При появлении элемента, превосходящего M, переменная M получит значение этого элемента. В результате после сравнения со всеми элементами массива переменная M будет иметь значение максимального из них. Приведенный алгоритм можно реализовать следующей последовательностью операторов:

```
100 LET M=X(1)
110 FOR I=2 TO N
120 IF M<=X(I) THEN 140
130 LET M=X(I)
140 NEXT I
150 PRINT M
```

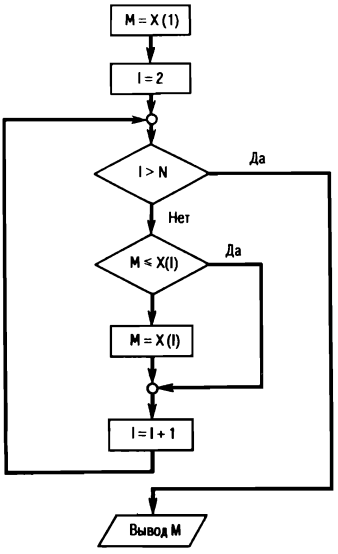


Рис. 16. Алгоритм поиска максимального значения в массиве

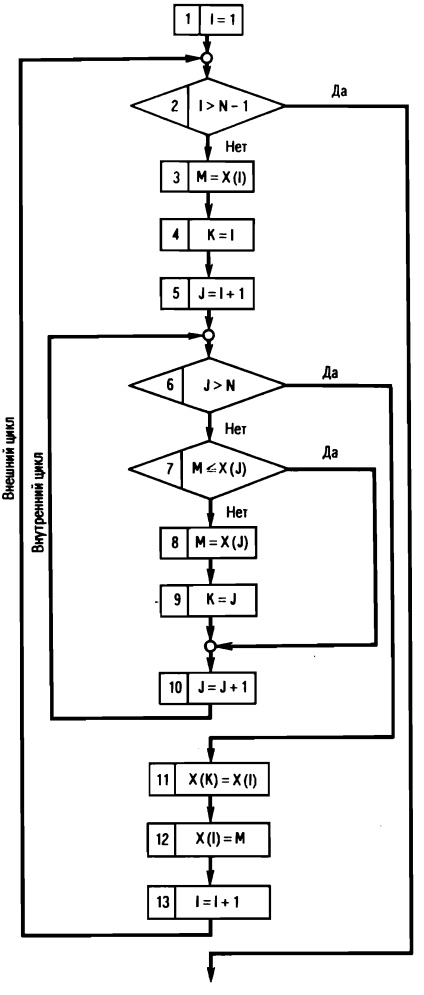


Рис. 17. Алгоритм упорядочения элементов массива по убыванию

Если помимо значения максимального элемента массива нужно определить еще и номер (индекс) этого элемента, то приведенную последовательность необходимо дополнить двумя операторами:

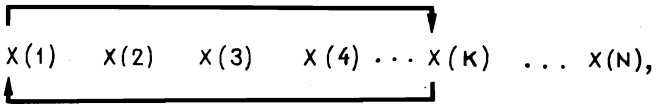
```
105 LET K=I
135 LET K=I    .
```

По окончании перебора переменная K будет иметь значение индекса максимального элемента.

Алгоритм поиска минимального элемента аналогичен. Изменяется лишь характер сравнения M с X(I).

Пример 37. Упорядочивание элементов массива по возрастанию или убыванию их значений сводится к перестановке элементов массива таким образом, чтобы каждый следующий элемент был больше (при

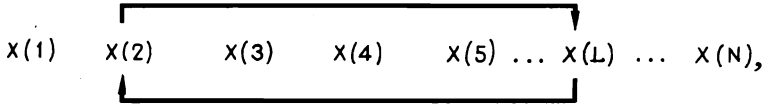
упорядочении по возрастанию) или меньше (при упорядочении по убыванию), чем предыдущий. Существует несколько алгоритмов упорядочения. Рассмотрим один из них (рис. 17). В соответствии с этим алгоритмом вначале просматривается весь массив, начиная с X(1), отыскивается максимальное значение и переменная, обладающая этим значением, меняется местами с первой переменной:



здесь X(K) — максимальное значение.

В результате первое место в массиве занимает переменная, обладающая максимальным значением.

Затем то же самое проделывается с элементами массива, начиная с X(2):



где X(L) — максимальное значение.

Затем просматривается массив, начиная с X(3), далее — начиная с X(4), и т. д. После каждого просмотра в начало просматриваемой части массива перемещается переменная, обладающая максимальным значением. В результате элементы массива размещаются по убыванию. Для упорядочивания массива по возрастанию при каждом просмотре необходимо отыскивать минимальный элемент.

Алгоритм реализован двумя вложенными циклами. Внешний цикл меняет значение начального индекса просматриваемой части массива (блоки 1, 2, 13, в программе — операторы 100 FOR I=1 TO N — 1 и 200 NEXT I). Внутренний цикл осуществляет поиск максимального значения среди элементов массива, начиная с X(I), где I задается внешним циклом, и, кончая X(N) (блоки 3—10). Блоки 11—12 осуществляют перемещение элемента с максимальным значением в начало просматриваемого участка. Программа, реализующая данный алгоритм, может иметь следующий вид:

```
100 FOR I=1 TO N-1
110 LET M=X(I)
120 LET K=I
130 FOR J=I+1 TO N
140 IF M<=X(J) THEN 170
150 LET M=X(J)
160 LET K=J
170 NEXT J
180 LET X(K)=X(I)
190 LET X(I)=M
200 NEXT I
```

1.8. Функции пользователя

Помимо встроенных функций, перечисленных в табл. 3, в языке Бейсик допустимы, так называемые, функции пользователя. В отличие от встроенной функции, функция пользователя должна быть предварительно определена с помощью оператора DEF (от слова definition — определение). Формат оператора:

DEF FN буква (формальный параметр) = выражение.

Например, DEF FNR(X) =(2 * X + 3.5) * X + 1.7. Часто вместо слов «определение функции» употребляют выражение «описание функции».

Буква после FN (в данном случае буква R) служит для идентификации функции пользователя — для отличия одной функции от другой. Формальный параметр — это переменная, (в данном случае переменная X). Формальный параметр может входить также в состав выражения, расположенного справа от знака "=". В соответствии с этим выражением происходит вычисление функции. Приведенный выше оператор описывает функцию вычисления полинома 2-й степени по формуле Горнера

$$2x^2 + 3.5x + 1.7 = (2x + 3{,}5)\,x + 1.7.$$

Функция пользователя применяется в выражениях так же, как встроенная функция: на место формального параметра помещается фактический параметр и функция вычисляется с помощью выражения, указанного в ее описании. Например, если программа содержит приведенное выше описание функции, то при выполнении оператора

LET Z = 3 * FNR(5)

вначале будет вычислено значение функции
FNR(5)=(2 * 5 + 3.5) * 5 + 1.7 = 69.2, а затем значение переменной Z = = 3 * 69.2 = 207.6.

Обычно функции пользователя применяют в тех случаях, когда в программе необходимо несколько раз выполнять вычисления по одной и той же формуле. Например, необходимо вычислить сумму полиномов, описанных функцией FNR при X = 3, X = 5 и X = 10.

Если программа содержит приведенный выше оператор DEF, то указанную сумму можно вычислить с помощью оператора

LET S = FNR(3) + FNR(5) + FNR(10).

В качестве фактического параметра можно использовать число (как в приведенном выше примере), переменную или выражение, в том числе и выражение, содержащее функцию, например:

```
100 INPUT A
110 LET Q=FNR(A)*FNR(2*SQR(A+3)+A)
```

Формальный параметр не обязательно должен входить в состав выражения, описывающего функцию. Например, функцию перевода угла из градусной меры в радианную можно описать следующим образом:

DEF FNM(X) = ((S/60 + M)/60 + G) * 0.01745,

где 0.01745 = π/180).

В этом случае все переменные, которые входят в состав описания функции (кроме формального параметра), должны получить в программе необходимые числовые значения. Только после этого можно обратиться к функции, указав в качестве фактического параметра любое число. Например, прежде чем обратиться к функции FNM, необходимо ввести (или вычислить и присвоить) значения переменным G, M, S:

```
100 INPUT G, M, S
110 LET A=FNM(Ø)  или  110 LET P=COS(FNM(Ø)).
```

Здесь в качестве значения фактического параметра мы использовали число Ø.

Пример 38. Рассмотрим программу, которая по заданной длине одной из сторон треугольного участка (L) в метрах и значениям прилегающих к этой стороне углов (α, β) в градусах, минутах и секундах, вычисляет площадь участка в гектарах по формуле

$$P=\frac{L^2 \sin\alpha \sin\beta}{20000 \sin(\alpha+\beta)}$$

и длины двух других его сторон

$$A_1=\frac{L \sin\alpha}{\sin(\alpha+\beta)} \text{ и } A_2=\frac{L \sin\beta}{\sin(\alpha+\beta)}.$$

Перед выводом результатов площадь округлим до 0,01 га, а длины сторон до 0,01 м.

В программе два раза придется осуществлять перевод угла из градусной меры в радианную и три раза округлять результат до 0,01. Для перевода угла применим функцию пользователя

```
FNR (X) = ((S/60+M)/60+G)*0.01745
```

и для округления (см. с. 18) — функцию пользователя

```
FNO (X) = INT (X*100+0,5)/100.
```

Вариант программы:

```
10  DEF FNR (X) = ((S/60+M)/60+G)*0.01745
20  DEF FNO (X) = INT (X*100+0.5)/100
30  PRINT "СТОРОНА L=";
40  INPUT L
50  PRINT "УГОЛ A=";
60  INPUT G, M, S
70  LET A=FNR(∅)
80  PRINT "УГОЛ B=";
90  INPUT G, M, S
100 LET B=FNR (Ø)
110 LET S1=SIN (A)
120 LET S2=SIN (B)
130 LET S3=SIN (A+B)
140 LET P=L*L*S1*S2/20000/S3
150 LET P=FNO (P)
```

```
160 LET A1=FNO (L*S1/S3)
170 LET A2=FNO (L*S2/S3)
180 PRINT "ПЛОЩАДЬ="P" ГА"
190 PRINT "СТОРОНЫ", "A1="A1"M", "A2="A2"M"
200 END
```

В программе функция FNR (X) не зависит от параметра X. Поэтому при использовании этой функции (строки 70 и 100) в качестве фактического параметра можно было использовать любое число (в программе использовано число Ø). Переменные G, M, S, которые используются при вычислении этой функции, получают свои значения с помощью операторов ввода (строки 60 и 90).

Функция FNO (X) зависит от параметра X. В первый раз (строка 150) в качестве фактического параметра используется вычисленная в строке 140 площадь, обозначенная переменной P. Результат округления присваивается той же переменной P. Во второй раз (строка 160) в качестве фактического параметра используется арифметическое выражение L*S1/S3, значение которого равно длине одной из сторон. Аналогично используется функция FNO (X) в третий раз (строка 170).

Можно описать функцию пользователя, которая зависит как от формального параметра, так и от других параметров. Например, функция, описываемая оператором

```
DEF FNP (X)=INT (X*N+0.5)/N
```

округляет значение переменной X с точностью, которая задается величиной N: при N=10 — округляем до 0.1, при N=1000 — округляем до 0,001 и т. д. Очевидно, что переменная N должна получить значение до обращения к функции FNP, например:

```
100 LET X=D*COS (R)
110 LET N=100
120 LET Z=FNP (X)
```

Здесь переменная Z получает значение переменной X, округленное до сотых.

Оператор DEF, как и все другие операторы, не может занимать более одной строки. Сложные функции можно описывать по частям, используя в последующих описаниях функции, описанные выше. Например, функцию

$$f(x)=\frac{A\sin x+B\cos x}{C/\sin x+D/\cos x}$$

можно описать тремя операторами:

```
10 DEF FNA (X)=A*SIN (X)+B*COS(X)
20 DEF FNB (X)=C/SIN (X)+D/COS(X)
30 DEF FNP (X)=FNA (X)/FNB (X)
```

Функции языка Бейсик, в том числе и функции пользователя, допускают рекурсивное использование (использование некоторой функции в качестве фактического параметра той же функции), например:

```
10 DEF FNF (X) = X*X
20 LET P=FNF (FNF (5))
30 LET R=FNF (1Ø+2*FNF (3))
```

Здесь оператор 20 вычисляет P = (5*5)*(5*5), оператор 30 вычисляет R = (10+2*3*3)*(10+2*3*3).

1.9. Подпрограммы

Функции используются в тех случаях, когда в различных местах программы необходимо вычислять одно и то же выражение при различных значениях, входящих в это выражение переменных. Однако с помощью функции нельзя описать вычислительную процедуру, результатами которой являются одновременно значения нескольких переменных. Например, нельзя воспользоваться функцией для решения системы уравнений, так как при этом вычисляется несколько корней, нельзя воспользоваться функцией для перевода угла из радиан в градусы, минуты и секунды, так как здесь результат представлен в виде трех чисел. Функцией нельзя также воспользоваться для вывода результатов. Во всех перечисленных ситуациях, в случае необходимости, можно воспользоваться подпрограммой.

Подпрограмма в языке Бейсик — это группа операторов, в конце которой расположен оператор RETURN (возвращать). Например, подпрограмма решения системы из двух линейных алгебраических уравнений

$$a_1x_1 + a_2x_2 = a_0;$$

$$b_1x_1 + b_2x_2 = b_0$$

по формулам Крамера

$$x_1 = D_1/D;\ x_2 = D_2/D,$$

где

$$D = a_1b_2 - a_2b_1;$$

$$D_1 = a_0b_2 - a_2b_0;$$

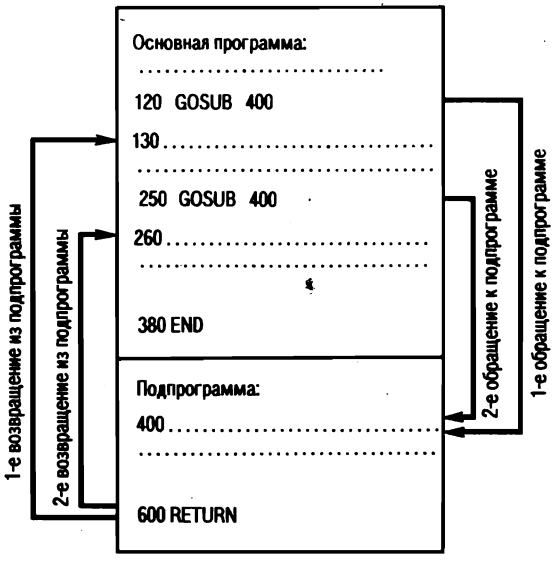
$$D_2 = a_1b_0 - a_0b_1,$$

может иметь вид:

```
200 LET D=A1*B2-A2*B1
210 IF ABS (D)>1E-10 THEN 250
220 PRINT "ОПРЕДЕЛИТЕЛЬ РАВЕН НУЛЮ"
230 LET Z=1
240 GOTO 280
250 LET X1=(AØ*B2-A2*BØ)/D
260 LET X2=(A1*BØ-AØ*B1)/D
270 LET Z=Ø
280 RETURN
```

Здесь переменная Z — признак результата. Если после выполнения

Рис. 18. Схема обращения к подпрограмме



подпрограммы $Z=\emptyset$, это значит, что система имеет решение и ее корни равны X1, X2. Если $Z=1$, то определитель системы близок к нулю и система не имеет решения.

Для обращения к подпрограмме используется оператор GOSUB (от слов go subroutine — перейти к подпрограмме). Формат этого оператора: GOSUB номер первой строки подпрограммы. Например, чтобы обратиться к приведенной выше подпрограмме, потребуется оператор GOSUB 200.

В дальнейшем программу, из которой происходит обращение к подпрограмме (вызов подпрограммы) будем называть основной или вызывающей программой.

Оператор RETURN, помещенный в конце программы, обеспечивает возвращение из нее в основную программу к строке, расположенной после строки с оператором GOSUB. Например, если в основной программе имеются операторы

```
30 GOSUB 200
40 IF Z=Ø THEN 70
50 PRINT "РЕШЕНИЯ НЕТ"
60 GOTO 80
70 LET R= (X1+X2)/2
80
```

то после выполнения подпрограммы будет выполняться оператор 40.

Пример использования подпрограмм приведен схематически на рис. 18.

Те переменные, значения которых используются при выполнении подпрограмм, обычно называют входными параметрами. Перед обращением к подпрограмме ее входные параметры должны получить числовые

значения. Например, до обращения к приведенной выше подпрограмме ее параметры AØ, A1, A2, BØ, B1, B2 должны получить числовые значения.

Те переменные, значения которых вычисляются подпрограммой, а затем используются в основной программе, называют выходными параметрами. В рассмотренной выше подпрограмме выходными параметрами являются переменные X1, X2 и Z.

Рассмотрим примеры программ, содержащих обращение к подпрограммам.

Пример 39. Написать программу для вычисления величин

$$Z1=(X1+Y1)/2;\ Z2=(X2+Y2)/2,$$

где $X1$, $X2$ — корни системы линейных уравнений:

$$a_1x_1+a_2x_2=a_0;$$

$$b_1x_1+b_2x_2=b_0,$$

а $Y1$, $Y2$ — корни системы:

$$a_1y_1+(a_2+r_1)y_2=a_0+r_2;$$

$$b_1y_1+(b_2+r_3)y_2=b_0+r_4.$$

Если, по крайней мере, одна из систем не имеет решения, то необходимо вывести на экран сообщение "РЕШЕНИЯ НЕТ". Для решения систем использовать приведенную выше подпрограмму.

Программу решения этой задачи можно оформить следующим образом:

```
5   PRINT "A1, A2, AØ, B1, B2, BØ=";
10  INPUT A1, A2, AØ, B1, B2, BØ
20  GOSUB 200
30  IF Z=1 THEN 170
40  LET S1=X1
50  LET S2=X2
55  PRINT "R1, R2, R3, R4=";
60  INPUT R1, R2, R3, R4
70  LET A2=A2+R1
80  LET AØ=AØ+R2
90  LET B2=B2+R3
100 LET BØ=BØ+R4
110 GOSUB 200
120 IF Z=1 THEN 170
130 LET Z1=(S1+X1)/2
140 LET Z2=(S2+X2)/2
150 PRINT Z1, Z2
160 GOTO 180
170 PRINT "РЕШЕНИЯ НЕТ"
180 END
200 Подпрограмма решения системы линейных
... уравнений.
280
```

Здесь обращение к подпрограмме происходит дважды. Перед первым обращением входные параметры подпрограммы получают числовые значения в результате выполнения оператора 10.

Значения корней X1, X2 после первого выполнения подпрограммы присваиваются переменным S1, S2. Перед вторым обращением к подпрограмме параметры A1, B1 сохраняют свои прежние значения (см. условие задачи). Параметры A2, AØ, B2, BØ получают новые значения с помощью операторов 60—100. Значение корней $X1$, $X2$ после второго обращения к подпрограмме эквивалентны переменным $Y1$, $Y2$ в условии задачи. Соответственно они используются в операторах 130, 140 для вычисления $Z1$ и $Z2$.

Пример 40. Написать программу определения острых углов прямоугольного треугольника α, β по заданным значениям его катетов A и B. Значения углов выводить на экран в градусах, минутах и секундах.

Углы α и β можно вычислить по формулам

$\alpha = \operatorname{arctg}(A/B)$; $\beta = 90° - \alpha$.

Для перевода этих углов в градусы, минуты и секунды можно воспользоваться алгоритмом, описанным на с. 19 и 25. В целом, программа может иметь следующий вид:

```
10  INPUT A, B
20  LET K=180/3. 141593
30  LET G1=ATN (A/B)*K
40  PRINT "АЛЬФА=";
50  GOSUB 100
60  LET G1=90-G1
70  PRINT "БЕТА=";
80  GOSUB 100
90  END
100 LET G=INT (G1)
110 LET M1=(G1-G)*60
120 LET M=INT (M1)
130 LET S=(M1-M)*60
140 LET S=INT (S+0.5)
150 IF S<60 THEN 210
160 LET S=Ø
170 LET M=M+1
180 IF M<60 THEN 210
190 LET M=Ø
200 LET G=G+1
210 PRINT G ; M; S
220 RETURN
```

Здесь в основной программе вычисляются значения углов в градусах с десятичной дробной частью и выводятся на экран наименования этих углов. Перевод углов в градусы, минуты и секунды, а также вывод этих значений на экран осуществляет подпрограмма, расположенная в строках 100—220.

Пример использования подпрограммы имеется и в разделе 1.6 (пример 12), где подпрограмма используется для вывода на экран или печать горизонтальной линии, состоящей из знаков "—". Подпрограмма расположена в строках 400—440 и имеет параметр K, значение которого определяет количество вводимых знаков "—" (длину линии). В данном примере параметр K получает значение 70 (операторы 90 и 220), когда нужно провести линию через всю таблицу, и значение 56 (оператор 150), когда нужно провести линию через все столбцы, кроме первого, в который оператор 140 выводит слова "БЕТА".

2. ПРОГРАММИРОВАНИЕ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ОБРАБОТКИ СЕТЕЙ ПЛАНОВОГО ОБОСНОВАНИЯ

Математическую обработку результатов геодезических измерений в сетях планового обоснования можно подразделить на два этапа — предварительные и окончательные вычисления.

Основной задачей предварительных вычислений является приведение измеренных направлений или углов к центрам пунктов. Эта задача может возникнуть при любом методе построения сетей планового обоснования. На первом этапе могут возникнуть и другие задачи, без решения которых невозможно выполнение второго этапа. Такими задачами являются следующие:

определение расстояний между смежными пунктами сети;

вычисление углов треугольника по измеренным сторонам;

передача координат пункта с вершины знака на землю в случае невозможности определения примычного угла при привязке сети;

перевычисление координат пунктов из системы в систему;

обеспечение данными для ориентирования и масштабирования несвободной сети;

определение азимута по результатам астрономических наблюдений для ориентирования свободной (автономной) сети.

2.1. Предварительные вычисления

Для решения основной задачи предварительных вычислений (определения поправок за центрировку и редукцию) требуются исходные данные: линейные и угловые элементы приведения измеренных направлений или углов к центрам пунктов (рис. 19), а также приближенные длины сторон до смежных пунктов. В сетях триангуляции приближенные длины сторон находят из предварительного решения треугольников.

2.1.1. Определение длин сторон треугольников

Для решения этой задачи используется программа «Решение треугольников». Программа работает в двух режимах: «Вычисление сторон» и «Определение углов». В данном случае используется первый режим. В этом режиме можно вычислить приближенные длины сторон треугольников сети по теореме синусов

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}.$$

В треугольниках сети угол A расположен против определяемой стороны (a), угол B — против исходной (b), угол C — против проме-

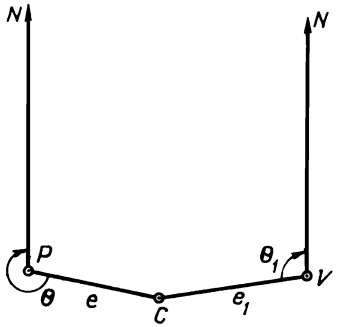


Рис. 19. Схема расположения элементов приведений:

N — начальное направление; *P* — проекция вертикальной оси теодолита; *C* — проекция центра пункта; *V* — проекция оси симметрии визирного цилиндра; 0 — угловой элемент центрировки; 0_1 — угловой элемент редукции; *e* — линейный элемент центрировки; e_1 — линейный элемент редукции

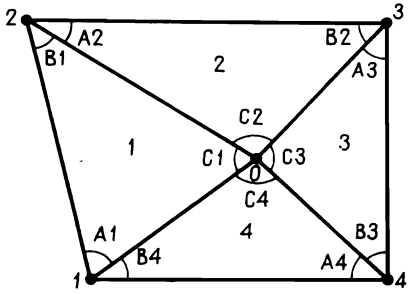


Рис. 20. Схема к программе «Решение треугольников»

Таблица 6
Диалог с программой «Решение треугольников» (режим «Вычисление сторон»)

Сообщение или запрос программы	Ответ пользователя
РЕШЕНИЕ ТРЕУГОЛЬНИКОВ.	
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН. КУДА?	1
ВВОД 1 — ВЫЧИСЛЕНИЕ СТОРОН, 0 — ОПРЕДЕЛЕНИЕ УГЛОВ. ЧТО?	1
ВЫЧИСЛЕНИЕ СТОРОН. ВВЕДИТЕ ЧИСЛО ТРЕУГОЛЬНИКОВ?	4
ВВОД ЧИСЛОВЫХ ЗНАЧЕНИЙ УГЛОВ ПО ТРЕУГОЛЬНИКАМ. УГЛЫ РАСПОЛОЖЕНЫ ПРОТИВ СТОРОН; А — ПРОТИВ ОПРЕДЕЛЯЕМОЙ, В — ПРОТИВ ИСХОДНОЙ, С — ПРОТИВ ПРОМЕЖУТОЧНОЙ.	
ФОРМАТ ВВОДА ЗНАЧЕНИЙ УГЛОВ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ И НАЖАТЬ ВК).	
ДОПУСТИМАЯ УГЛОВАЯ НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 100 СЕКУНД.	
ТРЕУГОЛЬНИК 1	
УГОЛ А1?	81,10,16
УГОЛ В1?	39,25,39
УГОЛ С1?	59,23,55
НЕВЯЗКА В ТРЕУГ. 1 РАВНА — 10 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 2	
УГОЛ А2?	30,33,32
УГОЛ В2?	54,43,03
УГОЛ С2?	94,43,05
НЕВЯЗКА В ТРЕУГ. 2 РАВНА —20 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 3	
УГОЛ А3?	41,11,33
УГОЛ В3?	49,20,48
УГОЛ С3?	89,27,26
НЕВЯЗКА В ТРЕУГ. 3 РАВНА —13 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 4	
УГОЛ А4?	34,51,06

Продолжение табл. 6

Сообщение или запрос программы	Ответ пользователя
УГОЛ В4?	28,43,30
УГОЛ С4?	116,25,34
НЕВЯЗКА В ТРЕУГ. 4 РАВНА 10 СЕКУНДАМ.	
ВВЕДИТЕ ДЛИНУ НАЧАЛЬНОГО БАЗИСА В МЕТРАХ?	1854.01
ВВЕДИТЕ ДЛИНУ КОНЕЧНОГО БАЗИСА В МЕТРАХ?	1854.01

Таблица 7
Результаты выполнения тестовой задачи по программе «Решение треугольников» (режим «Вычисление сторон»)

ВЫЧИСЛЕНИЕ СТОРОН ТРЕУГОЛЬНИКОВ СЕТИ

НАЗВАНИЕ УГЛА	УГОЛ, ГРАДУСЫ, МИН, С	ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ, М
	ТРЕУГОЛЬНИК 1	
A1	81 10 19	2884.60
B1	39 25 42	1854.01
C1	59 23 58	2512.65
	ТРЕУГОЛЬНИК 2	
A2	30 33 39	1796.67
B2	54 43 10	2884.60
C2	94 43 12	3521.62
	ТРЕУГОЛЬНИК 3	
A3	41 11 37	1559.68
B3	49 20 52	1796.67
C3	89 27 30	2368.05
	ТРЕУГОЛЬНИК 4	
A4	34 51 03	1854.51
B4	28 43 27	1559.68
C4	116 25 31	2906.25

жуточной (c). Для масштабирования сети используются длины исходных сторон.

Для примера приведен диалог между программой и пользователем при решении тестовой задачи (табл. 6). Использованы исходные данные, соответствующие рис. 20. Результаты вычисления выдает машина (табл. 7).

2.1.2. Вычисление углов треугольников сети по измеренным сторонам

Задача выполняется по программе «Решение треугольников» во втором режиме. Определение углов треугольника осуществляется по теореме косинусов:

$$\cos A = \frac{b^2+c^2-a^2}{2bc}; \quad \cos B = \frac{a^2+c^2-b^2}{2ac};$$

$$\cos C = \frac{a^2+b^2-c^2}{2ab}.$$

Таблица 8
Диалог с программой «Решение треугольников» (режим «Определение углов»)

Сообщение или запрос программы	Ответ пользователя
РЕШЕНИЕ ТРЕУГОЛЬНИКОВ.	
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН. КУДА?	1
ВВОД 1 — ВЫЧИСЛЕНИЕ СТОРОН, 0 — ОПРЕДЕЛЕНИЕ УГЛОВ. ЧТО?	0
ОПРЕДЕЛЕНИЕ УГЛОВ. ВВЕДИТЕ ЧИСЛО ТРЕУГОЛЬНИКОВ?	4
ВВОД СТОРОН (ГОРИЗ. ПРОЛОЖЕНИЙ) ПО ТРЕУГОЛЬНИКАМ.	
ВВОД СТОРОН ПО ХОДУ ЧАС. СТРЕЛКИ(А—В—С) В МЕТРАХ.	
ТРЕУГОЛЬНИК 1	
ВВЕДИТЕ СТОРОНУ А?	2884.51
СТОРОНУ В?	1854.01
СТОРОНУ С?	2512.55
ТРЕУГОЛЬНИК 2	
ВВЕДИТЕ СТОРОНУ А?	1796.51
СТОРОНУ В?	2884.51
СТОРОНУ С?	3521.43
ТРЕУГОЛЬНИК 3	
ВВЕДИТЕ СТОРОНУ А?	1559.46
СТОРОНУ В?	1796.51
СТОРОНУ С?	2367.76
ТРЕУГОЛЬНИК 4	
ВВЕДИТЕ СТОРОНУ А?	1854.01
СТОРОНУ В?	1559.46
СТОРОНУ С?	2905.62

Таблица 9
Результаты выполнения тестовой задачи по программе «Решение треугольников» (режим «Определение углов»)

ОПРЕДЕЛЕНИЕ УГЛОВ ТРЕУГОЛЬНИКОВ СЕТИ		
НАЗВАНИЕ УГЛА	УГОЛ, ГРАДУСЫ, МИН, С	ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ, М
ТРЕУГОЛЬНИК 1		
A1	81 10 18	2884.51
B1	39 25 47	1854.01
C1	59 23 55	2512.55
ТРЕУГОЛЬНИК 2		
A2	30 33 35	1796.51
B2	54 43 17	2884.51
C2	94 43 08	3521.43
ТРЕУГОЛЬНИК 3		
A3	41 11 34	1559.46
B3	49 21 01	1796.51
C3	89 27 26	2367.76
ТРЕУГОЛЬНИК 4		
A4	34 50 55	1854.01
B4	28 43 35	1559.46
C4	116 25 29	2905.62

Вычисленные значения углов выдаются с точностью до секунд.

Для примера приведен диалог при решении тестовой задачи (табл. 8). Исходные данные соответствуют рис. 20. Численные значения этих данных получены как результаты решения тестовой задачи (в табл. 7). Результаты решения данной тестовой задачи выдает машина в виде табл. 9.

2.1.3. Вычисление поправок, вызванных центрировкой и редукцией

Вычисление этих поправок осуществляется с помощью программы «Центрировка и редукция», цель которой приведение измеренных направлений или углов к центрам пунктов.

На основе измеренных угловых и линейных элементов приведений (см. рис. 19) вычисляют поправки, вызванные центрировкой c и редукцией r, до десятых долей секунды по формулам:

$$c = \frac{e \sin (M+\theta)}{S}\rho;$$

$$r = \frac{e \sin (M_1+\theta_1)}{S}\rho,$$

где M и M_1 — измеряемые углы; $\rho = 206\,265''$.

Расстояния S определяют с помощью программы (табл. 6 и 7). Поправки c и r вычисляют отдельно на каждом пункте сети.

Исходные данные к тестовой задаче для программы «Центрировка и редукция» соответствуют рис. 20. Диалог с этой программой приведен в табл. 10, а результаты решения этой задачи выдает машина в виде табл. 11.

2.1.4. Обратная геодезическая задача на плоскости

Программа предназначена для решения обратной геодезической задачи в системе прямоугольных координат (рис. 21).

Способ решения. Расстояние между исходной и конечной точками вычисляют по формуле

$$S = \sqrt{\Delta X^2 + \Delta Y^2},$$

где $\Delta X = X_B - X_A$; $\Delta Y = Y_B - Y_A$.

Если это расстояние окажется меньше 0,01 м, то машина выдает сообщение о том, что дважды введены координаты одной и той же точки и необходимо повторять ввод координат точек.

Если $|\Delta X| < 0{,}01$ и $\Delta Y > 0$, то дирекционный угол принимается равным 90° (восток); если $|\Delta X| < 0{,}01$ и $\Delta Y < 0$, то дирекционный угол принимается равным 270° (запад). Случай, когда $\Delta Y = 0$, здесь невозможен, так как при этом окажется, что $S < 0{,}01$.

Если $|\Delta X| \geqslant 0{,}01$, то вычисляется острый угол $\alpha = \operatorname{arctg}(\Delta Y/\Delta X)$. При использовании с этой целью встроенной функции ATN вычисленный угол будет лежать в диапазоне от -90 до $+90°$, причем отрицательные

Таблица 10
Диалог с программой «Центрировка и редукция»

Сообщение или запрос программы		Ответ пользователя
ЦЕНТРИРОВКА И РЕДУКЦИЯ.		
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН. КУДА?		1
ВЫЧИСЛЕНИЕ ПОПРАВОК ЗА ЦЕНТРИРОВКУ И РЕДУКЦИЮ.		
ВВЕДИТЕ НАЗВАНИЕ ПУНКТА (ДО 10 БУКВ)?		КОСТЮКИ
ВВЕДИТЕ ЧИСЛО НАПРАВЛЕНИЙ НА ПУНКТЕ?		3
ВВОД ФАКТИЧЕСКИХ НАЗВАНИЙ ПУНКТОВ (КАЖДОЕ ДО 10 БУКВ).		
УСЛОВНЫЕ: 0 — НАЧАЛЬНОЕ НАПРАВЛЕНИЕ, ДАЛЕЕ ПО ХОДУ ЧАС. СТР.		
УСЛОВНЫЙ НОМЕР 0, ФАКТИЧЕСКОЕ НАЗВАНИЕ?		ВЫСЕЛКИ
УСЛОВНЫЙ НОМЕР 1, ФАКТИЧЕСКОЕ НАЗВАНИЕ?		МИШИНО
УСЛОВНЫЙ НОМЕР 2, ФАКТИЧЕСКОЕ НАЗВАНИЕ?		ПРОНИНО
ФОРМАТ ВВОДА ЗНАЧЕНИЙ НАПРАВЛЕНИЙ (В ГРАДУСАХ, МИНУТАХ).		
ФОРМАТ ВВОДА ЗНАЧЕНИЙ РАССТОЯНИЙ (В ЦЕЛЫХ МЕТРАХ).		
ВВОД ИСХОДНЫХ ДАННЫХ:		
КОСТЮКИ — ВЫСЕЛКИ	НАПРАВЛЕНИЕ?	0,0
КОСТЮКИ — ВЫСЕЛКИ	РАССТОЯНИЕ?	1560
КОСТЮКИ — МИШИНО	НАПРАВЛЕНИЕ?	49,20
КОСТЮКИ — МИШИНО	РАССТОЯНИЕ?	2368
КОСТЮКИ — ПРОНИНО	НАПРАВЛЕНИЕ?	325,09
КОСТЮКИ — ПРОНИНО	РАССТОЯНИЕ?	2906
ВВОД ЛИНЕЙНОГО ЭЛЕМЕНТА ЦЕНТРИРОВКИ В МЕТРАХ?		0.015
ВВОД УГЛОВОГО ЭЛЕМЕНТА ЦЕНТРИРОВКИ (В ГРАДУСАХ)?		205
ВВОД ЛИНЕЙНОГО ЭЛЕМЕНТА РЕДУКЦИИ В МЕТРАХ?		0.06
ВВОД УГЛОВОГО ЭЛЕМЕНТА РЕДУКЦИИ (В ГРАДУСАХ)?		285

Таблица 11
Результаты решения тестовой задачи по программе «Центрировка и редукция»

ЦЕНТРИРОВКА И РЕДУКЦИЯ			
ПУНКТ	ЦЕНТРИРОВКА, С	РЕДУКЦИЯ, С	ПУНКТ НАПРАВЛЕНИЯ
КОСТЮКИ	—0.8	—7.7	ВЫСЕЛКИ
	—1.3	—2.3	МИШИНО
	0.2	—4.0	ПРОНИНО

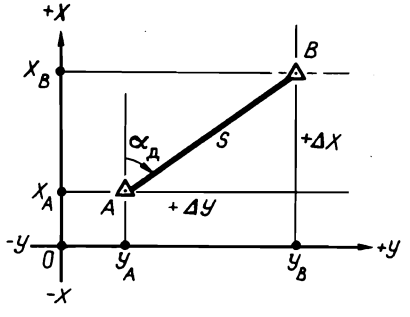


Рис. 21. Схема обратной геодезической задачи

Таблица 12
Диалог с программой «Обратная геодезическая задача»

Сообщение или запрос программы	Ответ пользователя
РЕШЕНИЕ ОБРАТНОЙ ГЕОДЕЗИЧЕСКОЙ ЗАДАЧИ НА ПЛОСКОСТИ. ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН. КУДА? КОНЕЦ РАБОТЫ — ВВОД ДВАЖДЫ НУЛЕВЫХ ДАННЫХ (0, 0, 0). В КООРДИНАТАХ ПОВТОРЯЮЩИЕСЯ ЦИФРЫ СЛЕВА ОТБРОСИТЬ.	1
ВВОД НОМЕРА И КООРДИНАТ НАЧАЛЬНОЙ ТОЧКИ (N, XH, YH)?	7, 7353.46, 2858.57
ВВОД НОМЕРА И КООРДИНАТ КОНЕЧНОЙ ТОЧКИ (N, XK, YK)?	15, 6978.08, 1903.34

значения он будет принимать в случае несовпадения знаков ΔX и ΔY.

Затем, в зависимости от знаков ΔX и ΔY определяется дирекционный угол:

если $\Delta X > 0$ и $\Delta Y \geqslant 0$, то $\alpha_д = \alpha$ (северо-восток или при $\Delta Y = 0$ — север);

если $\Delta X > 0$ и $\Delta Y < 0$, то $\alpha_д = 360° + \alpha$ (северо-запад);

если $\Delta X < 0$, то независимо от знака $\Delta Y \alpha_д = 180° + \alpha$ (юго-запад, юг, юго-восток).

В приведенных выше формулах угол α берется со своим знаком. Решение ведется с удержанием шести верных (значащих) цифр. Значение дирекционного угла выдается с точностью до секунд, а горизонтального проложения — до сантиметров.

Пример диалога при решении тестовой задачи приведен в табл. 12. Результаты решения тестовой задачи машина выдает в следующей форме:

ОБРАТНАЯ ГЕОДЕЗИЧЕСКАЯ ЗАДАЧА НА ПЛОСКОСТИ
НАЧ. ТОЧКА 7 X=7353.46 Y=2858.57
КОН. ТОЧКА 15 X=6978.08 Y=1903.34
ДЛИНА ЛИНИИ /7—15/=1026.34 М
ДИРЕКЦИОННЫЙ УГОЛ=248 42 48 .

2.1.5. Перевычисление координат из системы в систему

Программа предназначена для перевода декартовых координат точек из одной системы (старой) в другую (новую). Старая и новая системы могут иметь различные масштабы, для определения которых используются координаты двух опорных точек в обеих системах. Введем обозначения: $X^с$, $Y^с$ — координаты некоторой точки в старой системе; $X^н$, $Y^н$ — искомые координаты той же точки в новой системе; $X^с_1$, $Y^с_1$, $X^н_1$, $Y^н_1$, $X^с_2$, $Y^с_2$, $X^н_2$, $Y^н_2$ — известные координаты двух опорных точек в старой и новой системах.

Формулы перевода имеют вид:

$$X^н = X^н_2 + K\cos\tau(X^с - X^с_2) - K\sin\tau(Y^с - Y^с_2);$$

$$Y^н = Y^н_2 + K\cos\tau(Y^с - Y^с_2) - K\sin\tau(X^с - X^с_2).$$

Здесь $K=S^{н}_{12}/S^{с}_{12}$ — масштабный коэффициент.

Расстояние между опорными точками в старой и новой системах определяют по формулам:

$$S^{н}=\sqrt{(X^{н}_{2}-X^{н}_{1})^{2}+(Y^{н}_{2}-Y^{н}_{1})^{2}};$$

$$S^{с}=\sqrt{(X^{с}_{2}-X^{с}_{1})^{2}+(Y^{с}_{2}-Y^{с}_{1})^{2}}.$$

Угол $\tau=\alpha^{с}_{12}-\alpha^{н}_{12}$, где $\alpha^{с}_{12}$ и $\alpha^{н}_{12}$ — дирекционные углы из первой опорной точки на вторую в старой и новой системах координат.

Программа содержит примерно 110 операторов, из которых 14 предназначены для вывода на экран информации о назначении и возможностях программы; 17 — содержат диалог, посвященный выбору режима работы программы; 16 — предназначены для ввода исходных данных; 18 — осуществляют вывод результатов; 7 — осуществляют контроль исходных данных (проверяют, не совпали ли координаты обеих опорных точек); 33 — предназначены для вычислений. Соотношение между сервисной и вычислительной частями программы составляет примерно 2 : 1. Такое соотношение является нормальным (если не заниженным) для программного обеспечения персональных ЭВМ.

Для ввода координат опорных точек используется подпрограмма. Параметры K и τ вычисляются один раз после ввода координат опорных точек, причем для вычисления расстояний $S^{с}_{12}$, $S^{н}_{12}$ и дирекционных углов $\alpha^{с}_{12}$, $\alpha^{н}_{12}$ также используются подпрограммы.

Программа в любом случае выводит результаты на экран и по требованию пользователя выводит на печать таблицу координат заданных точек в исходной и новой системах.

Переход к переводу координат очередной точки или завершение работы программы происходит по указанию пользователя. Необходимые действия пользователя и реакция программы приведены в табл. 13.

Таблица 13
Диалог с программой «Перевычисление координат»

Номер п/п	Действие пользователя	Реакция программы
1	RUN	СООБЩЕНИЕ: РАБОТАЕТ ПРОГРАММА РЕКООR. НУЖЕН ЛИ ВАМ ПОЯСНИТЕЛЬНЫЙ ТЕКСТ? ДА,— ВВЕДИТЕ Y, НЕТ,— ВВЕДИТЕ N. ПОСЛЕ ВВОДА НА ЭКРАН БУКВЫ Y ИЛИ N НЕ ЗАБУДЬТЕ НАЖАТЬ КЛАВИШУ ВК
2	N	Пункты 3, 4 — пропускаются. Пункт 5 в этом случае выполняется без согласия пользователя
3		СООБЩЕНИЕ И ЗАПРОС: ПРОГРАММА ПРЕДНАЗНАЧЕНА ДЛЯ ПЕРЕВОДА ДЕКАРТОВЫХ КООРДИНАТ ТОЧЕК ИЗ НЕКОТОРОЙ ИСХОДНОЙ СИСТЕМЫ В ДРУГУЮ (НОВУЮ). Масштабные множители в исходной и новой системах могут быть разными. Для вычисления параметров уравнений переноса координат необходимо предварительно ввести известные координаты двух точек в исходной и новой системах. После этого можно вводить координаты заранее необусловленного числа точек в исходной системе

Продолжение табл. 13

Номер п/п	Действие пользователя	Реакция программы
		с целью их перевода в новую систему. Координаты очередной точки в новой системе выводятся на экран непосредственно после ввода координат в исходной системе. Кроме того, по требованию пользователя результаты могут быть отпечатаны в виде таблицы.
		«ХОТИТЕ ВОСПОЛЬЗОВАТЬСЯ ПРОГРАММОЙ? ДА,— ВВЕДИТЕ Y, НЕТ — ВВЕДИТЕ N»
4	N	СООБЩЕНИЕ: «ВЫ ОТКАЗАЛИСЬ ОТ ПРОГРАММЫ. ДО СВИДАНИЯ». ПЕРЕХОД К П. 19
5	Y	ЗАПРОС: «БУДЕТЕ ПЕЧАТАТЬ РЕЗУЛЬТАТЫ? ДА,— ВВЕДИТЕ Y, НЕТ,— ВВЕДИТЕ N»
6	N	Пропускается пункт 7
7	Y	Открывается выходной файл на печатающем устройстве
		СООБЩЕНИЕ: ТАБЛИЦА ИСХОДНЫХ И НОВЫХ КООРДИНАТ БУДЕТ ОТПЕЧАТАНА ПО ЗАВЕРШЕНИЮ ВВОДА ВСЕХ ИСХОДНЫХ ДАННЫХ
8		ЗАПРОС: ВВОДИТЕ КООРДИНАТЫ ОПОРНЫХ ТОЧЕК. В ИСХОДНОЙ СИСТЕМЕ: ПЕРВАЯ ТОЧКА X_1, Y_1=?
9	Вводит значения X_1^c, Y_1^c	ЗАПРОС: ВТОРАЯ ТОЧКА X_2, Y_2=?
10	Вводит по ошибке повторно значения X_1^c, Y_1^c	СООБЩЕНИЕ: ВЫ ДВАЖДЫ ВВЕЛИ КООРДИНАТЫ ОДНОЙ И ТОЙ ЖЕ ТОЧКИ. ПРОВЕРЬТЕ ИСХОДНЫЕ ДАННЫЕ И ПОВТОРИТЕ ВВОД. ПЕРЕХОД К П.8
11	Вводит значения X_2^c, Y_2^c	ЗАПРОС: В НОВОЙ СИСТЕМЕ. ПЕРВАЯ ТОЧКА X1, Y1=?
12	Вводит значения X_1'', X_1''	ЗАПРОС: ВТОРАЯ ТОЧКА X2, Y2=?
13	Вводит по ошибке повторно значения X_1'', Y_1''	СООБЩЕНИЕ: ВЫ ДВАЖДЫ ВВЕЛИ КООРДИНАТЫ ОДНОЙ И ТОЙ ЖЕ ТОЧКИ. ПРОВЕРЬТЕ ИСХОДНЫЕ ДАННЫЕ И ПОВТОРИТЕ ВВОД. ПЕРЕХОД К П.8
14	Вводит значения X_2'', Y_2''	СООБЩЕНИЕ: ПРИСТУПАЙТЕ К ПЕРЕВОДУ КООРДИНАТ
15		ВВОДИТЕ КООРДИНАТЫ ТОЧКИ В ИСХОДНОЙ СИСТЕМЕ: ХИСХ, УИСХ=?
16	Вводит значения X^c, Y^c	ВЫВОД РЕЗУЛЬТАТА НА ЭКРАН: ХНОВ = ЗНАЧЕНИЕ X'', УНОВ = ЗНАЧЕНИЕ Y''. ЗАСЫЛКА В БУФЕР ПЕЧАТИ ЗНАЧЕНИЙ X^c, Y^c, X'', Y''. ЗАПРОС: ЕСТЬ ЕЩЕ ТОЧКА? ДА — ВВЕДИТЕ Y, НЕТ — ВВЕДИТЕ N.
17	Y	ПЕРЕХОД К П. 15
18	N	Вывод таблицы со значениями X^c, Y^c, X'', Y'' для всех точек, если такой вывод был предусмотрен пользователем
19		Прекращение работы программы

Сообщения этой программы могут показаться слишком многословными. Вполне возможно, что аналогичная программа с более лаконичным диалогом придется более по вкусу многим пользователям. С другой

стороны контроль в программе можно было бы увеличить, введя, например, контроль значений вводимых координат на вхождение в некоторый диапазон или диапазоны, заранее указанные пользователем.

2.1.6. Передача координат с вершины знака на землю

Назначение программы. Программа предназначена для вычисления координат X и Y пункта P (рис. 22) по известным координатам пункта A и координатам удаленных пунктов B и C существующей исходной сети (один из них для контроля), а также по измеренным двум базисам L_1 и L_2 и шести углам β_1, β_2, β_3, β_4, δ_1 и δ_2. Точка A является вершиной знака пункта триангуляции, на который нельзя встать с прибором. Второй базис и углы при нем необходимы для контроля определения расстояния AP и повышения точности получения окончательного его значения. Угол δ_2 обеспечивает контроль правильности произведенных измерений и выписки исходных данных, что способствует повышению точности определения окончательного значения координат точки P.

Программа написана на исходной версии языка Бейсик, что позволяет реализовать ее в среде, предназначенной для работы с другими более развитыми версиями.

Способ решения. Программа составлена по приведенным ниже формулам. Дирекционный угол α_{AB} и расстояние S_{AB} между исходными пунктами A и B вычисляют по формулам:

$$\operatorname{tg}\alpha_{AB}=\frac{Y_B-Y_A}{X_B-X_A};\quad S_{AB}=\sqrt{(X_B-X_A)^2+(Y_B-Y_A)^2}.$$

Недоступное расстояние d от вершины знака A до определяемого пункта P определяют дважды по формулам

$$d_1=\frac{L_1\sin\beta_2}{\sin\gamma_1};\quad d_2=\frac{L_2\sin\beta_4}{\sin\gamma_2},$$

где $\gamma_1=180^\circ-(\beta_1+\beta_2)$; $\gamma_2=180^\circ-(\beta_3+\beta_4)$.

Если $\dfrac{|d_1-d_2|}{d}\leqslant\dfrac{1}{1000}$,

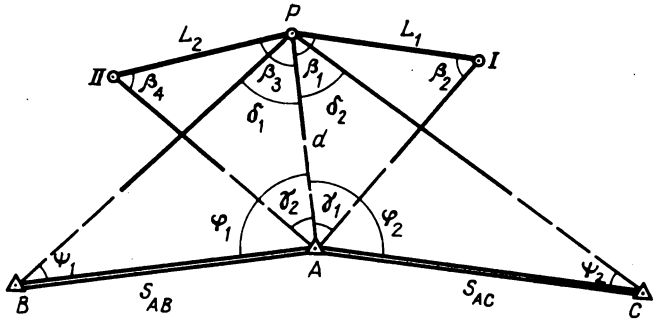


Рис. 22. Схема к программе «Передача координат с вершины знака на землю»

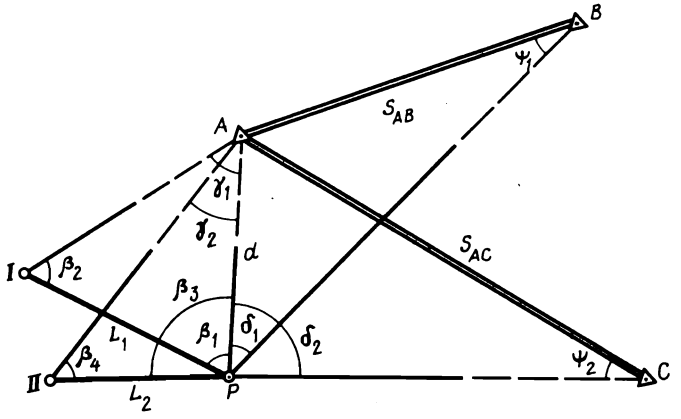


Рис. 23. Схема к тестовой задаче для программы «Передача координат с вершины знака на землю»

то

$$d_{ср}=(d_1+d_2)/2.$$

Вспомогательные углы ψ_1 и φ_1 треугольника ABP вычисляют по формулам:

$$\sin\psi_1=\frac{d_{ср}\sin\delta_1}{S_{AB}};$$

$$\varphi_1=180^\circ-(\delta_1+\psi_1).$$

Дирекционный угол α_{AP} линии AP определяют по формуле

$$\alpha_{AP}=\alpha_{AB}\pm\varphi_1.$$

Знак «плюс» или «минус» берется в зависимости от расположения угла φ_1 относительно направления AB.

Вычисление координат определяемого пункта P производится по формулам:

$$\Delta X_{AP}=d_{ср}\cos\alpha_{AP};$$

$$\Delta Y_{AP}=d_{ср}\sin\alpha_{AP};$$

$$X_P=X_A+\Delta X_{AP};$$

$$Y_P=Y_A+\Delta Y_{AP}.$$

Второе решение задачи по координатам другого удаленного исходного пункта C производится по аналогичным формулам.

Расхождение между двумя значениями дирекционного угла линии AP не должно превышать $6m$ (m — средняя квадратическая погрешность измерения угла).

Подготовка исходных данных. Подготовка информации для решения задачи включает в себя: составление схематического чертежа (рис. 23)

Таблица 14
Координаты исходных пунктов к тестовой задаче для программы передачи координат с вершины знака на землю

Название пункта	Координаты, м	
	X	Y
Вершина знака A	6 327,46	12 351,48
Первый удаленный исходный пункт B	7 049,74	17 577,28
Второй удаленный исходный пункт C	3 700,05	13 936,62

Таблица 15
Результаты измерений к тестовой задаче для программы передачи координат с вершины знака на землю

Обозначения	Числовое значение
Средняя квадратическая погрешность измерения угла m, с	5
Базисы, м:	
L_1	88,71
L_2	66,13
Углы:	
β_1	70°08′54″
β_2	38 26 00
β_3	87 28 00
β_4	42 26 36
δ_1	71 55 44
δ_2	138 34 31

и составление таблиц координат исходных пунктов (табл. 14) и результатов измерений (табл. 15).

На схематический чертеж наносят пункт A, координаты которого будут передавать на землю, определяемый пункт P, два удаленных исходных пункта B, C, два базиса L_1, L_2. Исходные пункты обозначают треугольничком с точкой посередине, определяемый пункт — кружком, направления с вершины знака на исходные пункты — двойными сплошными линиями, базисы — утолщенными сплошными линиями, направления с определяемого пункта на исходные — полусплошными линиями, углы — дугами. Чертеж должен быть ориентирован на север.

Линия AP может иметь северо-восточное, юго-восточное, юго-западное или северо-западное направление. Данная программа обрабатывает все варианты. Для этого надо в начале программы ввести признак «1», если линия AP имеет юго-восточное или юго-западное направление, и «0», если линия AP имеет северо-западное или северо-восточное направление. Например, при решении задачи для случая расположения линии AP (см. рис. 22), которая имеет северо-западное направление, при вводе данных следует ввести признак «0». На рис. 23 линия имеет юго-западное направление, поэтому следует ввести признак «1».

В таблицу исходных данных (см. табл. 14) записывают координаты вершины знака и координаты двух исходных удаленных пунктов.

В таблицу измеренных величин (см. табл. 15) записывают среднюю квадратическую погрешность измерения угла, длины двух базисов, углы,

Таблица 16
Диалог с программой «Передача координат с вершины знака на землю»

Сообщение или запрос программы	Ответ пользователя
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН. КУДА?	1
ВВЕДИТЕ ПРИЗНАК: 1,— ЕСЛИ ЛИНИЯ ОТ ВЕРШИНЫ ЗНАКА ДО ОПРЕДЕЛЯЕМОГО ПУНКТА ИМЕЕТ ЮВ ИЛИ ЮЗ НАПРАВЛЕНИЕ, 0,— ЕСЛИ ЭТА ЛИНИЯ ИМЕЕТ СЗ ИЛИ СВ НАПРАВЛЕНИЕ.	
?	1
ВВЕДИТЕ КООРДИНАТЫ ВЕРШИНЫ ЗНАКА В МЕТРАХ.	
?	6327.46, 12351.48
ВВЕДИТЕ КООРДИНАТЫ ПЕРВОГО УДАЛЕННОГО ИСХОДНОГО ПУНКТА В МЕТРАХ.	
?	7049.74, 17577.28
ВВЕДИТЕ КООРДИНАТЫ ВТОРОГО УДАЛЕННОГО ИСХОДНОГО ПУНКТА В МЕТРАХ.	
?	3700.05, 13936.62
ВВЕДИТЕ СРЕДН. КВАДРАТ. ПОГРЕШНОСТЬ ИЗМЕРЕНИЯ УГЛА В СЕКУНДАХ.	
?	5
ВВЕДИТЕ ДЛИНУ ПЕРВОГО БАЗИСА В МЕТРАХ L1 ?	88,71
ВВЕДИТЕ ДЛИНУ ВТОРОГО БАЗИСА В МЕТРАХ L2 ?	66.13
ВВЕДИТЕ ДВА УГЛА, ИЗМЕРЕННЫЕ НА КОНЦАХ ПЕРВОГО БАЗИСА, В ГРАДУСАХ, МИНУТАХ, СЕКУНДАХ:	
ВВЕДИТЕ УГОЛ С ВЕРШИНОЙ В ОПРЕДЕЛЯЕМОМ ПУНКТЕ.	
?	70,08,54
ВВЕДИТЕ УГОЛ С ВЕРШИНОЙ НА ПРОТИВОПОЛОЖНОМ КОНЦЕ БАЗИСА.	
?	38,26,00
ВВЕДИТЕ ДВА УГЛА, ИЗМЕРЕННЫЕ НА КОНЦАХ ВТОРОГО БАЗИСА, В ГРАДУСАХ, МИНУТАХ, СЕКУНДАХ:	
ВВЕДИТЕ УГОЛ С ВЕРШИНОЙ В ОПРЕДЕЛЯЕМОМ ПУНКТЕ.	
?	87,28,00
ВВЕДИТЕ УГОЛ С ВЕРШИНОЙ НА ПРОТИВОПОЛОЖНОМ КОНЦЕ БАЗИСА.	
?	42,26,36
ВВЕДИТЕ УГОЛ С ВЕРШИНОЙ В ОПРЕД. ПУНКТЕ И НАПРАВЛЕНИЯМИ НА ВЕРШИНУ ЗНАКА И ПЕРВЫЙ УДАЛЕННЫЙ ИСХОДНЫЙ ПУНКТ В ГРАДУСАХ, МИНУТАХ, СЕКУНДАХ.	
?	71,55,44
ВВЕДИТЕ УГОЛ С ВЕРШИНОЙ В ОПРЕД. ПУНКТЕ И НАПРАВЛЕНИЯМИ НА ВЕРШИНУ ЗНАКА И ВТОРОЙ УДАЛЕННЫЙ ИСХОДНЫЙ ПУНКТ В ГРАДУСАХ, МИНУТАХ, СЕКУНДАХ.	
?	138,34,31

измеренные на концах первого и второго базисов, а также углы, измеренные на определяемом пункте между направлениями на вершину знака и удаленные исходные пункты.

Диалог с программой в процессе решения тестовой задачи приведен

в табл. 16. Результаты решения тестовой задачи машина выводит в следующей форме:

ПЕРЕДАЧА КООРДИНАТ С ВЕРШИНЫ ЗНАКА НА ЗЕМЛЮ.
РАССТОЯНИЕ ОТ ВЕРШИНЫ ЗНАКА ДО ПЕРВОГО УДАЛЕННОГО ИСХОДНОГО ПУНКТА S1=5725.48 М.
ДИРЕКЦИОННЫЙ УГОЛ ПЕРВОЙ ИСХОДНОЙ СТОРОНЫ A1= =82 ГРАДУСА, 7 МИН, 51 С.
РАССТОЯНИЕ ОТ ВЕРШИНЫ ЗНАКА ДО ВТОРОГО УДАЛЕННОГО ИСХОДНОГО ПУНКТА S2=3068.54 М.
ДИРЕКЦИОННЫЙ УГОЛ ВТОРОЙ ИСХОДНОЙ СТОРОНЫ A2= =148 ГРАДУСОВ, 53 МИН, 49 С.
НЕДОСТУПНОЕ РАССТОЯНИЕ ОТ ВЕРШИНЫ ЗНАКА ДО ОПРЕДЕЛЯЕМОГО ПУНКТА D=58.18 М.
ДИРЕКЦИОННЫЙ УГОЛ ОТ ВЕРШИНЫ ЗНАКА ДО ОПРЕДЕЛЯЕМОГО ПУНКТА, ПЕРЕДАННЫЙ ОТ ПЕРВОЙ ИСХОДНОЙ СТОРОНЫ (В ГРАДУСАХ), A3=189,601.
ДИРЕКЦИОННЫЙ УГОЛ ОТ ВЕРШИНЫ ЗНАКА ДО ОПРЕДЕЛЯЕМОГО ПУНКТА, ПЕРЕДАННЫЙ ОТ ВТОРОЙ ИСХОДНОЙ СТОРОНЫ (В ГРАДУСАХ), A4=189,603.
КООРДИНАТЫ ОПРЕДЕЛЯЕМОГО ПУНКТА: X=6270.1, Y= =12341.8 М.

2.1.7. Определение астрономических азимутов исходных направлений

Способ решения. Для получения азимутов с необходимой точностью без дорогостоящего специального оборудования можно применять способ, основанный на измерении часового угла светила и горизонтального угла между звездой и земным предметом.

При правильной организации наблюдений, применении теодолита 2Т2 и спортивного секундомера можно получить азимут со средней квадратической погрешностью 3—5″ по наблюдениям Полярной звезды (из 4—6 приемов) и 5—10″ — по наблюдениям Солнца (из 3—4 приемов).

При этом необходимо знать поправку хронометра u, долготу λ и ширину φ места наблюдения. Значения широты и долготы места должны быть получены из астрономических определений (или вычислены по прямоугольным координатам пункта) с погрешностью соответственно 5″ и $0{,}3^s$.

Выгоднейшими условиями определения азимута A направления на земной предмет по часовым углам t Солнца являются утренние и вечерние часы. Этот способ определения азимута применяется практически на любой широте, но Солнце необходимо наблюдать на зенитных расстояниях Z от 50 до 80°. Способ определения азимута по часовым углам Полярной звезды рекомендуется применять в широтной зоне от 10 до 60° северного полушария, но при этом следует учитывать, что погрешность определения азимута при одной и той же программе и средствах измерений возрастает с увеличением широты пункта пропорционально $\sec \varphi$.

Программы составлены по приведенным ниже формулам. Алгоритм определения астрономических азимутов исходных направлений разбит на две части — для упрощенных наблюдений (по Солнцу), и точных (по Полярной звезде).

Вычисление азимута по часовым углам Солнца. По результатам сравнения показаний X хронометра (часов) с радиосигналами точного времени (до и после наблюдений) вычисляют поправки и часовой ход $\omega_\odot$ хронометра по формулам:

$$u = D_n + \Delta n - X;$$

$$\omega_\odot = \frac{u_2 - u_1}{X_2 - X_1},$$

где D_n — декретный момент подачи сигналов проверки времени (СПВ), формируемый на базе шкалы координированного времени СССР UTC_{SU}; $\Delta n = n - n_n$ — разность часовых поясов приема и подачи эталонных сигналов времени.

Момент наблюдения $T_\odot$ Солнца и отсчеты $N_\odot$ по горизонтальному кругу теодолита, приведенные к центру диска Солнца, вычисляют по формулам:

$$T_{\odot I} = 0{,}5\,(T_1 + T_4);$$

$$T_{\odot II} = 0{,}5\,(T_2 + T_3);$$

$$N_{\odot I} = 0{,}5\,|NЛ_1 + (NП_4 \pm 180°)|;$$

$$N_{\odot II} = 0{,}5\,|NЛ_2 + (NП_3 \pm 180°)|,$$

где NЛ, NП — отсчеты по горизонтальному кругу при положениях вертикального круга слева (КЛ) и справа (КП).

Для повышения точности фиксирования моментов времени используют стрелочный секундомер. Тогда показание хронометра в момент приема радиосигналов времени будет складываться из отсчета по хронометру $T_{ПС}$ (в момент пуска секундомера) и отсчета по секундомеру $T_{СКМ}$ в момент начала подачи шестого радиосигнала, соответствующего началу часа, т. е.

$$X = T_{пс} + T_{скм}.$$

Аналогичным образом момент наблюдений T будет состоять из отсчета по хронометру в момент пуска секундомера и отсчета по секундомеру в момент касания края видимого диска Солнца с вертикальной нитью сетки нитей трубы теодолита, т. е.

$$T = T_{пс} + T_{скм}.$$

Для выявления возможных просчетов выполняют полевой контроль:

$$\Delta T = T_{\odot I} - T_{\odot II};$$

$$\Delta N_{пр} = N_{\odot I} - N_{\odot II}.$$

Практическое значение разности отсчетов $\Delta N_{пр}$ не должно отличаться от теоретического $\Delta N_т$ более чем на 15″. Причем, значение $\Delta N_т$ можно подсчитать по приближенной формуле

$$\Delta N_\tau \approx 15\Delta T \sin \varphi.$$

Коллимационная ошибка не должна превышать 10″. Если условия контроля выполнены, то осуществляется перенос поправки хронометра на приведенные моменты наблюдений по формулам:

$$u = u_1 + \omega_\odot (T_\odot - X_1);$$

$$D = T_\odot + u.$$

Затем приводят средний момент наблюдений D_m к эфемеридной шкале времени ET:

$$M = D_m - (n + k);$$

$$M^* = M + \Delta T\,(A),$$

где M — всемирное время по шкале UT; k — сезонный коэффициент (в зимнее время $k=1$, в летнее — $k=2$); $\Delta T(A)$ — поправка за переход к эфемеридному времени M*.

Координаты Солнца δ и E вычисляют по формулам:

$$\delta = \delta_0 + M v_0 + \Delta\delta;$$

$$\Delta\delta = \frac{M^{*2}}{48}(v_1 - v_0),$$

где δ_0 — склонение (на дату наблюдений) для 0^h земного динамического времени; v_0 — часовое изменение для δ_0; v_1 — часовое изменение для склонения на следующую дату.

Вспомогательная величина

$$E = E_0 + M^* v_0 + \Delta E;$$

$$\Delta E = \frac{M^*}{48}(v_1 - v_0),$$

где E_0 — уравнение времени на $+12^h$ для 0^h земного динамического времени (на дату наблюдений); v_0 — часовое изменение для E_0; v_1 — часовое изменение для величины E на следующую дату.

По вычисленной величине E подсчитывают часовые углы истинного Солнца для моментов D:

$$t_\odot = m + E - \mu\Delta T\,(A);$$

$$m = D - |(n + k) - \lambda_E|,$$

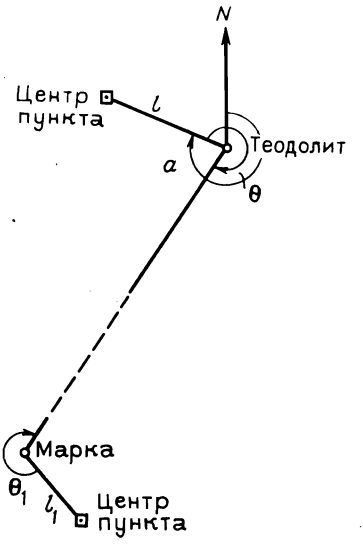
где m — среднее солнечное время; $\mu\Delta T\,(A)$ — поправка за переход от динамического среднего Солнца к среднему Солнцу; λ_E — долгота пункта, считаемая положительной к востоку от Гринвича.

По значению склонения Солнца вычисляют относительные постоянные κ_1 и κ_2, необходимые при многократных определениях азимутов направлений с одного пункта, по формулам:

$$\kappa_1 = \sin \varphi;$$

$$\kappa_2 = \operatorname{tg} \delta \cos \varphi.$$

Рис. 24. Схема вычисления азимута по часовым углам Солнца



Затем вычисляют азимуты Солнца $a_\odot$:

$$\operatorname{ctg} a_\odot = \kappa_1 \operatorname{ctg} t_\odot - \kappa_2 \operatorname{cosec} t_\odot,$$

которые отсчитываются от северного направления меридиана к востоку, как это принято в геодезии. Четверть, в которой находится направление на Солнце определяется по величине часового угла и знаку $\operatorname{ctg} a_\odot$.

Определение азимута земного предмета из J-го приема осуществляется с учетом «места севера» (MN) по формулам:

$$MN = N_\odot - a_\odot;$$

$$A_0 = N - MN,$$

где N — средний отсчет (по горизонтальному кругу) на земной предмет.

За окончательное значение азимута принимают среднее арифметическое из n приемов. Оценка точности окончательного результата выполняется по формуле

$$m_A = \mu_A / \sqrt{n},$$

где $\mu_A = \sqrt{\frac{[vv]}{n-1}}$ — погрешность единицы веса; v — уклонение отдельных значений A от среднего.

При наблюдении азимута (рис. 24) теодолит может быть установлен не над центром геодезического пункта, а в стороне. Визирная цель в свою очередь обычно не совмещена с центром пункта, азимут направления на который определяется. Следовательно, азимут необходимо привести к центру пункта, т. е. ввести в наблюденное значение азимута три поправки: за центрировку теодолита Δa_l, за редукцию визирной цели Δa_r и за сближение меридианов Δa_γ (рис. 24).

Таблица 17

Журнал наблюдений

Прием радиосигналов (до наблюдений):	Пункт: *АП-6*	Направление: *ОРП-2 (марки)*	Прием радиосигналов (после наблюдений):
$D_1 6^h 00^m 00{,}0^s$	Дата: 6.06.80	Теодолит: ТБ-1 № 6300	$D_2 = 8^h 00^m 00{,}0^s$
$T_{пс} = 5^h 59^m 00{,}0^s$	Хронометр № 1212	Секундомер «Златоуст» № 95040	$T_{пс} = 7^h 58^m 00{,}0^s$
$T_{скм} = 1^m 15{,}5^s$	$\varphi = 55°54'12''$	$\lambda = 2^h 32^m 21^s{,}0^s$	$T_{скм} = 2^m 18{,}5^s$

Прием 1. *Наблюдал Бондаренко А. М.*

Номер наблюдения	Объект наблюдения	Момент наблюдений по хронометру h	m	s	по секундомеру m	s	Общий момент h	m	s	Отсчет по горизонтальному кругу °	′	″	Полевой контроль	Элементы приведения
										КЛ				
	ОРП-2									149	02	12,0	$T_{⊙I} = 7^h 00^m 22{,}0^s$	$l = 5{,}8$ м
													$T_{⊙II} = 6\ 59\ 57{,}5$	$\theta = 49°30'$
1		6	57	00	0	35,0	6	57	35,0	304	19	40,0	$\Delta T = +24{,}5$	$l_1 = 0{,}05$ м
2		6	58	00	1	05,0	6	59	05,0	305	12	31,5	$N_{⊙I} = 305°09'57{,}5''$	$\theta_1 = 68°45'$
													$N_{⊙II} = 305°05'02{,}5''$	
										КП			$\Delta N_{пр} = +4'55{,}0''$	$S = 386{,}5$ м
3		7	00	00	0	50,0	7	00	50,0	124	57	33,5	$\Delta N_т = +5'04{,}3''$	$a = 220°30'$
4		7	02	00	1	09,0	7	03	09,0	126	00	15,0	$\delta = -9{,}3''$	
	ОРП-2									329	02	27,0	$2c = -15{,}0''$	Солнце: северо-восток

Элементы центрировки: l — горизонтальное проложение между центром теодолита и центром пункта; θ — угол между направлениями на центр данного пункта и на сигнал (визирную цель).

Элементы редукции: l_1 — горизонтальное проложение между центром полигонометрической марки и центром пункта, над которым она установлена; θ_1 — угол при центре марки между направлениями на центр данного пункта и сигнал, с которого определяют азимут.

Элементы центрировки и редукции определяют перед началом наблюдений и записывают в журнал наблюдений (табл. 17). Поправки в азимут вычисляют по формулам:

$$\Delta a_l = \frac{l}{S}(\rho \sin \theta);$$

$$\Delta a_r = \frac{l_1}{S}(\rho \sin \theta_1);$$

$$\Delta a_\gamma = l \sin a\,|2|\,\mathrm{tg}\,\varphi,$$

где S — расстояние в метрах (из каталога) между центрами пунктов; a — азимут направления теодолит — центр пункта; $[2] \approx 0{,}032255''$ — поправка на 1 м сечения первого вертикала.

Азимут, приведенный к центру пункта, определяют по формуле

$$A = A_0 + \Delta a_l + \Delta a_r + \Delta a_\gamma.$$

Подготовка исходных данных. Ввод исходных данных для решения задачи на ЭВМ может осуществляться как в диалоговом режиме, так и с помощью заранее подготовленного файла исходных данных непосредственно из журналов наблюдений (см. табл. 17). Дополнительные исходные данные (табл. 18) готовятся заранее на основании приема сигналов точного времени и с помощью астрономического ежегодника (АЕ). Программа предусматривает обработку десяти приемов.

Таблица 18
Бланк ввода дополнительных исходных данных

Описание исходных данных	Обозначение	Числовое значение
Номер часового пояса места наблюдения	n	2
Часовой пояс подачи радиосигналов	$n_п$	2
Поправка за переход от всемирного времени к эфемеридному (из АЕ)	$\Delta T(A)$	50,0ˢ
Склонение Солнца на дату наблюдения	δ_0	$+23°20'41{,}9''$
Часовые изменения склонения:		
на дату	v_0	$+5{,}42''$
на следующую дату	v_1	$+4{,}39''$
Уравнение времени $+12^h$ на дату	E_0	$11^h59^m27{,}59ˢ$
Часовые изменения величины E:		
на дату	v_0	$-0{,}539ˢ$
на следующую дату	v_1	$-0{,}542ˢ$

Во время работы программы промежуточные результаты выводятся на экран, а результаты, необходимые для дальнейшего использования, выводятся на печать.

Тестовая задача. Исходные данные для решения тестовой задачи берут из табл. 17 и 18. Диалог с программой в процессе решения тестовой задачи приведен в табл. 19. Результаты решения тестовой задачи печатаются в следующей форме:

ОПРЕДЕЛЕНИЕ АЗИМУТА НАПРАВЛЕНИЯ НА ЗЕМНОЙ ПРЕДМЕТ ПО ЧАСОВЫМ УГЛАМ СОЛНЦА
ШИРОТА МЕСТА 55 54 12
ДОЛГОТА МЕСТА 2 32 21.
РЕЗУЛЬТАТ ОБРАБОТКИ 1-ГО ПРИЕМА:
ЧАСОВОЙ ХОД ХРОНОМЕТРА (СЕКУНДЫ В ЧАС) = —1.500.
ЗНАЧЕНИЕ ПОГРЕШНОСТИ ПРИВЕДЕНИЯ НАБЛЮДЕНИЙ К ЦЕНТРУ СОЛНЦА (СЕКУНДЫ) = —9.33 .
ЗНАЧЕНИЕ АСТРОНОМИЧЕСКОГО АЗИМУТА ИЗ 1-ГО ПРИЕМА (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)
274 47 15.4 .
ОКОНЧАТЕЛЬНОЕ ЗНАЧЕНИЕ АЗИМУТА (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)
274 47 15.4.
СКП ОПРЕДЕЛЕНИЯ АЗИМУТА ОДНИМ ПРИЕМОМ (СЕКУНДЫ), 0 .
ВЕРОЯТНЕЙШЕЕ ЗНАЧЕНИЕ СКП НА ПУНКТЕ (СЕКУНДЫ), 0 .
АЗИМУТ, ПРИВЕДЕННЫЙ К ЦЕНТРУ ПУНКТА (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)
275 26 53.9
НАБЛЮДАЛ: БОНДАРЕНКО А. М.
ДАТА: 6 ИЮНЯ 1980 Г.

Таблица 19
Диалог с программой вычисления азимута по часовым углам Солнца

Сообщение или запрос программы	Ответ пользователя
ВЫЧИСЛЕНИЕ АЗИМУТА ЗЕМНОГО ПРЕДМЕТА ПО ЧАСОВЫМ УГЛАМ СОЛНЦА.	
ВЫВОД РЕЗУЛЬТАТОВ НА ЭКРАН — 0, НА ПЕЧАТЬ — 1	1
ВВЕДИТЕ ФАМИЛИЮ	БОНДАРЕНКО А. М
ВВЕДИТЕ ДАТУ НАБЛЮДЕНИЯ	6 ИЮНЯ 1980
ВВЕДИТЕ ЧИСЛО ОБРАБАТЫВАЕМЫХ ПРИЕМОВ В ДАННОЙ ПРОГРАММЕ ЧИСЛО ОБРАБАТЫВАЕМЫХ ПРИЕМОВ НЕ БОЛЕЕ 10.	6
ВВЕДИТЕ НОМЕР ЧАСОВОГО ПОЯСА МЕСТА НАБЛЮДЕНИЯ	2
ВВЕДИТЕ НОМЕР ЧАСОВОГО ПОЯСА ПОДАЧИ РАДИОСИГНАЛОВ	2
ВВЕДИТЕ ШИРОТУ МЕСТА: ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ	55,54,12
ВВЕДИТЕ ДОЛГОТУ МЕСТА: ЧАСЫ, МИНУТЫ, СЕКУНДЫ	2,32,21.0

Продолжение табл. 19

Сообщение или запрос программы	Ответ пользователя
ВВЕДИТЕ, ЕСЛИ ЗИМНЕЕ ВРЕМЯ, ТО 1, ЕСЛИ ЛЕТНЕЕ,—2	2
ВВОД ДАННЫХ ПЕРВОГО ПРИЕМА.	
ВВЕСТИ ВРЕМЯ ПОДАЧИ РАДИОСИГНАЛА ДО НАБЛЮДЕНИЙ: ЧАСЫ	6
ВВЕСТИ ВРЕМЯ ПОДАЧИ РАДИОСИГНАЛА ПОСЛЕ НАБЛЮДЕНИЙ: ЧАСЫ	8
ВВЕСТИ ВРЕМЯ ПО ХРОНОМЕТРУ ДО НАБЛЮДЕНИЙ: ЧАСЫ, МИНУТЫ, СЕКУНДЫ	5,59,00
ВВЕСТИ ВРЕМЯ ПО СЕКУНДОМЕРУ ДО НАБЛЮДЕНИЙ: МИНУТЫ, СЕКУНДЫ	1,15.5
ВВЕСТИ ВРЕМЯ ПО ХРОНОМЕТРУ ПОСЛЕ НАБЛЮДЕНИЙ: ЧАСЫ, МИНУТЫ, СЕКУНДЫ	7,58,00
ВВЕСТИ ВРЕМЯ ПО СЕКУНДОМЕРУ ПОСЛЕ НАБЛЮДЕНИЙ: МИНУТЫ, СЕКУНДЫ	2,18.5
ЕСЛИ ОШИБКА, НАБЕРИТЕ 1; ЕСЛИ НЕТ — 0	0
ВВЕСТИ ВРЕМЯ ПО ХРОНОМЕТРУ ПРИ I-ОМ НАБЛЮДЕНИИ СОЛНЦА (ЧАСЫ, МИНУТЫ, СЕКУНДЫ):	1—6,57,00 2—6,58,00 3—7,00,00 4—7,02,00
ВВЕСТИ ВРЕМЯ ПО СЕКУНДОМЕРУ ПРИ I-ОМ НАБЛЮДЕНИИ СОЛНЦА (МИНУТЫ, СЕКУНДЫ):	1—0,35.0 2—1,05.0 3—0,50.0 4—1,09.0
ЕСЛИ ОШИБКА, НАБЕРИТЕ 1; ЕСЛИ НЕТ,—0	0
ВВЕСТИ ОТСЧЕТ НА МАРКУ ПРИ КЛ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)	149,02,12.0
ВВЕСТИ ОТСЧЕТ НА МАРКУ ПРИ КП (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)	329,02,27.0
НЕДОПУСТИМОЕ РАСХОЖДЕНИЕ ОТСЧЕТОВ НА МАРКУ.	
ВВЕСТИ РЕЗУЛЬТАТЫ 1-ГО, 2-ГО НАБЛЮДЕНИЙ СОЛНЦА (КЛ) И 3-ГО, 4-ГО (КП) ПО ГОРИЗОНТАЛЬНОМУ КРУГУ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ):	1—304,19,40.0 2—305,12,31.5 3—124,57,33.5 4—126,00,15.0
ЕСЛИ ОШИБКА, НАБЕРИТЕ 1; ЕСЛИ НЕТ,— 0	0
ОШИБКА ИСХОДНЫХ ДАННЫХ.	
ВЫЧИСЛЕНИЕ КООРДИНАТ СОЛНЦА.	
ВВЕСТИ ПОПРАВКУ К ЭФЕМЕРИДНОМУ ВРЕМЕНИ (СЕКУНДЫ)	50
ВВЕСТИ 1, 2 — КООРДИНАТУ СОЛНЦА НА ДАТУ НАБЛЮДЕНИЯ: 1 — ДЛЯ СКЛОНЕНИЯ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ): 2 — ДЛЯ ВЕЛИЧИНЫ Е (ЧАСЫ, МИНУТЫ, СЕКУНДЫ)	1—23,20,41.9
ВВЕСТИ 1 — ЧАСОВЫЕ ИЗМЕНЕНИЯ НА ДАТУ НАБЛЮДЕНИЯ (СЕКУНДЫ): 1 — ДЛЯ СКЛОНЕНИЯ; 2 — ДЛЯ ВЕЛИЧИНЫ Е	5,42
ВВЕСТИ 1 — ЧАСОВЫЕ ИЗМЕНЕНИЯ НА СЛЕДУЮЩУЮ ДАТУ(СЕКУНДЫ):	

Продолжение табл. 19

Сообщение или запрос программы	Ответ пользователя
1 ДЛЯ СКЛОНЕНИЯ; 2 ДЛЯ ВЕЛИЧИНЫ Е:	4,39 2—11,59,27.590 —0.539 —0.542
ЕСЛИ ОШИБКА, НАБЕРИТЕ 1; ЕСЛИ НЕТ,— 0	0
ВЫЧИСЛЕНИЕ АЗИМУТА НАПРАВЛЕНИЯ НА ЗЕМНОЙ ПРЕДМЕТ.	
РЕЗУЛЬТАТЫ ОБРАБОТКИ 1-ГО ПРИЕМА.	
ЧАСОВОЙ ХОД ХРОНОМЕТРА (СЕКУНДЫ В ЧАС) РАВЕН —1.499.	
ПОГРЕШНОСТЬ ПРИВЕДЕНИЯ НАБЛЮДЕНИЙ К ЦЕНТРУ СОЛНЦА (СЕКУНДЫ) РАВНА —9.3.	
СРЕДНИЙ МОМЕНТ НАБЛЮДЕНИЙ (ЧАСЫ, МИНУТЫ, СЕКУНДЫ) РАВЕН 6 59 52.8.	
ЭФЕМЕРИДНОЕ ВРЕМЯ (ЧАСЫ, МИНУТЫ, СЕКУНДЫ) РАВНО 3 00 42.8.	
ВСПОМОГАТЕЛЬНАЯ ВЕЛИЧИНА Е (ЧАСЫ, МИНУТЫ, СЕКУНДЫ) РАВНА 11 59 25.970.	
ЧАСОВОЙ УГОЛ ИСТИННОГО СОЛНЦА (ЧАСЫ, МИНУТЫ, СЕКУНДЫ) РАВЕН 17 31 39.58.	
АЗИМУТ СОЛНЦА (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ) РАВЕН 70 52 26.2.	
АСТРОНОМИЧЕСКИЙ АЗИМУТ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ) 274 47 15.6.	
ОКОНЧАТЕЛЬНОЕ ЗНАЧЕНИЕ АЗИМУТА (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ) 274 47 16.	
СКП ОПРЕДЕЛЕНИЯ АЗИМУТА ОДНИМ ПРИЕМОМ (СЕКУНДЫ) РАВНА 4.12.	
ВЕРОЯТНЕЙШЕЕ ЗНАЧЕНИЕ СКП НА ПУНКТЕ (СЕКУНДЫ) РАВНО 1.7.	
ПРИВЕДЕНИЕ АЗИМУТА К ЦЕНТРУ ПУНКТА.	
1. ЭЛЕМЕНТЫ ЦЕНТРИРОВКИ.	
ВВЕСТИ ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ МЕЖДУ ЦЕНТРОМ ТЕОДОЛИТА И ЦЕНТРОМ АСТРОНОМИЧЕСКОГО ПУНКТА (МЕТРЫ)	5,8
ВВЕСТИ УГОЛ МЕЖДУ НАПРАВЛЕНИЯМИ НА ЦЕНТР АСТРОНОМИЧЕСКОГО ПУНКТА И НА СИГНАЛ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)	49,30,00
2. ЭЛЕМЕНТЫ РЕДУКЦИИ.	
ВВЕСТИ ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ МЕЖДУ ЦЕНТРОМ ВИЗИРНОЙ МАРКИ И ЦЕНТРОМ ГЕОДЕЗИЧЕСКОГО ПУНКТА (МЕТРЫ)	0.05
ВВЕСТИ УГОЛ ПРИ ЦЕНТРЕ ВИЗИРНОЙ МАРКИ МЕЖДУ НАПРАВЛЕНИЯМИ НА ЦЕНТР ДАННОГО ПУНКТА И АСТРОНОМИЧЕСКИЙ ПУНКТ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)	68,45.00
ВВОД ДОПОЛНИТЕЛЬНЫХ ЭЛЕМЕНТОВ	
ВВЕСТИ АЗИМУТ НАПРАВЛЕНИЯ (ТЕОДОЛИТ — АСТР. ПУНКТ) (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)	220,30,00
ВВЕСТИ РАССТОЯНИЕ МЕЖДУ ЦЕНТРАМИ ПУНКТОВ (МЕТРЫ)	386.5
ЕСЛИ ОШИБКА, НАБЕРИТЕ 1; ЕСЛИ НЕТ — 0.	
ПОПРАВКА ЗА ЦЕНТРИРОВКУ ТЕОДОЛИТА (СЕКУНДЫ) РАВНА 2353.7.	
ПОПРАВКА ЗА РЕДУКЦИЮ ВИЗИРНОЙ ЦЕЛИ (СЕКУНДЫ) РАВНА 24.9.	

Продолжение табл. 19

Сообщение или запрос программы	Ответ пользователя
ПОПРАВКА ЗА СБЛИЖЕНИЕ МЕРИДИАНОВ (СЕКУНДЫ) РАВНА 0.2. АЗИМУТ, ПРИВЕДЕННЫЙ К ЦЕНТРУ ПУНКТА (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ), РАВЕН 275 26 54. ШИРОТА МЕСТА (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ) РАВНА 55 54 12. ДОЛГОТА МЕСТА (ЧАСЫ, МИНУТЫ, СЕКУНДЫ) РАВНА 2 32 21. НАБЛЮДАЛ: БОНДАРЕНКО А. М. ДАТА: 6 ИЮНЯ 1980	

Вычисление азимута по часовым углам Полярной звезды. В процессе выполнения азимутальных наблюдений в каждом приеме помощник производит вычисления и контроль. Дальнейшая обработка журнала наблюдений и проведение последующих вычислений осуществляется в следующем порядке.

Вычисление поправок u^* и хода звездного хронометра ω^*:

$$u^* = M_п(1+\mu) + S_0 + \lambda_E - X^*;$$

$$\omega^* = \frac{u_2^* - u_1^*}{X_2^* - X_1^*},$$

где $X^* = T_{пс}^* + T_{скм} + (T_{скм})''' 0,164$;

$\mu = 0{,}0027379$ — коэффициент перехода от среднего времени к звездному; S_0 — истинное звездное время в гринвичскую полночь, т. е. на 0^h всемирного времени (выбирается из таблиц АЕ по дате); $M_п$ — момент подачи сигналов проверки времени (СПВ) по шкале всемирного времени; X^* — показание хронометра в момент приема радиосигналов времени.

Формула определения поправок до и после наблюдений предусматривает перевод моментов подачи радиосигналов времени из декретной шкалы в местную звездную. Если ход звездного хронометра в пределах допуска, то осуществляют перенос поправки хронометра на каждый момент наблюдений T^*:

$$s = T^* + u^*;$$

$$T^* = T_{пс}^* + T_{скм} + (T_{скм})''' 0,164;$$

$$u^* = u_1^* + \omega^*(T^* - X_1),$$

где s — исправленный момент наблюдений по шкале местного звездного времени.

Далее проводят интерполирование координат звезды на средний момент наблюдений s_m в следующем порядке:

определяют гринвичское звездное время момента наблюдения

$$S_m = s_m - \lambda_E;$$

по одной из формул вычисляют гринвичский часовой угол:

$T = S_m - \alpha_0$, если $S_0 \to \alpha_0 \to S_m$, или

$T = -(\alpha_0 - S_m)$, если $S_0 \to S_m \to \alpha_0$,

и интерполяционный множитель в долях суток $H=T/24$;

пропорционально аргументу интерполирования вычисляют изменение координат по прямому восхождению v_α и склонению v_δ:

$$v_\alpha=\Delta\alpha H;$$

$$v_\delta=\Delta\delta H,$$

где $\Delta\alpha$ и $\Delta\delta$ — изменение координат за сутки; затем считают видимые координаты звезды на момент наблюдений

$$\delta=\delta_0+v_\delta;$$

$$\alpha=\alpha_0+v_\alpha,$$

где α_0 и δ_0 — табличные значения координат в момент верхней кульминации на Гринвиче.

Отсчеты по горизонтальному кругу (КЛ* и КП*) на звезду исправляют поправкой за наклон горизонтальной оси прибора Δb, которую вычисляют по формулам:

$$\Delta b=b\ \mathrm{ctg}\ z\cdot\tau/2;$$

$$b=(i_1+i_2)/2,$$

где $i=|(\text{Л}+\text{П})-2m|$,— если нуль шкалы уровня справа; $i=|2m-(\text{Л}+\text{П})|$,— если нуль слева; $i=\text{Л}+\text{П}$ — если нуль на середине; b — наклон горизонтальной оси прибора в полуделениях; τ — цена деления накладного уровня; z — зенитное расстояние Полярной звезды выбирается из рабочих эфемерид или вычисляется по отсчету вертикального круга L* с учетом места зенита MZ; i — наклоны оси уровня, полученные из отсчетов Л и П по концам пузырька уровня (до и после перекладки при КЛ и КП); $2m$ — число делений шкалы уровня.

По исправленным отсчетам

$$\text{КЛ}=\text{КЛ}^*+\Delta b;$$

$$\text{КП}=\text{КП}^*+\Delta b$$

выполняют полевой контроль измерений; незамыкание при КЛ и КП не должно превышать 4″, двойная коллимационная ошибка не должна быть более 8″. Кроме этого, отклонение нуль-пункта уровня x не должно превышать трех делений, т. е.

$$x=(i_1-i_2)/2\leqslant 3\ \text{дел.}$$

После проведения контроля вычисляют непосредственно азимут Полярной звезды a^* по формуле

$$\mathrm{tg}\ a^*=\mathrm{m}\sin t/(1-\mathrm{n}),$$

где $\mathrm{m}=-\sec\varphi\ \mathrm{ctg}\ \delta$;

$$\mathrm{n}=\mathrm{tg}\ \varphi\ \mathrm{ctg}\ \delta\cos t;$$

$$t=s_m-\alpha.$$

Правильность определения четверти для вычисления азимута звезды проверяется как и в предыдущей задаче (по знаку $\mathrm{tg}\ a^*$ и по ве-

личине часового угла), имея в виду, что звезда может находиться к востоку или к западу от северного направления меридиана.

Азимут направления A на земной предмет вычисляют с учетом значения места севера по формулам:

$$MN = N^* - a^*;$$

$$A_0 = N - MN,$$

где N^* и N — средние отсчеты по горизонтальному кругу в направлении на звезду и земной предмет.

Полученное значение азимута исправляют поправками за влияние суточной аберрации δA и за азимутальное ускорение ΔA_W, которые вычисляют по формулам:

$$\delta A = 0{,}32 \cos\varphi \operatorname{cosec} z \cos a^*;$$

$$\Delta A_W = 5{,}454 \left(\frac{\Delta T}{100}\right)^2 \frac{d^2A}{dt^2};$$

$$\frac{d^2A}{dt^2} = K_1 \sin a^* - K_2 \sin 2a^*;$$

$$K_1 = 0{,}5 \sin 2\varphi \operatorname{ctg} z;$$

$$K_2 = 0{,}5 \cos^2\varphi (\operatorname{ctg}^2 z + \operatorname{cosec}^2 z);$$

$$\Delta T = T_m - T_{КЛ,\ КП},$$

где T_m — средний момент наблюдений Полярной звезды в приеме.

Окончательное значение азимута земного предмета

$$A = A_0 + \delta A + \Delta A_W.$$

Таблица 20

Журнал наблюдений Полярной звезды

Пункт: *АП «Чкаловская»* Теодолит: *ТТ 5/10 № 14639* MZ *0°00′15,5″*
λ $2^h32^m21{,}0^s$ Хронометр: *Nardin № 220* 2*m 35*
φ *55°54′12″* Секундомер: *«Златоуст» № 11170* τ *4,57″*
Дата: *15/16.06.86* L* *33°55′00″*
Наблюдал: Столяров А. И.

Номер наблюдения	Объект наблюдений	Моменты наблюдений					Отсчет по уровню		Отсчет по горизонтальному кругу		
		по хронометру			по секундомеру		Л	П	°	′	″
		h	m	s	m	s					
										КЛ	
	ОРП-1								105	22	15.1
1	*αUMi*	17	45	00,0	0	40,0	6,3	30,0	180	10	59.5
2	*αUMi*	17	52	00,0	0	43,5	29,9	6,0	180	12	41.4
	ОРП-1								105	22	18.5
										КП	
	ОРП-1								285	22	08.6
3	*αUMi*	18	12	00,0	0	13,5	29,0	5,8	0	14	24.5
4	*αUMi*	18	15	00,0	0	29,5	7,9	31,1	0	15	03.3
	ОРП-1								285	222	12.6

Контроль: $2C_1 = 6{,}5''$; $2C_2 = 5{,}9''$; $\Delta NЛ = 3{,}4'$; $\Delta NП = 4{,}0''$; $x_1 = -1{,}1$ дел.; $x_2 = 1{,}9$ дел.

Таблица 21
Бланк ввода дополнительных исходных данных

Описание исходных данных	Обозначение	Числовое значение
Номер часового пояса места наблюдения	n	2
Коэффициент равен нулю, если прием сигналов времени в пределах суток; в противном случае и если $T_2 < T_1$, то коэффициент равен 1, 2, 3 и т. д.	$k_{1,2}$	0
Время подачи радиосигналов:		
до наблюдений	D_1	23^h
после наблюдений	D_2	3^h
Звездное время в гринвичскую полночь:		
на дату наблюдений	S_{01}	$17^h33^m36{,}8^s$
на следующую дату	S_{02}	17 37 33,3
Время по хронометру в момент пуска секундомера:		
до наблюдений	$T_{нс.1}$	15 05 00,0
после наблюдений	$T_{нс.2}$	19 07 00,0
Отсчеты по секундомеру:		
до наблюдений	$T_{скм.2}$	2 44,2
после наблюдений	$T_{скм.2}$	1 23,0
Коэффициент равен единице, если момент наблюдений $T_i < T_{нс.1}$, или наблюдение проводили более, чем через сутки после приема радиосигналов	$k_{срв}$	0
Прямое восхождение звезды на дату	α_0	$2^h08^m26{,}14^s$
Изменение координаты за сутки	$\Delta\alpha$	$+1{,}185^s$
Склонение звезды на дату	δ_0	$+89°09'08{,}1''$
Изменение координаты за сутки	$\Delta\delta$	$-0{,}18$
Зенитное расстояние звезды, измеренное $Z = L^* - MZ$ или вычисленное по АЕ $Z = (90° - \varphi) - \delta$	Z	$33°54'44{,}5''$

Подготовка исходных данных. Ввод исходных данных осуществляется аналогично предыдущей задаче — из журнала наблюдений Полярной звезды (табл. 20) и бланка дополнительных исходных данных (табл. 21). Программа предусматривает обработку одного приема наблюдений с предварительным вычислением часового хода звездного хронометра.

Тестовая задача. Диалог с программой приведен в табл. 22.

Таблица 22
Диалог с программой вычисления азимута по часовым углам Полярной звезды

Сообщение или запрос программы	Ответ пользователя
ОПРЕДЕЛЕНИЕ АЗИМУТА НАПРАВЛЕНИЯ НА ЗЕМНОЙ ПРЕДМЕТ ПО ЧАСОВЫМ УГЛАМ ПОЛЯРНОЙ.	
ВЫВОД РЕЗУЛЬТАТОВ НА ПРИНТЕР — 1, НЕТ — 0	1
ВВЕДИТЕ ФАМИЛИЮ НАБЛЮДАТЕЛЯ	СТОЛЯРОВ А. И.
ВВЕДИТЕ ДАТУ НАБЛЮДЕНИЙ	15/16 июня 1986
ПРИ ОШИБКЕ НАЖМИТЕ ⟨СУ⟩, ⟨С⟩ И НАБЕРИТЕ GOTO 70.	
ВВЕДИТЕ ДОЛГОТУ ТОЧКИ СТОЯНИЯ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)	2,32,21.0
ВВЕДИТЕ ШИРОТУ ТОЧКИ СТОЯНИЯ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)	55,54,12
ВВЕДИТЕ НОМЕР ЧАСОВОГО ПОЯСА МЕСТА НАБЛЮДЕНИЯ	2

Продолжение табл. 22

Сообщение или запрос программы	Ответ пользователя
ВВЕДИТЕ, ЕСЛИ ЗИМНЕЕ ВРЕМЯ — 1; ЕСЛИ ЛЕТНЕЕ — 2	2
ВВЕДИТЕ КОЭФФИЦИЕНТ К1.2	0
ПРИ ОШИБКЕ НАЖМИТЕ ⟨СУ⟩, ⟨С⟩ И GOTO 190.	
ВВЕДИТЕ ВРЕМЯ 1-ОЙ ПОДАЧИ РАДИОСИГНАЛОВ В ЧАСАХ	23
ВВЕДИТЕ ЗВЕЗДНОЕ ВРЕМЯ SØ1 (ЧАСЫ, МИНУТЫ, СЕКУНДЫ)	17,33,36.8
ВВЕДИТЕ ВРЕМЯ 2-ОЙ ПОДАЧИ РАДИОСИГНАЛОВ В ЧАСАХ	3
ВВЕДИТЕ ЗВЕЗДНОЕ ВРЕМЯ SØ2 (ЧАСЫ, МИНУТЫ, СЕКУНДЫ)	17,37,33.3
ВВЕДИТЕ ВРЕМЯ 1-ГО ПУСКА СЕКУНДОМЕРА (ЧАСЫ, МИНУТЫ, СЕКУНДЫ)	15,05,00.0
ВВЕДИТЕ ОТСЧЕТ ПО СЕКУНДОМЕРУ (МИНУТЫ, СЕКУНДЫ)	2,44.2
ВВЕДИТЕ ВРЕМЯ 2-ГО ПУСКА СЕКУНДОМЕРА (ЧАСЫ, МИНУТЫ, СЕКУНДЫ)	19,07,00.0
ВВЕДИТЕ ОТСЧЕТ ПО СЕКУНДОМЕРУ (МИНУТЫ, СЕКУНДЫ)	1,23.0
ОШИБКА ВЫЧИСЛЕНИЯ ХОДА ХРОНОМЕТРА.	
ЧАСОВОЙ ХОД ЗВЕЗДНОГО ХРОНОМЕТРА РАВЕН 0.199.	
ВВЕДИТЕ МОМЕНТ НАБЛЮДЕНИЯ ЗВЕЗДЫ:	
КЛ	
1 Т-ПС (ЧАСЫ, МИНУТЫ, СЕКУНДЫ)	17,45,00
Т-СКМ (МИНУТЫ, СЕКУНДЫ)	0,40.0
2 Т-ПС (ЧАСЫ, МИНУТЫ)	17,52,00
Т-СКМ (МИНУТЫ, СЕКУНДЫ)	0,43.5
КП	
3 Т-ПС (ЧАСЫ, МИНУТЫ, СЕКУНДЫ)	18,12,00
Т-СКМ (МИНУТЫ, СЕКУНДЫ)	0,13,5
4 Т-ПС (ЧАСЫ, МИНУТЫ, СЕКУНДЫ)	18,15,00
Т-СКМ (МИНУТЫ, СЕКУНДЫ)	0,29.5
ПРИ ОШИБКЕ НАЖМИТЕ ⟨СУ⟩ ⟨С⟩ И НАБЕРИТЕ GOTO 380.	
ВВЕДИТЕ КОЭФФИЦИЕНТ К-СРВ	0
ВВЕДИТЕ:	
ПРЯМОЕ ВОСХОЖДЕНИЕ (ЧАСЫ, МИНУТЫ, СЕКУНДЫ)	2,08,25.4
ИЗМЕНЕНИЕ КООРДИНАТЫ ЗА СУТКИ (СЕКУНДЫ)	1.1
СКЛОНЕНИЕ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)	89,09,08.1
ИЗМЕНЕНИЕ КООРДИНАТЫ ЗА СУТКИ (СЕКУНДЫ)	0.2
ПРИ ОШИБКЕ НАЖМИТЕ ⟨СУ⟩, ⟨С⟩ И НАБЕРИТЕ GOTO 578.	
ВВЕДИТЕ:	
ЗЕНИТНОЕ РАССТОЯНИЕ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ)	33,54,44.5
ЦЕНУ ДЕЛЕНИЯ УРОВНЯ ТАУ	4.57
ДЛИНУ ШКАЛЫ 2М	35
ВВЕДИТЕ ОТСЧЕТ ПО УРОВНЮ ДЛЯ:	
КЛ	
Л	6.3
П	30.0
Л	29.9
П	6.0
КП	
Л	29.0

Продолжение табл. 22

Сообщение или запрос программы	Ответ пользователя
П	5.8
Л	7.9
П	31.1
ПРИ ОШИБКЕ НАЖМИТЕ ⟨СУ⟩, ⟨С⟩ И НАБЕРИТЕ GOTO 947.	
НЕДОПУСТИМЫЕ РАСХОЖДЕНИЯ ПОЛОЖЕНИЯ НУЛЬ-ПУНКТА УРОВНЯ.	
ВВЕДИТЕ ОТСЧЕТ НА ЗВЕЗДУ ПО ГОРИЗОНТАЛЬНОМУ КРУГУ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ):	
ПРИ КЛ	
1	180,10,59.5
2	180,12,41.4
ПРИ КП	
3	0,14,24.5
4	0,15,03.3
ПРИ ОШИБКЕ НАЖМИТЕ ⟨СУ⟩, ⟨С⟩ И НАБЕРИТЕ GOTO 1150.	
ВВЕДИТЕ ОТСЧЕТЫ НА ЗЕМНОЙ ПРЕДМЕТ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ):	
ПРИ КЛ	
1	105,22,15.1
2	105,22,24.4
ПРИ КП	
3	285,22,08.6
4	285,22,20.2
ПРИ ОШИБКЕ НАЖМИТЕ ⟨СУ⟩, ⟨С⟩ И НАБЕРИТЕ GOTO 1400.	
НЕДОПУСТИМАЯ ПОГРЕШНОСТЬ НЕЗАМЫКАНИЯ.	
НЕДОПУСТИМАЯ КОЛЛИМАЦИОННАЯ ПОГРЕШНОСТЬ.	
АЗИМУТ РАВЕН 286 ГРАДУСОВ 25 МИН 25.7 С	
НАБЛЮДАЛ: СТОЛЯРОВ А. И.	
ДАТА: 15/16 ИЮНЯ 1986	
Результаты решения тестовой задачи печатаются в следующей форме:	
ОПРЕДЕЛЕНИЕ АЗИМУТА НАПРАВЛЕНИЯ НА ЗЕМНОЙ ПРЕДМЕТ ПО ЧАСОВЫМ УГЛАМ ПОЛЯРНОЙ.	
ДОЛГОТА МЕСТА 2 ЧАСА 32 МИН 21.0 С	
ШИРОТА МЕСТА 55 ГРАДУСОВ 54 МИН 12 С	
ЧАСОВОЙ ХОД ЗВЕЗДНОГО ХРОНОМЕТРА 0.198 С/ЧАС	
АЗИМУТ=288 ГРАДУСОВ 25 МИН 22.2 С	
НАБЛЮДАЛ: СТОЛЯРОВ А. И.	
ДАТА: 15/16.06.86	

2.2. Уравнивание систем теодолитных ходов

Назначение программы. Программа предназначена для вычисления координат точек планового обоснования в виде систем теодолитных ходов и может быть использована для уравнивания полигонометрических сетей, если для этих сетей допустимо раздельное уравнивание. По программе выполняется контроль полевых измерений, уравнивание дирекционных углов узловых линий, абсцисс и ординат узловых точек, вычисление координат точек ходов между узловыми пунктами и оценка точности измерений.

Одиночный теодолитный ход, являющийся частным случаем системы ходов, также может быть обработан по данной программе.

Размеры уравниваемых сетей определяются возможностями используемой машины.

Способ решения. По результатам угловых измерений вычисляют приближенные дирекционные углы узловых линий и находят невязки ходов, которые сравнивают с допустимыми. Если невязка превышает допуск, программа выдает сообщение об этом и в зависимости от принятого пользователем решения может либо продолжить вычисления с недопустимыми невязками, либо прекратить работу.

Уравнивание дирекционных углов узловых линий выполняется параметрическим способом. Неизвестными при этом являются поправки в дирекционные углы узловых линий, а измеренными величинами — суммы горизонтальных углов по ходам. За вес хода принимается величина, обратно пропорциональная числу углов в ходе.

После уравнивания дирекционных углов в каждом ходе выполняется распределение суммарной угловой поправки и вычисление сумм приращений координат.

По суммам приращений координат и координатам твердых точек находят приближенные координаты узловых точек и вычисляют невязки в приращениях по каждому ходу, которые сравнивают с допустимыми. При недопустимых невязках в приращениях выполняют те же действия, что и при недопустимых угловых невязках.

Уравнивание абсцисс и ординат узловых точек выполняется параметрическим способом. Неизвестными являются поправки в абсциссы и ординаты, а измеренными величинами считаются суммы приращений координат по ходам. За вес хода принимается величина, обратно пропорциональная длине хода.

После вычисления уравненных значений координат узловых точек выполняется вычислительная обработка каждого хода сети, состоящая в распределении суммарных поправок по отдельным приращениям (прямо пропорционально длинам сторон) и вычислении координат точек ходов по исправленным приращениям и уравненным координатам узловых точек.

Для оценки точности измерений вычисляют следующие величины:

среднюю квадратическую погрешность измерения угла

$$m_\beta = \sqrt{[p_\beta v_\beta^2]/r};$$

среднюю квадратическую погрешность единицы веса для абсцисс и ординат

$$\mu_x = \sqrt{[p_x v_x^2]/r};$$

$$\mu_y = \sqrt{[p_y v_y^2]/r};$$

среднюю квадратическую погрешность положения узловых точек

$$M = \sqrt{\mu_x^2 q_{i,i} + \mu_y^2 q_{i+1,\,i+1}}.$$

В этих формулах введены следующие обозначения: r — число избыточных измерений при уравнивании; p_β, p_x, p_y — веса ходов при урав-

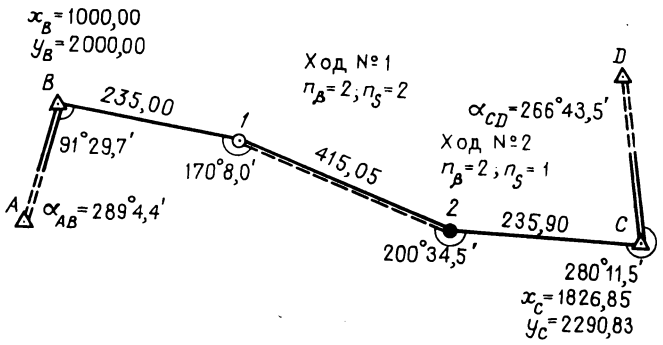


Рис. 25. Схема теодолитного хода к тестовой задаче для программы «Уравнивание систем теодолитных ходов»

нивании углов, абсцисс и ординат ($p_x = p_y$); v_β, v_x, v_y — суммарные поправки ходов при уравнивании углов, абсцисс и ординат; q — элементы матрицы обратных весов, соответствующие оцениваемому узловому пункту.

Подготовка исходных данных. Составляют схему сети (рис. 25), намечают узловые линии. В произвольном порядке (но так, чтобы не было номера равного —1) нумеруют твердые точки и точки, принадлежащие узловым линиям. Нумерация остальных точек ходов не имеет значения. При выдаче результатов решения программа нумерует промежуточные точки каждого хода подряд, начиная с номера 1.

На схему выписывают горизонтальные углы и горизонтальные проложения сторон, а также твердые исходные данные. Горизонтальные углы могут быть и правыми, и левыми, но для одного и того же хода все углы должны быть либо правыми, либо левыми. Углы могут быть заданы до минут или до секунд.

Твердыми исходными данными являются координаты твердых точек и дирекционные углы твердых направлений. Вместо последних можно задавать координаты точек, составляющих твердое направление, тогда дирекционный угол будет вычислен программным путем.

Для работы программы необходимо, чтобы хотя бы один из ходов, идущих от твердых пунктов, имел примычный угол.

Точки в конце висячих ходов должны быть объявлены узловыми (с выбором узловой линии при них).

Все ходы на схеме должны быть пронумерованы числами от 1 до t, где t — число ходов. Для каждого хода должно быть указано направление и выписаны величины n_β и n_s, где n_β — число горизонтальных углов, участвующих в передаче дирекционного угла от одной узловой линии к другой, n_s — число сторон хода между узловыми точками.

Для работы программы необходимо подготовить четыре массива информации.

1-й массив (общая информация о задаче) включает в себя следующие величины:

число твердых точек, для которых задаются координаты;

число определяемых узловых точек;

число ходов в сети;

признак того, с какой точностью заданы горизонтальные углы (признак равен 1, если углы заданы до секунд, и нулю,— если до минут).

2-й массив содержит характеристики точности измерений, которые используются при оценке допустимости невязок:

среднюю квадратическую погрешность измерения угла (в секундах);

знаменатель допустимой относительной ошибки измерения сторон (или величину постоянной ошибки измерения сторон в сантиметрах).

3-й массив состоит из номеров и координат точек, идущих в следующем порядке:

номер твердой точки;

x и y твердой точки и т. д. для всех твердых точек, координаты которых будут использоваться при работе программы;

номера определяемых узловых точек;

номера точек, являющихся концами узловых или твердых направлений (имеются в виду только те точки, номера которых не были названы выше).

3-й массив должен заканчиваться величиной «—1», вводимой в ответ на очередной запрос о номере точки.

4-й массив составляет информация о ходах, которая для каждого хода включает в себя следующие величины:

номер хода;

номера первой и второй точек названия начального дирекционного угла;

номера первой и второй точек названия конечного дирекционного угла;

число углов n_β;

число сторон n_s;

признак правых или левых углов (0 — признак правых, 1 — левых);

величину начального дирекционного угла $\alpha_{нач}$;

величину конечного дирекционного угла $\alpha_{кон}$; если величины $\alpha_{нач}$ или $\alpha_{кон}$ неизвестны, то вводится «—1».

Далее следуют горизонтальные углы, а за ними — горизонтальные проложения сторон.

Признаком конца информации 4-го массива является $\emptyset$, вводимый в ответ на очередной запрос о номере хода.

Тестовая задача и диалог. Исходные данные для решения тестовой задачи готовятся в соответствии со схемой, приведенной на рис. 25. Здесь приведен одиночный теодолитный ход с точкой *2*, названной узловой, и узловой линией *1—2*.

1-й массив информации: *2* — число твердых точек с заданными координатами; 1 — число определяемых узловых точек; 2 — число ходов; 0 — углы заданы до минут.

2-й массив информации: 60 — средняя квадратическая погрешность измерения угла (в с); 2000 — знаменатель допустимой относительной ошибки измерения сторон.

3-й массив — названия и координаты точек: *B* — название твердой точки; 1000 и 2000 — x и y точки *B*; *C* — название твердой точки;

1826.85 и 2290.83 — x и y точки C; 2 — номер узловой точки; A, 1, D — названия точек, являющихся концами твердых и узловой линий и неуказанных выше; —1 — признак конца 3-го массива.

4-й массив: 1 — номер хода; A, B — название $\alpha_{нач}$; 1, 2 — название $\alpha_{кон}$; 2 — число углов; 2 — число сторон; 0 — углы правые; 289 и 4.4 — $\alpha_{нач}$; —1 — $\alpha_{кон}$; 91 и 29.7 — β_1; 170 и 8 — β_2; 235 — S_1; 415.05 — S_2; 2 — номер хода; 1, 2 — название $\alpha_{нач}$; C, D — название $\alpha_{кон}$; 2 — число углов; 1 — число сторон; 0 — углы правые; —1 — $\alpha_{нач}$; 266 и 43,5 — $\alpha_{кон}$; 200 и 34.5 — β_1; 280 и 11.5 — β_2; 235.9 — S_1; 0 — признак конца 4-го массива.

Диалог с программой в процессе решения тестовой задачи приведен в табл. 23. Результаты решения тестовой задачи приведены в табл. 24, 25.

Таблица 23
Диалог с программой «Уравнивание систем теодолитных ходов»

Сообщение или запрос программы	Ответ пользователя
УРАВНИВАНИЕ СИСТЕМ ТЕОДОЛИТНЫХ ХОДОВ.	
ВВЕДИТЕ ЧИСЛО ТВЕРДЫХ ТОЧЕК, КООРДИНАТЫ КОТОРЫХ БУДУТ ИСПОЛЬЗОВАТЬСЯ ПРИ РЕШЕНИИ ЗАДАЧИ	2
ВВЕДИТЕ ЧИСЛО ОПРЕДЕЛЯЕМЫХ УЗЛОВЫХ ТОЧЕК	1
ВВЕДИТЕ ЧИСЛО ХОДОВ	2
ВВЕДИТЕ 1, ЕСЛИ УГЛЫ ЗАДАЮТСЯ ДО СЕКУНД; 0,— ЕСЛИ ДО МИНУТ	0
ХОТИТЕ ПОВТОРИТЬ ВВОД? (ДА/НЕТ)	НЕТ
ВВЕДИТЕ СР. КВАДР. ПОГРЕШНОСТЬ ИЗМЕРЕНИЯ УГЛА (В СЕКУНДАХ)	60
ВВЕДИТЕ ЗНАМЕНАТЕЛЬ ДОПУСТИМОЙ ОТНОСИТЕЛЬНОЙ ПОГРЕШНОСТИ ИЗМЕРЕНИЯ СТОРОН (ИЛИ ВЕЛИЧИНУ ПОСТОЯННОЙ ПОГРЕШНОСТИ ИЗМЕРЕНИЯ СТОРОН, В САНТИМЕТРАХ)	2000
ХОТИТЕ ВНЕСТИ ИЗМЕНЕНИЯ? (ДА/НЕТ)	НЕТ
ВВОД НАЗВАНИЙ И КООРДИНАТ ТВЕРДЫХ ТОЧЕК.	
ВВЕДИТЕ НАЗВАНИЕ ТОЧКИ	B
ВВЕДИТЕ X И Y ТОЧКИ	1000
?	2000
ВВЕДИТЕ НАЗВАНИЕ ТОЧКИ	C
ВВЕДИТЕ X И Y ТОЧКИ	1826.85
?	2290.83
ВВОДИТЕ НАЗВАНИЯ ОПРЕДЕЛЯЕМЫХ УЗЛОВЫХ ТОЧЕК.	
?	2
ВВОДИТЕ НАЗВАНИЯ ТОЧЕК, ЯВЛЯЮЩИХСЯ КОНЦАМИ ТВЕРДЫХ ИЛИ УЗЛОВЫХ ЛИНИЙ, ЕСЛИ ОНИ НЕ БЫЛИ ВВЕДЕНЫ ВЫШЕ.	
ЕСЛИ ТАКИХ НАЗВАНИЙ НЕТ (ИЛИ УЖЕ ВСЕ ВВЕДЕНЫ), ВВЕДИТЕ —1.	
ВВЕДИТЕ НАЗВАНИЕ (ИЛИ —1)	A
ВВЕДИТЕ НАЗВАНИЕ (ИЛИ —1)	1
ВВЕДИТЕ НАЗВАНИЕ (ИЛИ —1)	D
ВВЕДИТЕ НАЗВАНИЕ (ИЛИ —1)	—1
ПРОВЕРЬТЕ:	

Продолжение табл. 23

Сообщение или запрос программы	Ответ пользователя

Номер П/П	НАЗВАНИЕ ТОЧКИ	КООРДИНАТЫ ТОЧЕК, М	
		X	Y
1.	B	1000.00	2000.00
2.	C	1826.85	2290.83
3.	A		
4.	I		
5.	D		

ХОТИТЕ ИСПРАВИТЬ? (ДА/НЕТ)	НЕТ
ВВОДИТЕ ИНФОРМАЦИЮ О ХОДАХ.	
ВВЕДИТЕ НОМЕР ХОДА (ИЛИ 0, ЕСЛИ ВСЯ ИНФОРМАЦИЯ ВВЕДЕНА)	1
НАЗВАНИЕ НАЧАЛЬНОГО ДИРЕКЦИОННОГО УГЛА:	
ПЕРВАЯ ТОЧКА	A
ВТОРАЯ ТОЧКА	B
НАЗВАНИЕ КОНЕЧНОГО ДИРЕКЦИОННОГО УГЛА: ПЕРВАЯ ТОЧКА	1
ВТОРАЯ ТОЧКА	2
ЧИСЛО УГЛОВ В ХОДЕ	2
ЧИСЛО СТОРОН В ХОДЕ	2
0, ЕСЛИ УГЛЫ ПРАВЫЕ; 1,— ЕСЛИ ЛЕВЫЕ	0
ВВЕДИТЕ НАЧАЛЬНЫЙ ДИРЕКЦИОННЫЙ УГОЛ (—1, ЕСЛИ УГОЛ НЕИЗВЕСТЕН, ИЛИ ГРАДУСЫ, МИНУТЫ)	289
?	4,4
ВВЕДИТЕ КОНЕЧНЫЙ ДИРЕКЦИОННЫЙ УГОЛ (—1, ЕСЛИ УГОЛ НЕИЗВЕСТЕН, ИЛИ ГРАДУСЫ, МИНУТЫ)	—1
ВВОДИТЕ ИЗМЕРЕННЫЕ ГОРИЗОНТАЛЬНЫЕ УГЛЫ.	
1-Й УГОЛ (ГРАДУСЫ, МИНУТЫ)	91
?	29.7
2-Й УГОЛ (ГРАДУСЫ, МИНУТЫ)	170
?	8
ВВОДИТЕ ГОРИЗОНТАЛЬНЫЕ ПРОЛОЖЕНИЯ СТОРОН (В МЕТРАХ).	
1-Е ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ	235
2-Е ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ	415.05
ИНФОРМАЦИЯ ПО ХОДУ 1 ВВЕДЕНА	
ОШИБКИ ИСПРАВИТЕ ПО ОКОНЧАНИИ ВВОДА ВСЕЙ ИНФОРМАЦИИ.	
ПРОДОЛЖАЙТЕ ВВОД.	
ВВЕДИТЕ НОМЕР ХОДА (ИЛИ 0, ЕСЛИ ВСЯ ИНФОРМАЦИЯ ВВЕДЕНА)	2
НАЗВАНИЕ НАЧАЛЬНОГО ДИРЕКЦИОННОГО УГЛА: ПЕРВАЯ ТОЧКА	1
ВТОРАЯ ТОЧКА	2
НАЗВАНИЕ КОНЕЧНОГО ДИРЕКЦИОННОГО УГЛА: ПЕРВАЯ ТОЧКА	C
ВТОРАЯ ТОЧКА	D
ЧИСЛО УГЛОВ В ХОДЕ	2
ЧИСЛО СТОРОН В ХОДЕ	1
0, ЕСЛИ УГЛЫ ПРАВЫЕ; 1,— ЕСЛИ ЛЕВЫЕ	0
ВВЕДИТЕ НАЧАЛЬНЫЙ ДИРЕКЦИОННЫЙ УГОЛ (—1, ЕСЛИ УГОЛ НЕИЗВЕСТЕН, ИЛИ ГРАДУСЫ, МИНУТЫ)	—1
ВВЕДИТЕ КОНЕЧНЫЙ ДИРЕКЦИОННЫЙ УГОЛ (—1, ЕСЛИ	

Продолжение табл. 23

Сообщение или запрос программы	Ответ пользователя
УГОЛ НЕИЗВЕСТЕН, ИЛИ ГРАДУСЫ, МИНУТЫ) ? ВВОДИТЕ ИЗМЕРЕННЫЕ ГОРИЗОНТАЛЬНЫЕ УГЛЫ. 1-Й УГОЛ (ГРАДУСЫ, МИНУТЫ) ? 2-Й УГОЛ (ГРАДУСЫ, МИНУТЫ) ? ВВОДИТЕ ГОРИЗОНТАЛЬНЫЕ ПРОЛОЖЕНИЯ СТОРОН (В МЕТРАХ). 1-Е ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ ИНФОРМАЦИЯ ПО ХОДУ 2 ВВЕДЕНА. ОШИБКИ ИСПРАВИТЕ ПО ОКОНЧАНИИ ВВОДА ВСЕЙ ИНФОРМАЦИИ. ПРОДОЛЖАЙТЕ ВВОД. ВВЕДИТЕ НОМЕР ХОДА (ИЛИ 0, ЕСЛИ ВСЯ ИНФОРМАЦИЯ ВВЕДЕНА) ПРОВЕРЬТЕ ИСХОДНЫЕ ДАННЫЕ (на экран выдается вся введенная информация). ХОТИТЕ ВНЕСТИ ИСПРАВЛЕНИЯ? (ДА/НЕТ)	266 43.5 200 34.5 280 11.5 235.9 0 НЕТ

Таблица 24
Результаты решения тестовой задачи по программе «Уравнивание систем теодолитных ходов» (оценка точности)

ОЦЕНКА ТОЧНОСТИ								
НОМЕР ХОДА	ДЛИНА ХОДА, М	ЧИСЛО УГЛОВ	УГЛОВАЯ НЕВЯЗКА, С	НЕВЯЗКИ В ПРИРАЩЕНИЯХ, М		ОТНОСИТЕЛЬНАЯ ОШИБКА R	М НАЧ. УЗЛ. ТОЧКИ, М	М КОН. УЗЛ. ТОЧКИ, М
				FX	FY			
1	650	2	1.4	—0.29	—0.00	2277	0.00	0.17
2	236	2	1.4	—0.10	—0.00	2277	0.17	0.00

СРЕДНЯЯ КВАДРАТИЧЕСКАЯ ПОГРЕШНОСТЬ ИЗМЕРЕНИЯ УГЛА 84 С.
СРЕДНЯЯ КВАДРАТИЧЕСКАЯ ПОГРЕШНОСТЬ ЕДИНИЦЫ ВЕСА (НА 1 КМ):
ДЛЯ АБСЦИСС 0.41 М,
ДЛЯ ОРДИНАТ 0.01 М.

2.3. Уравнивание теодолитного хода без примычных углов

Назначение программы. По программе вычисляют координаты точек теодолитного хода, опирающегося в начале и конце на твердые точки, примычные углы на которых не измерены. Исходными данными для работы программы являются координаты начальной и конечной точек, горизонтальные проложения сторон и горизонтальные углы (левые или правые). Результаты счета — координаты точек хода, уравнен-

Таблица 25

Результаты решения тестовой задачи по программе «Уравнивание систем теодолитных ходов» (результаты уравнивания)

РЕЗУЛЬТАТЫ УРАВНИВАНИЯ					
НАЗВАНИЕ ТОЧКИ	ДИРЕКЦИОННЫЙ УГОЛ, ГРАДУСЫ, МИНУТЫ	ДЛИНА СТОРОНЫ, М	КООРДИНАТЫ, М		НАЗВАНИЕ ТОЧКИ
			X	Y	
A		ХОД № 1 (B—2)			A
	289 4,4				
B			1000.000	2000.00	B
	17 35.0	235.099			
1			1224.115	2071.019	1
	27 27.4	415,213			
2			1592.557	2262.468	2
		ХОД № 2 (2—C)			
1					1
2			1592.557	2262.468	2
	6 54.1	236.003			
C			1826.850	2290.830	C
	226 43.5				
D					D

ные дирекционные углы и длины сторон и величины абсолютной и относительной невязок хода.

Способ решения. По условному дирекционному углу 1-й стороны хода ($\alpha_1 = 0$), координатам начальной точки A (рис. 26) и результатам измерений величин β_i и S_i последовательно вычисляют условные координаты точек хода $1, 2, \ldots, n$ (где n — число сторон хода) по формулам:

$$X_i = X_{i-1} + mS_i \cos(\alpha_{i-1} + (\beta_{i-1} - 180°)k);$$

$$Y_i = Y_{i-1} + mS_i \sin(\alpha_{i-1} + (\beta_{i-1} - 180°)k), \quad (5)$$

где $i = 1, 2\ldots, n$; $m = 1$ — коэффициент масштабирования; $k = 1$, если

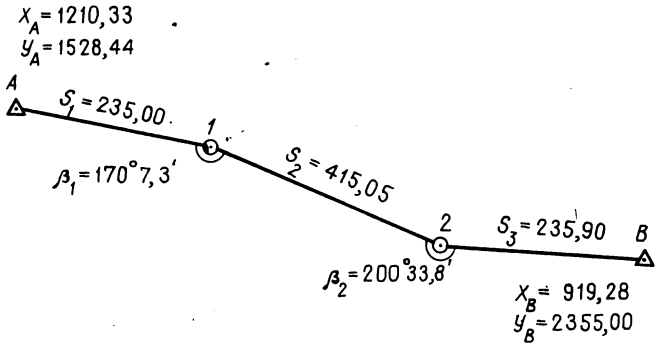


Рис. 26. Схема теодолитного хода к тестовой задаче для программы «Теодолитный ход без примычных углов»

измерены левые углы, $k=-1$, если измерены правые углы; $\beta_0=180°$; $\alpha_0=\alpha_1$; $X_0=X_A$; $Y_0=Y_A$.

По вычисленным условным координатам X_n, Y_n точки B находят угол разворота сети $\Delta\alpha$ и коэффициент масштабирования m по формулам:

$$m=\sqrt{(X_B-X_A)^2+(Y_B-Y_A)^2}\ /\ \sqrt{(X_n-X_A)^2+(Y_n-Y_A)^2};$$

$$\Delta\alpha=\operatorname{arctg}\ [(Y_B-Y_A)/(X_B-X_A)]-\operatorname{arctg}\ [(Y_n-Y_A)/(X_n-X_A)]+C,$$

где C — константа равная 180 или 360° в зависимости от знаков приращений координат.

Вычисления по формулам (5) повторяют с найденным значением m и величиной $\alpha_1=\Delta\alpha$; в результате получают уравненные значения координат точек хода. Контролем правильности решения является совпадение вычисленных координат точки B с заданными.

Подготовка исходных данных. Составляют схему сети. Точки хода на схеме должны быть пронумерованы так, как показано на рис. 26. На схему выписывают исходные данные — координаты твердых точек, горизонтальные углы и горизонтальные проложения сторон. Измеренные величины необходимо пронумеровать подряд, начиная с единицы, т. е. углы должны иметь номера *1*, *2*, ..., $n-1$, проложения — *1*, *2*, ..., n.

Тестовая задача и диалог. Исходные данные для решения тестовой задачи готовятся в соответствии со схемой, приведенной на рис. 26. Диалог с программой в процессе решения тестовой задачи приведен в табл. 26. Диалог предусматривает исправление ошибок, допущенных при вводе исходных данных. Результаты решения тестовой задачи выдаются в виде табл. 27.

2.4. Уравнивание комбинированных геодезических сетей. Решение засечек

Назначение программы. Программа предназначена для уравнивания и оценки точности геодезических сетей произвольной конфигурации, измеренными величинами в которых могут быть углы, направления, дирекционные углы и длины сторон. Допускается наличие линий и базисов, измеренных с разной точностью. Остальные измеренные величины приняты равноточными внутри каждого вида измерекий.

После уравнивания выполняется оценка точности заданных сторон, оцениваемые стороны могут быть заданы между любыми двумя точками сети.

Программа может быть использована для обработки сетей, в которых отсутствуют избыточные измерения. В этом случае при оценке точности в качестве средней квадратической погрешности единицы веса используется заданная средняя квадратическая погрешность.

Предусмотрено использование программы не только для уравнивания, но и для предрасчета точности проектируемых сетей. По этой же программе вычисляют координаты точек, определенных засечками любого вида (прямой, обратной, линейной, комбинированной, взаимно обратной — задачей Ганзена), выполняют оценку точности определе-

Таблица 26
Диалог с программой «Теодолитный ход без примычных углов»

Сообщение или запрос программы	Ответ пользователя
ТЕОДОЛИТНЫЙ ХОД БЕЗ ПРИМЫЧНЫХ УГЛОВ.	
ВВЕДИТЕ ЧИСЛО СТОРОН	3
ВВЕДИТЕ 1, ЕСЛИ ИЗМЕРЕНЫ ЛЕВЫЕ УГЛЫ; Ø — ЕСЛИ ПРАВЫЕ	Ø
ВВЕДИТЕ 1, ЕСЛИ УГЛЫ ЗАДАНЫ ДО СЕКУНД, Ø — ЕСЛИ ДО МИНУТ	Ø
ВВЕДИТЕ X НАЧАЛЬНОЙ ТОЧКИ	1210.33
ВВЕДИТЕ У НАЧАЛЬНОЙ ТОЧКИ	1528.44
ВВЕДИТЕ X КОНЕЧНОЙ ТОЧКИ	919.28
ВВЕДИТЕ У КОНЕЧНОЙ ТОЧКИ	2355
ВВОДИТЕ ГОРИЗОНТАЛЬНЫЕ УГЛЫ.	
ВВЕДИТЕ НОМЕР УГЛА (ИЛИ Ø, ЕСЛИ ВСЕ УГЛЫ ВВЕДЕНЫ)	1
ВВЕДИТЕ ВЕЛИЧИНУ УГЛА (ГРАДУСЫ, МИНУТЫ)	170
?	7.3
ВВЕДИТЕ НОМЕР УГЛА (ИЛИ Ø, ЕСЛИ ВСЕ УГЛЫ ВВЕДЕНЫ)	2
ВВЕДИТЕ ВЕЛИЧИНУ УГЛА (ГРАДУСЫ, МИНУТЫ)	200
?	33.8
ВВЕДИТЕ НОМЕР УГЛА (ИЛИ Ø, ЕСЛИ ВСЕ УГЛЫ ВВЕДЕНЫ	Ø
ВВОДИТЕ ГОРИЗОНТАЛЬНЫЕ ПРОЛОЖЕНИЯ)	
ВВЕДИТЕ НОМЕР СТОРОНЫ (ИЛИ Ø, ЕСЛИ ВСЕ СТОРОНЫ ВВЕДЕНЫ)	1
ВВЕДИТЕ ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ	235
ВВЕДИТЕ НОМЕР СТОРОНЫ (ИЛИ Ø, ЕСЛИ ВСЕ СТОРОНЫ ВВЕДЕНЫ)	2
ВВЕДИТЕ ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ	415.05
ВВЕДИТЕ НОМЕР СТОРОНЫ (ИЛИ Ø, ЕСЛИ ВСЕ СТОРОНЫ ВВЕДЕНЫ)	3
ВВЕДИТЕ ГОРИЗОНТАЛЬНОЕ ПРОЛОЖЕНИЕ	235.9
ВВЕДИТЕ НОМЕР СТОРОНЫ (ИЛИ Ø, ЕСЛИ ВСЕ СТОРОНЫ ВВЕДЕНЫ)	Ø
ПРОВЕРЬТЕ ВВЕДЕННУЮ ИНФОРМАЦИЮ (на экран выдаются все введенные исходные данные).	
ХОТИТЕ ИСПРАВИТЬ ОШИБКИ? (ДА/НЕТ)	НЕТ
ОТНОСИТЕЛЬНАЯ НЕВЯЗКА РАВНА 1/5255.	
ПРОДОЛЖИТЬ ВЫЧИСЛЕНИЯ? (ДА/НЕТ)	ДА

Таблица 27
Результаты решения тестовой задачи по программе «Теодолитный ход без примычных углов» (каталог координат)

НОМЕР ТОЧКИ	КООРДИНАТЫ, М		ДИРЕКЦИОННЫЙ УГОЛ	ДЛИНА СТОРОНЫ, М
	X	Y		
A	1210.33	1528.44		
			107°36.1′	235.05
1	1139.25	1752.48		
			117 28.8	415.13
2	947.69	2120.77		
			96 55.0	235.95
B	919.28	2355.00		

АБСОЛЮТНАЯ НЕВЯЗКА РАВНА 0,169 М,
ОТНОСИТЕЛЬНАЯ НЕВЯЗКА 1/5255.

ния положения точек, а при наличии избыточных измерений (многократной засечки) выполняют уравнивание и оценку точности по результатам уравнивания.

Размер сети, которая может быть обработана, зависит от используемой машины.

Способ решения. Уравнивание выполняется параметрическим способом, в качестве неизвестных выбраны поправки к приближенным координатам определяемых пунктов. Эти координаты вычисляют путем решения разного рода засечек.

Для измеренных в сети элементов составляют уравнения поправок с весами, вычисляемыми по формуле

$$p = c^2/m^2,$$

где c — средняя квадратическая погрешность единицы веса; m — средняя квадратическая погрешность элемента (угла, направления, длины стороны и т. п.), вес которого вычисляется.

Средние квадратические погрешности длин сторон m_S вычисляют в программе по различным формулам в зависимости от способа их измерения:

$m_S = \mu\sqrt{S}$, если стороны измерены проволокой или лентой и задан коэффициент случайного влияния μ;

$m_S = S/R$, если точность измерения длин сторон задана величиной относительной погрешности, знаменатель которой R;

$m_S = m_{S_1} + m_{S_2} 10^{-6} S$, если стороны измерены светодальномером; здесь m_{S_1} — постоянная часть погрешности, m_{S_2} — коэффициент при переменной части, S — длина стороны.

По коэффициентам уравнений поправок вычисляют матрицу A коэффициентов нормальных уравнений и вектор свободных членов L. Для решения системы нормальных уравнений $AX = L$ выполняют обращение матрицы A и вычисление вектора X поправок координат по формуле $X = QL$, где $Q = A^{-1}$.

Уравненные координаты определяемых точек вычисляют путем суммирования приближенных координат и поправок (элементов вектора X). Если максимальная поправка координат превышает допуск (в программе он принят равным 0,01 м), то процесс составления и решения нормальных уравнений повторяется с использованием вычисленных координат в качестве приближенных.

Для оценки точности сети вычисляют средние квадратические погрешности единицы веса, положения пунктов на концах оцениваемой стороны, дирекционного угла и знаменатель относительной средней квадратической погрешности длины оцениваемой стороны.

Подготовка исходных данных начинается с составления схемы сети (рис. 27).

Точки на схеме необходимо пронумеровать: порядок нумерации произвольный (не должно быть точек с одинаковыми номерами).

Исходными данными в сети могут быть координаты твердых точек, твердые дирекционные углы. При уравнивании сети, в которой отсутствуют точки с твердыми координатами, необходимо принять условно одну из точек за твердую. При отсутствии данных, обеспечивающих

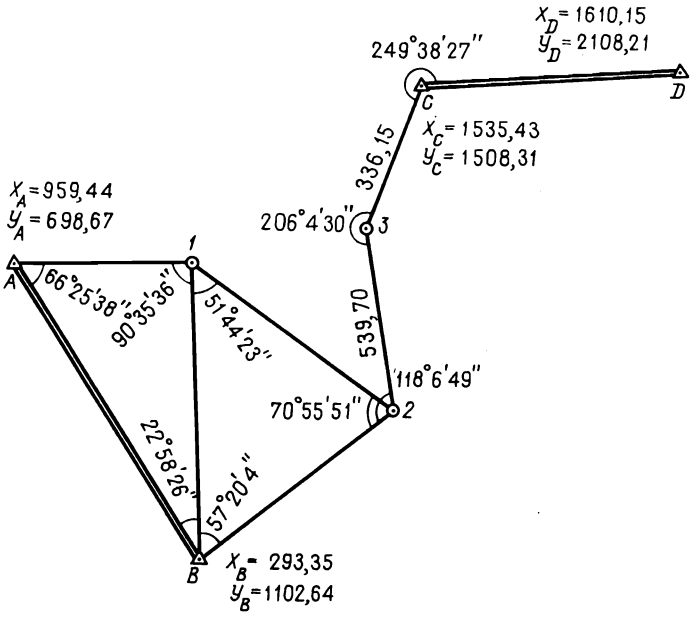


Рис. 27. Схема геодезической сети к тестовой задаче для программы «Уравнивание, решение засечек»

ориентирование сети, необходимо считать условный дирекционный угол какого-либо направления с твердой точки измеренным дирекционным углом, задав для него пренебрегаемо малую ошибку измерения.

На схему выписывают все твердые данные и результаты измерений.

Для ввода необходимо подготовить пять массивов информации.

1-й массив составляет, так называемая, общая информация о задаче, в нее входят следующие пять параметров:

признак P уравнивания сети или предрасчета точности; если необходимо выполнить уравнивание сети или решение засечек, то $P=1$, при предрасчете точности $P=0$;

число твердых точек T;

число определяемых точек K;

число твердых дирекционных углов D;

число всех измерений W;

число оцениваемых сторон O.

Дирекционный угол направления AB следует относить к числу твердых, если либо обе точки A и B имеют твердые координаты, либо одна из точек имеет твердые координаты, а координаты второй не используются при решении задачи.

2-й массив составляют следующие характеристики точности измерений, задаваемые для расчета весов:

m_β — средняя квадратическая погрешность измерения угла (в с):

$m_н$ — средняя квадратическая погрешность измерения направления (в с):

m_α — средняя квадратическая погрешность измерения дирекционного угла (в с):

μ_S — коэффициент случайного влияния при измерении длин сторон; если последние измерены светодальномером или точность линейных измерений характеризуется относительной погрешностью, то коэффициент случайного влияния должен быть задан равным нулю;

m_{S_1} — постоянная погрешность при светодальномерных измерениях; если длины сторон измерены проволокой или лентой, то величина m_{S_1} должна быть задана равной нулю;

m_{S_2} — коэффициент при переменной части (светодальномерные измерения); если точность линейных измерений характеризуется относительной погрешностью, то величина m_{S_2} должна быть задана равной R, где R — знаменатель относительной погрешности;

μ_b, m_{b1}, m_{b2} — величины, аналогичные трем предыдущим, характеризующие точность измерения базисов b.

Средние квадратические погрешности измерений, которых нет в сети, должны быть заданы равными нулю.

3-й массив информации составляют названия и координаты точек, вводимые в следующем порядке:

название твердой точки, X твердой точки, Y твердой точки и т. д. для всех T твердых точек;

если требуется выполнить предрасчет точности, то далее следует:

название определяемой точки, X и Y определяемой точки и т. д. для всех K определяемых точек;

если же требуется уравнять сеть или решить засечки, то вводят только названия (без координат) определяемых точек.

4-й массив готовится при наличии в сети твердых дирекционных углов и включает для каждого из D углов его название (номера двух точек по направлению измерения угла) и величину.

5-й массив включает в себя информацию об измерениях. Его удобно рассматривать состоящим из подмассивов, в которых объединяются измерения одного вида, выполненные с одной и той же точки.

Состав подмассива:

1) название точки, с которой выполнены измерения, входящие в подмассив;

2) вид измерений (задается условным числом): 1 — углы, 2 — направления, 3 — дирекционные углы, 4 — линии, 5 — базисы;

3) число измерений данного вида, выполненных с данной точки.

Перечисленные три величины составляют как бы заголовок подмассива; далее идут сами измерения, объявленные в заголовке:

a) название точки, на которую выполнено измерение; если измерениями являются углы, то это должно быть название левой точки угла;

b) название правой точки угла; если измерениями являются не углы, то данный элемент массива отсутствует;

c) результат измерения; если измерения угловые, то вводятся три величины (градусы, минуты, секунды), если линейные,— одна величина (длина стороны); если выполняется предрасчет точности, т. е.

результатов измерений нет, то данный элемент массива отсутствует.

Далее идут аналогичные группы (a, b, c), пока не будет исчерпано заявленное в заголовке подмассива число измерений.

Если измерениями являются направления, то точки, на которые измерены направления должны следовать по ходу часовой стрелки, и первой должна быть указана точка с нулевым направлением.

За первым подмассивом следует второй, третий и т. д., пока не будут перечислены все измеренные в сети элементы.

Пятый массив может включать и, так называемую, дополнительную информацию. Она готовится в двух случаях.

1. Если в сети есть точки, координаты которых определяются однократной линейной засечкой, то в этом случае задача имеет два решения; при отсутствии дополнительной информации программа выдаст два решения.

Дополнительная информация содержит сведения о числе точек, определяемых однократной линейной засечкой, для каждой из таких точек должны быть следующие данные:

название точки;

название твердого направления, с точек которого выполнена засечка;

положение точки относительно этого направления (справа или слева).

2. Если в сети есть точки, определяемые взаимно обратной засечкой (задачей Ганзена), то в этом случае дополнительная информация включает в себя число задач Ганзена и название пар точек, определяемых данным способом.

Названия (номера) точек, определяемых взаимно обратной засечкой должны указываться в пятом массиве последними.

6-й массив составляют названия сторон, которые необходимо оценить. Данный массив может отсутствовать, если число оцениваемых сторон задано равным нулю. Массив включает номера начальных и конечных точек сторон, каждый номер вводится отдельно.

Тестовая задача и диалог. Исходные данные для решения тестовой задачи готовятся в соответствии со схемой, приведенной на рис. 27:

1-й массив: 1 — необходимо выполнить уравнивание сети; 4 — число твердых точек; 3 — число определяемых точек; 0 — число твердых дирекционных углов; 11 — число измерений; 2 — число оцениваемых сторон.

2-й массив: 15 — средняя квадратическая погрешность измерения угла (в с); 0 — направлений в сети нет; 0 — дирекционных углов в сети нет; 0 — точность линейных измерений задана величиной относительной ошибки, поэтому $\mu_S=0$; 0 — точность линейных измерений задана относительной ошибкой, поэтому $m_{S_1}=0$; 3000 — относительная ошибка линейных измерений 1/3000; 0 0 0 — базисных измерений нет.

3-й массив. Названия и координаты твердых точек:

A 959.44 698.67
B 293.35 1102.64
C 1535.43 1508.31
D 1610.15 2108.21;

1, 2, 3 — номера определяемых точек.

4-й массив отсутствует.

5-й массив:

A 1 1 — заголовок 1-го подмассива: измерения с точки *A*, вид измерений — углы (число 1), количество измерений — 1;
1 — левая точка угла;
B — правая точка угла;
66 25 38 — градусы, минуты и секунды угла;
B 1 2 — заголовок 2-го подмассива: измерения с точки *B*,
вид измерений — углы, количество — 2;
A — левая точка угла;
1 — правая точка угла;
22 58 26 — градусы, минуты и секунды угла;
1 — левая точка угла;
2 — правая точка угла;
57 20 4 — градусы, минуты и секунды угла;
Далее без пояснений приведена информация по пяти подмассивам:

```
1  1  2    2 B  51  44  23    B A   90  35  36
2  1  2    B 1  70  55  51    B 3  118   6  49
3  1  1    2 C 206   4  30
C  1  1    3 Д 249  38  27
3  4  2    2 539.7   C  336.15.
```

Дополнительной информации нет.

6-й массив: 1 2 2 3 — здесь указано, что оценить нужно сторону *1—2* и сторону *2—3*.

Диалог в процессе решения тестовой задачи приведен в табл. 28. Результаты решения тестовой задачи приведены в табл. 29, 30.

Таблица 28
Диалог с программой «Уравнивание, решение засечек»

Сообщение или запрос программы	Ответ пользователя
УРАВНИВАНИЕ, РЕШЕНИЕ ЗАСЕЧЕК..	
ВВЕДИТЕ: 1, ЕСЛИ НУЖНО РЕШИТЬ ЗАСЕЧКИ ИЛИ УРАВНЯТЬ СЕТЬ; 0, — ЕСЛИ ВЫПОЛНИТЬ ПРЕДРАСЧЕТ ТОЧНОСТИ	1
ЧИСЛО ТВЕРДЫХ ТОЧЕК	4
ЧИСЛО ОПРЕДЕЛЯЕМЫХ ТОЧЕК	3
ЧИСЛО ТВЕРДЫХ ДИРЕКЦИОННЫХ УГЛОВ	0
ЧИСЛО ИЗМЕРЕНИЙ	11
ЧИСЛО ОЦЕНИВАЕМЫХ СТОРОН	2
ХОТИТЕ ПОВТОРИТЬ ВВОД? (ДА/НЕТ)	НЕТ
ВВЕДИТЕ ХАРАКТЕРИСТИКИ ТОЧНОСТИ В ПОСЛЕДОВАТЕЛЬНОСТИ, ЗАДАННОЙ ИНСТРУКЦИЕЙ:	15
?	0
?	0
?	0
?	0
?	3 000
?	0
?	0

Продолжение табл. 28

Сообщение или запрос программы	Ответ пользователя
?	0
ХОТИТЕ ПОВТОРИТЬ ВВОД? (ДА/НЕТ)	НЕТ
ВВОД НОМЕРОВ И КООРДИНАТ ТВЕРДЫХ ТОЧЕК.	
ВВЕДИТЕ НОМЕР ТОЧКИ	A
ВВЕДИТЕ X И Y ТОЧКИ	959.44
?	698.67
ВВЕДИТЕ НОМЕР ТОЧКИ	B
ВВЕДИТЕ X И Y ТОЧКИ	293.35
?	1102.64
ВВЕДИТЕ НОМЕР ТОЧКИ	C
ВВЕДИТЕ X И Y ТОЧКИ	1535.43
?	1508.31
ВВЕДИТЕ НОМЕР ТОЧКИ	D
ВВЕДИТЕ X И Y ТОЧКИ	1610.15
?	2108.21
ВВЕДИТЕ НОМЕР ОПРЕДЕЛЯЕМОЙ ТОЧКИ	1
ВВЕДИТЕ НОМЕР ОПРЕДЕЛЯЕМОЙ ТОЧКИ	2
ВВЕДИТЕ НОМЕР ОПРЕДЕЛЯЕМОЙ ТОЧКИ	3
ПРОВЕРЬТЕ:	

НОМЕР П/П	НАЗВАНИЕ ТОЧКИ	КООРДИНАТЫ ТОЧЕК, М	
		X	Y
1.	A	959.44	698.67
2.	B	293.35	1102.64
3.	C	1535.43	1508.31
4.	D	1610,15	2108.21
5.	1	—	—
6.	2	—	—
7.	3	—	—

ХОТИТЕ ИСПРАВИТЬ? (ДА/НЕТ)	НЕТ
ВВОД ИНФОРМАЦИИ ОБ ИЗМЕРЕНИЯХ.	
ВВЕДИТЕ НОМЕР ТОЧКИ, ВИД ИЗМЕРЕНИЙ (1 — УГЛЫ, 2 — НАПРАВЛЕНИЯ, 3 — ДИРЕКЦИОННЫЕ УГЛЫ, 4 — ЛИНИИ, 5 — БАЗИСЫ) И КОЛИЧЕСТВО ИЗМЕРЕНИЙ	A
?	1
?	1
ВВЕДИТЕ НАЗВАНИЯ ИЗМЕРЕННЫХ ВЕЛИЧИН И РЕЗУЛЬТАТЫ ИЗМЕРЕНИЙ.	
ЛЕВАЯ ТОЧКА	1
ПРАВАЯ ТОЧКА УГЛА	B
ВВЕДИТЕ ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ	66
?	25
?	38
ВВЕДИТЕ НОМЕР ТОЧКИ, ВИД ИЗМЕРЕНИЙ (1 — УГЛЫ, 2 — НАПРАВЛЕНИЯ, 3 — ДИРЕКЦИОННЫЕ УГЛЫ, 4 — ЛИНИИ, 5 — БАЗИСЫ) И КОЛИЧЕСТВО ИЗМЕРЕНИЙ	B
?	1
?	2
Далее аналогичным образом вводится информация об измерениях, приведенная в 5-м массиве. Пропустив продолжение ввода 2-го подмассива, а также 3—6-й подмассивы, приведем ввод последнего 7-го подмассива (измерения линий с точки *3*).	
ВВЕДИТЕ НОМЕР ТОЧКИ, ВИД ИЗМЕРЕНИЙ (1 — УГЛЫ, 2 — НАПРАВЛЕНИЯ, 3 — ДИРЕКЦИОННЫЕ УГЛЫ, 4 — ЛИНИИ,	

Продолжение табл. 28

Сообщение или запрос программы	Ответ пользователя
5 — БАЗИСЫ) И КОЛИЧЕСТВО ИЗМЕРЕНИЙ	3
?	4
?	2
ВВЕДИТЕ НАЗВАНИЯ ИЗМЕРЕННЫХ ВЕЛИЧИН И РЕЗУЛЬТАТЫ ИЗМЕРЕНИЙ:	
ТОЧКА	2
ДЛИНА СТОРОНЫ	539.7
ТОЧКА	С
ДЛИНА СТОРОНЫ	336.15
БУДЕТЕ ВВОДИТЬ ДОПОЛНИТЕЛЬНУЮ ИНФОРМАЦИЮ? (ДА/НЕТ)	НЕТ
ВВЕДИТЕ НАЗВАНИЕ ОЦЕНИВАЕМОЙ СТОРОНЫ	1
?	2
ВВЕДИТЕ НАЗВАНИЕ ОЦЕНИВАЕМОЙ СТОРОНЫ	2
?	3
ПРОВЕРЬТЕ НАЗВАНИЯ ОЦЕНИВАЕМЫХ СТОРОН 1) 1—2 2) 2—3.	
БУДЕТЕ ИСПРАВЛЯТЬ ОШИБКИ? (ДА/НЕТ)	НЕТ
ПРОВЕРЬТЕ ИСХОДНЫЕ ДАННЫЕ: Осуществляется выдача на экран (или печать) массивов 1—5 исходной информации. Для тестовой задачи эта выдача завершается фразами: ЛИНЕЙНЫХ ЗАСЕЧЕК НЕТ. ЗАДАЧ ГАНЗЕНА НЕТ.	
ХОТИТЕ ВНЕСТИ ИСПРАВЛЕНИЯ? (ДА/НЕТ)	НЕТ

Таблица 29
Результаты решения тестовой задачи по программе «Уравнивание, решение засечек» (оценка точности)

	ОЦЕНКА ТОЧНОСТИ							
СТОРОНА I—J	MXI, СМ	MYI, СМ	MI, СМ	MXJ, СМ	MYJ, СМ	MJ, СМ	MA, С	R
1—2	0.99	2.33	2,53	2,63	1.68	3.12	9.6	30628
2—3	2.63	1.68	3.12	5.08	1.22	5.23	6.5	9469

СРЕДНЯЯ КВАДРАТИЧЕСКАЯ ПОГРЕШНОСТЬ ЕДИНИЦЫ ВЕСА 9.44 С.

2.5. Уравнивание типовых фигур триангуляции

Назначение программ. Программы раздела предназначены для математической обработки результатов измерений типовых фигур триангуляции, которые по точности их построений не выше второго разряда, т. е. измерения углов в этих системах выполнены со средней квадратической погрешностью в 10″ и грубее. Математическая обра-

Таблица 30
Результаты решения тестовой задачи по программе «Уравнивание, решение засечек» (каталог координат)

КАТАЛОГ КООРДИНАТ

НАЗВАНИЕ ТОЧКИ	КООРДИНАТЫ, М		ДИРЕКЦИОННЫЙ УГОЛ, ГРАДУСЫ, МИН, С		ДЛИНА СТОРОНЫ, М
	X	Y			
A	959.440	698.670	B	148 45 50.6	779.017
			1	82 20 5.7	304.106
B	293.350	1102.640			
			A	328 45 50.6	779.017
			1	351 44 25.2	714.059
			2	49 4 26.0	593.238
C	1535.430	1508.310			
			D	82 54 0.5	604.535
			3	193 15 41.3	336.194
D	1610.150	2108.210			
1	1000.002	1000,059			
			A	262 20 5.7	304.106
			B	171 44 25.2	714.059
			2	120 0 6.5	636.027
2	681.971	1550.864			
			B	229 4 26.0	593.238
			1	300 0 6.5	636.027
			3	347 11 15.6	539.666
3	1208.201	1431.189			
			C	13 15 41.3	336.194
			2	167 11 15.6	539.666

ботка по программам ведется с удержанием шести (верных) значащих цифр, т. е. координаты пунктов определяются с шестью значащими цифрами и по запросу выдаются на экран или печатаются на бумагу до сотых долей метра.

Программы рассчитаны на реальное число пунктов в любой из типовых фигур. Режим работы оператора с программой — диалоговый. Исходными данными служат приведенные к центрам пунктов углы, координаты исходных пунктов, длины исходных сторон и их дирекционные углы.

Исходные данные, как правило, готовятся по схеме сети.

Способ решения. В программах использована методика упрощенного уравнивания коррелатным способом с разделением условных уравнений на группы, что позволяет поэтапно контролировать вычисление свободных членов условий, возникающих в каждой из типовых фигур. О превышении допусков выдаются сообщения и работа программы прерывается. В этом случае требуется устранение ошибки и программа запускается заново.

Программа перед выдачей каталога координат сообщает о контрольном смещении (в метрах) одного из исходных пунктов. Координаты этого исходного пункта, полученные в результате вычислений, не должны отличаться от их заданного значения более чем на 5 см.

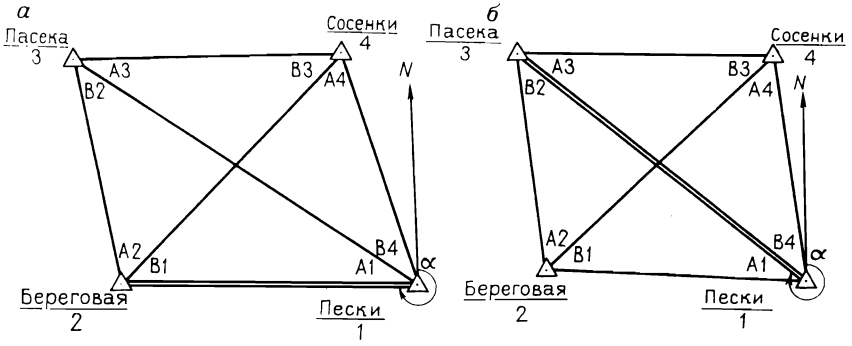


Рис. 28. Два варианта расположения базиса к программе «Геодезический четырехугольник» (1—4 — условные номера населенных пунктов)

2.5.1. Программа «Геодезический четырехугольник»

Схемы для составления программы приведены на рис. 28. Начало базиса — пункт Пески/1 (далее по ходу часовой стрелки). Угол A1 всегда расположен на сети у начала базиса. Базисными могут быть линии между пунктами 1—2 (рис. 28, *а*) и 1—3 (рис. 28, *б*).

Исходные данные для ввода:

1) средняя квадратическая погрешность измерения угла;
2) фактические названия пунктов вместо условных номеров, выдаваемых программой;
3) значения углов A1 и B1;
4) условный номер пункта конца базиса;
5) исходный дирекционный угол линии базиса от его начала;
6) длина базиса;
7) координаты исходных пунктов.

Тестовая задача и диалог. Исходные данные для решения тестовой задачи соответствуют схеме, приведенной на рис. 28:

1) M=10;
2) 1—Пески, 2—Береговая, 3—Пасека, 4—Сосенки;
3) A1=30,04,55 B1=63,09,59
   A2=62,51,06 B2=23,54,06
   A3=30,01,30 B3=63,13,12
   A4=53,09,15 B4=33,35,54

целесообразно заранее отделять минуты от градусов и секунды от минут запятой;

4) номер пункта конца базиса — 2;
5) дирекционный угол 283,13,50;
6) длина базиса 1107.18;
7) X(1)=7248.61; Y(1)=2286.23.

Координаты конца базиса можно не вводить, так как программой предусмотрено в этих случаях вычисление координат конца базиса.

Диалог в процессе решения тестовой задачи приведен в табл. 31, результаты решения приведены в табл. 32.

Таблица 31
Диалог с программой «Геодезический четырехугольник»

Сообщение или запрос программы	Ответ пользователя
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН, КУДА?	1
ГЕОДЕЗИЧЕСКИЙ ЧЕТЫРЕХУГОЛЬНИК (М НЕ МЕНЕЕ 10 СЕКУНД).	
ВВОД В СЕКУНДАХ СР.КВ.ПОГР. ИЗМЕРЕННОГО УГЛА?	10
ВВОД ФАКТИЧЕСКИХ НАЗВАНИЙ ПУНКТОВ (КАЖДОЕ ДО 10 БУКВ).	
УСЛОВНЫЕ: НАЧАЛО БАЗИСА — 1, ДАЛЕЕ ПО ХОДУ ЧАС. СТРЕЛКИ.	
УСЛОВНЫЙ НОМЕР 1, ФАКТИЧЕСКОЕ НАЗВАНИЕ?	ПЕСКИ
УСЛОВНЫЙ НОМЕР 2, ФАКТИЧЕСКОЕ НАЗВАНИЕ?	БЕРЕГОВАЯ
УСЛОВНЫЙ НОМЕР 3, ФАКТИЧЕСКОЕ НАЗВАНИЕ?	ПАСЕКА
УСЛОВНЫЙ НОМЕР 4, ФАКТИЧЕСКОЕ НАЗВАНИЕ?	СОСЕНКИ
ВВОД ЧИСЛОВЫХ ЗНАЧЕНИЙ УГЛОВ: A1, B1, ..., A4, B4.	
ФОРМАТ ВВОДА ЗНАЧЕНИЙ УГЛОВ (ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ И НАЖАТЬ ВК):	
A1?	30,04,55
B1?	63,09,59
A2?	62,51,06
B2?	23,54,06
A3?	30,01,30
B3?	63,13,12
A4?	53,09,15
B4?	33,35,54
НЕВЯЗКА В ЧЕТЫРЕХУГОЛЬНИКЕ РАВНА — 3 СЕКУНДАМ.	
ПОЛЮСНОЕ УСЛОВИЕ — 10 СЕКУНД.	
УСЛОВНЫЙ НОМЕР ПУНКТА НАЧАЛА БАЗИСА 1, КОНЦА — 2, ИЛИ 3.	
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА КОНЦА БАЗИСА: 1 — ?	2
ВВЕДИТЕ ИСХОДНЫЙ ДИРЕКЦ. УГОЛ ЛИНИИ 1 — 2?	283,13,50
ВВЕДИТЕ ДЛИНУ БАЗИСА В МЕТРАХ?	1107,18
ВВОД КООРДИНАТ ИСХОДНЫХ ПУНКТОВ.	
ВВЕДИТЕ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777)?	1
X1?	7248.61
Y1?	2286.23
ВВЕДИТЕ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777)?	777
КОНТР. СМЕЩ. ПУНКТА 1 В МЕТРАХ = .01	

2.5.2. Программа «Центральная система»

Программа предназначена для обработки центральной системы триангуляционной сети с различными вариантами расположения базиса (рис. 29). Условные номера пунктов вырабатывает программа. Центру всегда соответствует 0. Наименования остальных пунктов вводятся, начиная с номера 1 (начало базиса), по ходу часовой стрелки. Угол A1 всегда расположен у пункта начала базиса. Условный номер пункта конца базиса может быть 0 (рис. 29, *а*) или 2 (рис. 29, *б*).

Таблица 32
Результаты решения тестовой задачи по программе «Геодезический четырехугольник»

ГЕОДЕЗИЧЕСКИЙ ЧЕТЫРЕХУГОЛЬНИК

КАТАЛОГ КООРДИНАТ ПУНКТОВ (СКП ИЗМ. УГЛА 10 СЕКУНД)

НАЗВАНИЕ ПУНКТА	КООРДИНАТЫ, М		ДЛИНА СТОРОНЫ, М	ДИРЕКЦИОННЫЙ УГОЛ, ГРАДУСЫ, МИН, С	ПУНКТ НАПРАВЛЕНИЯ
	X	Y			
1. ПЕСКИ	7248.61	2286.23	1107.18	283 13 49	БЕРЕГОВАЯ
			2210.31	313 18 42	ПАСЕКА
			1234.55	346 54 36	СОСЕНКИ
2. БЕРЕГОВАЯ	7502.00	1208.44	1107.18	103 13 49	ПЕСКИ
			1369.71	337 12 46	ПАСЕКА
			1240.10	40 3 53	СОСЕНКИ
3. ПАСЕКА	8764.81	677.94	2210.31	133 18 42	ПЕСКИ
			1369.71	157 12 46	БЕРЕГОВАЯ
			1365.23	103 17 7	СОСЕНКИ
4. СОСЕНКИ	8451.08	2006.63	1234.55	166 54 36	ПЕСКИ
			1240.10	220 3 53	БЕРЕГОВАЯ
			1365.23	283 17 7	ПАСЕКА

Вводу подлежат следующие исходные данные:

1) средняя квадратическая погрешность измерения угла M;
2) число треугольников системы N;
3) фактические названия пунктов вместо условных номеров, выдаваемых программой;
4) значения углов A (I), B (I), C (I), где I=1, 2, ..., N;
5) условный номер пункта конца базиса;
6) исходный дирекционный угол линии базиса от его начала;
7) длина базиса;
8) координаты исходных пунктов.

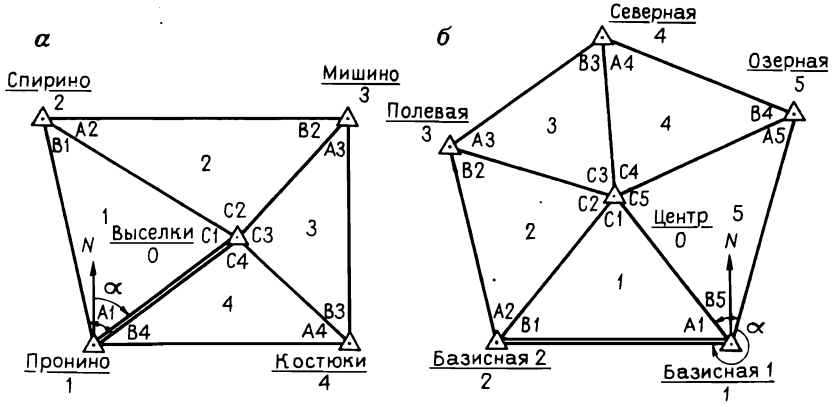


Рис. 29. Два варианта расположения базиса в центральной системе сети триангуляции к программе «Центральная система»

Тестовая задача и диалог. Исходные данные для решения тестовой задачи соответствуют рис. 29, *а*:

1) M = 10;

2) N = 4;

3) 0 — Выселки, 1 — Пронино, 2 — Спирино, 3 — Мишино, 4 — Костюки;

4) A1 = 81,10,16 B1 = 39,25,39 C1 = 59,23,55
A2 = 30,33,32 B2 = 54,43,03 C2 = 94,43,05
A3 = 41,11,33 B3 = 49,20,48 C3 = 89,27,26
A4 = 34,51,06 B4 = 28,43,30 C4 = 116,25,34

(целесообразно разделять минуты, градусы и секунды запятыми);

5) условный номер конца базиса — 0;

6) дирекционный угол 64,01,16;

7) длина базиса 1854.01;

8) X(1) = 4098.30; Y(1) = 0.00

(координаты конца базиса можно не вводить, программа в этом случае их определит).

Диалог в процессе решения тестовой задачи приведен в табл. 33, результаты ее решения — в табл. 34.

Таблица 33
Диалог с программой «Центральная система»

Сообщение или запрос программы	Ответ пользователя
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН, КУДА ?	1
ЦЕНТРАЛЬНАЯ СИСТЕМА ТРИАНГУЛЯЦИИ (М НЕ МЕНЕЕ 10 СЕКУНД).	
ВВОД В СЕКУНДАХ СР.КВ.ПОГР.ИЗМЕР.УГЛА (М) ?	10
ВВОД ЧИСЛА ТРЕУГОЛЬНИКОВ СИСТЕМЫ ?	4
ВВОД ФАКТИЧЕСКИХ НАЗВАНИЙ ПУНКТОВ (КАЖДОЕ ДО 10 БУКВ).	
УСЛОВНЫЕ — ЦЕНТР — 0, НАЧ.БАЗИСА — 1, ДАЛЕЕ ПО ХОДУ ЧАС.СТРЕЛКИ.	
УСЛОВНЫЙ НОМЕР 0, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	ВЫСЕЛКИ
УСЛОВНЫЙ НОМЕР 1, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	ПРОНИНО
УСЛОВНЫЙ НОМЕР 2, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	СПИРИНО
УСЛОВНЫЙ НОМЕР 3, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	МИШИНО
УСЛОВНЫЙ НОМЕР 4, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	КОСТЮКИ
ВВОД ЧИСЛОВЫХ ЗНАЧЕНИЙ УГЛОВ ПО ТРЕУГОЛЬНИКАМ.	
УГЛЫ ПРОТИВ СТОРОН: 1—A1—ПРОТИВ ОПРЕДЕЛЯЕМОЙ, 2—B1—ПРОТИВ ИСХОДНОЙ, 3—C1—ПРОТИВ ПРОМЕЖУТОЧНОЙ.	
ФОРМАТ ВВОДА ЗНАЧЕНИЙ УГЛОВ: ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ И НАЖАТЬ ВК.	
ДОПУСТИМАЯ УГЛОВАЯ НЕВЯЗКА В ТР-КЕ 40 СЕКУНД.	
ТРЕУГОЛЬНИК 1:	
УГОЛ A1 ?	81,10,16
УГОЛ B1 ?	39,25,39
УГОЛ C1 ?	59,23,55
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 1 РАВНА —10 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 2:	
УГОЛ A2 ?	30,33,32

Продолжение табл. 33

Сообщение или запрос программы	Ответ пользователя
УГОЛ В2 ?	54,43,03
УГОЛ С2 ?	94,43,05
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 2 РАВНА —20 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 3:	
УГОЛ А3 ?	41,11,33
УГОЛ В3 ?	49,20,48
УГОЛ С3 ?	89,27,26
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 3 РАВНА —13 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 4:	
УГОЛ А4 ?	34,51,06
УГОЛ В4 ?	28,43,30
УГОЛ С4 ?	116,25,34
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 4 РАВНА 10 СЕКУНДАМ.	
УСЛОВНЫЙ НОМЕР ПУНКТА НАЧ.БАЗИСА ВСЕГДА 1, КОНЦА — 2 ИЛИ 0.	
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА КОНЦА БАЗИСА: 1 — ?	0
ВВЕДИТЕ ИСХОДНЫЙ ДИРЕКЦИОННЫЙ УГОЛ ЛИНИИ 1—0?	64,01,16
ВВЕДИТЕ ДЛИНУ БАЗИСА В МЕТРАХ?	1854.01
ВВОД КООРДИНАТ ИСХОДНЫХ ПУНКТОВ.	
ВВЕДИТЕ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	1
X1 ?	4098.30
Y1 ?	0.00
ВВЕДИТЕ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	777
ПОЛЮСНОЕ УСЛОВИЕ — 55 СЕКУНД	
КОНТР.СМЕЩ. ПУНКТА 1 (В МЕТРАХ) — 0	

Таблица 34
Результаты решения тестовой задачи по программе «Центральная система»

ЦЕНТРАЛЬНАЯ СИСТЕМА

КАТАЛОГ КООРДИНАТ ПУНКТОВ (СКП ИЗМ.УГЛА 10 СЕКУНД)

НАЗВАНИЕ ПУНКТА	КООРДИНАТЫ, М		ДЛИНА СТОРОНЫ, М	ДИРЕКЦИОННЫЙ УГОЛ, ГРАДУСЫ, МИН, С	ПУНКТ НАПРАВЛЕНИЯ
	X	Y			
1. ПРОНИНО	4098.30	0.00	2512.55	342 50 58	СПИРИНО
2. СПИРИНО	6499.13	—740.92	3521.43	92 51 36	МИШИНО
3. МИШИНО	6323.42	2776.13	2367.76	176 56 45	КОСТЮКИ
4. КОСТЮКИ	3959.02	2902.28	2905.62	272 44 51	ПРОНИНО
0. ВЫСЕЛКИ	4910.43	1666.67	1854.01	244 1 16	ПРОНИНО
			2884.51	303 25 11	СПИРИНО
			1796.51	38 8 19	МИШИНО
			1559.46	127 35 46	КОСТЮКИ

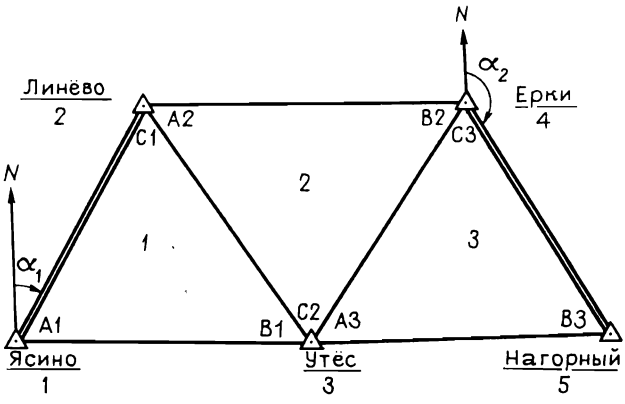


Рис. 30. Схема сети триангуляции к программе «Цепь треугольников триангуляции»

2.5.3. Программа «Цепь треугольников триангуляции»

В сети триангуляции (рис. 30) нумерация пунктов ведется от начала исходного базиса через всю систему по ходовой линии.

В каждом треугольнике углы против сторон обозначены: A (I) — против определяемой; B(I) — против исходной; C(I) — против промежуточной.

Углы C(I) служат для передачи дирекционного угла по ходовой линии, по которой идет также передача координат от пункта с условным номером 2 к пункту N+1, где N — число треугольников в сети.

Вводу подлежат следующие исходные данные:

1) средняя квадратическая погрешность измерения угла (M);
2) число треугольников системы (N);
3) фактические названия пунктов вместо условных, выдаваемых программой;
4) значения углов A(I), B(I), C(I), (где I=1, 2, ..., N);
5) дирекционный угол линии 1—2 (начальный базис);
6) дирекционный угол линии конечного базиса;
7) длина начального базиса (1—2);
8) длина конечного базиса;
9) координаты точек начала и конца двух базисов.

Тестовая задача и диалог. Исходные данные для решения тестовой задачи соответствуют схеме, приведенной на рис. 30:

1) M=10;
2) N=3;
3) 1 — Ясино, 2 — Линево, 3 — Утес, 4 — Ерки, 5 — Нагорный;
4) A1=53,21,43 B1=64,59,20 C1=61,38,48
   A2=60,33,22 B2=41,33,10 C2=77,53,31
   A3=43,29,43 B3=62,20,08 C3=74,10,07

(градусы, минуты, секунды отделены запятыми для удобства при вводе);

5) дирекционный угол линии (1—2) 28,10,15;
6) дирекционный угол конечной стороны 150,14,58;

7) длина начального базиса (1—2) 2166,48;
8) длина конечного базиса 1956,99;
9) X(1)=2682.14 Y(1)=2366.06
X(2)=4591.99 Y(2)=3388.86
X(4)=4790.77 Y(4)=6209.52
X(5)=3091.71 Y(5)=7180.63.

Диалог в процессе решения тестовой задачи приведен в табл. 35, результаты решения — в табл. 36.

Таблица 35
Диалог с программой «Цепь треугольников триангуляции»

Сообщение или запрос программы	Ответ пользователя
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН, КУДА?	1
ЦЕПЬ ТРЕУГОЛЬНИКОВ ТРИАНГУЛЯЦИИ (М НЕ МЕНЕЕ 10 СЕКУНД).	
ВВОД В СЕКУНДАХ СР.КВ.ПОГР.ИЗМЕР.УГЛА (М)?	10
ВВЕДИТЕ ЧИСЛО ТРЕУГОЛЬНИКОВ СИСТЕМЫ?	3
ПРОГРАММА ИСПОЛЬЗУЕТ УСЛОВНЫЕ НОМЕРА ПУНКТОВ.	
ВВОД ФАКТИЧЕСКИХ НАЗВАНИЙ ПУНКТОВ (КАЖДОЕ ДО 10 БУКВ).	
УСЛОВНЫЙ НОМЕР 1 — НАЧ. ИСХОДНОГО БАЗИСА, ДАЛЕЕ ПО ХОДОВОЙ ЛИНИИ.	
ПО ХОДОВОЙ ЛИНИИ ИДЕТ ПЕРЕДАЧА КООРД. ОТ П.2 К П. 4.	
УСЛОВНЫЙ НОМЕР 1, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	ЯСИНО
УСЛОВНЫЙ НОМЕР 2, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	ЛИНЕВО
УСЛОВНЫЙ НОМЕР 3, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	УТЕС
УСЛОВНЫЙ НОМЕР 4, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	ЕРКИ
УСЛОВНЫЙ НОМЕР 5, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	НАГОРНЫЙ
ВВОД ЧИСЛОВЫХ ЗНАЧЕНИЙ УГЛОВ ПО ТРЕУГОЛЬНИКАМ.	
УГЛЫ ПРОТИВ СТОРОН: 1—A1 — ПРОТИВ ОПРЕДЕЛЯЕМОЙ, 2—B1 — ПРОТИВ ИСХОДНОЙ, 3—C1 — ПРОТИВ ПРОМУЖУТОЧНОЙ	
ФОРМАТ ВВОДА ЗНАЧЕНИЙ УГЛОВ:	
ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ И НАЖАТЬ ВК.	
ДОПУСТИМАЯ УГЛОВАЯ НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 40 СЕКУНД.	
ТРЕУГОЛЬНИК 1:	
УГОЛ A1 ?	53,21,43
УГОЛ B1 ?	64,59,20
УГОЛ C1 ?	61,38,48
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 1 РАВНА —9 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 2:	
УГОЛ A2 ?	60,33,22
УГОЛ B2 ?	41,33,10
УГОЛ C2 ?	77,53,31
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 2 РАВНА 3 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 3:	
УГОЛ A3 ?	43,29,43
УГОЛ B3 ?	62,20,08
УГОЛ C3 ?	74,10,07
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 3 РАВНА —2 СЕКУНДАМ.	
СР.КВ.ПОГР.ИЗМ.УГЛА 3 СЕКУНДЫ.	

Продолжение табл. 35

Сообщение или запрос программы	Ответ пользователя
ВВЕДИТЕ ДИРЕКЦ.УГОЛ ЛИНИИ 1—2 ?	28,10,15
ВВЕДИТЕ ДИРЕКЦ.УГОЛ ЛИНИИ 4—5 ?	150,14,58
ВВЕДИТЕ В МЕТРАХ ДЛИНУ НАЧАЛЬНОГО БАЗИСА ?	2166.48
ВВЕДИТЕ В МЕТРАХ ДЛИНУ КОНЕЧНОГО БАЗИСА ?	1956.99
ВВОД КООРДИНАТ ИСХОДНЫХ ПУНКТОВ.	
В КООРДИНАТАХ ПОВТОРЯЮЩИЕСЯ ЦИФРЫ СЛЕВА ОТБРОСИТЬ.	
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	1
X1 ?	2682.14
Y1 ?	2366.06
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	2
X2 ?	4591.99
Y2 ?	3388.86
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	4
X4 ?	4790.77
Y4 ?	6209.52
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	5
X5 ?	3091.71
Y5 ?	7180.63
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	777
КОНТР. СМЕЩ. ПУНКТА 4 (В МЕТРАХ) —.12 .	
ДЛИНА ХОДОВОЙ ЛИНИИ (В МЕТРАХ) 4436 .	
ОТНОСИТЕЛЬНАЯ ПОГРЕШНОСТЬ ПЕРЕДАЧИ КООРДИНАТ 1/36000.	

Таблица 36
Результаты решения тестовой задачи по программе «Цепь треугольников триангуляции»

ЦЕПЬ ТРЕУГОЛЬНИКОВ

КАТАЛОГ КООРДИНАТ ПУНКТОВ (СКП ИЗМ.УГЛА 10 СЕКУНД)

НАЗВАНИЕ ПУНКТА	КООРДИНАТЫ, М		ДЛИНА СТОРОНЫ, М	ДИРЕКЦИОННЫЙ УГОЛ, ГРАДУСЫ, МИН, С	ПУНКТ НАПРАВЛЕНИЯ
	X	Y			
2. ЛИНЕВО	4591.99	3388.86	2166.48	208 10 15	ЯСИНО
			1918.25	146 31 21	УТЕС
			2827.66	85 58 8	ЕРКИ
3. УТЕС	2991.97	4446.98	2103.86	261 31 53	ЯСИНО
			1918.25	326 31 22	ЛИНЕВО
			2518.38	44 25 0	ЕРКИ
			2735.47	87 54 38	НАГОРНЫЙ
4. ЕРКИ	4790.77	6209.52	2827,66	265 58 8	ЛИНЕВО
			2518.38	224 25 0	УТЕС
			1957.00	150 14 58	НАГОРНЫЙ

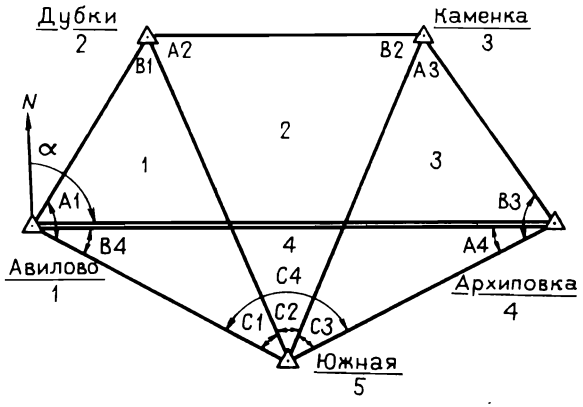


Рис. 31. Схема сети триангуляции к программе «Веер»

2.5.4. Программа «Веер»

В сети триангуляции (рис. 31) пункты 2 и 3 должны располагаться слева от направления пункт 1 (начало базиса) — пункт 4 (конец базиса). Если два указанных пункта будут расположены справа, то необходимо изменить нумерацию, присвоив условный номер 1 пункту Архиповка.

Вводу подлежат следующие исходные данные:

1) средняя квадратическая погрешность M измерения угла;

2) фактические названия пунктов вместо условных номеров; выдаваемых программой;

3) значения углов A(I), B(I), C(I), где I=1, 2, 3, 4;

4) дирекционный угол линии 1—4;

5) длина базиса 1—4;

6) координаты начального и конечного пунктов базиса.

Тестовая задача и диалог. Исходные данные для решения тестовой задачи соответствуют схеме, приведенной на рис. 31:

1) M=10;

2) 1 — Авилово, 2 — Дубки, 3 — Каменка, 4 — Архиповка, 5 — Южная;

3) A1=68,27,27 B1=58,39,28 C1=52,53,14
A2=66,30,49 B2=62,47,04 C2=50,42,04
A3=45,38,48 B3=95,08,12 C3=39,12,54
A4=20,39,42 B4=16,31,54 C4=142,48,12

(градусы, минуты, секунды отделены запятыми для удобства при вводе);

4) дирекционный угол 89,12,36;

5) длина базиса 2180.47;

6) X(1)=7323.41 Y(1)=678.30
X(4)=7353.48 Y(4)=2858.56.

Диалог в процессе решения тестовой задачи приведен в табл. 37, результаты решения — в табл. 38.

Таблица 37
Диалог с программой «Веер»

Сообщение или запрос программы	Ответ пользователя
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН. КУДА?	1
ВЕЕР—ВСТАВКА ТРЕХ ПУНКТОВ НА СТОРОНЕ (М НЕ МЕНЕЕ 10 СЕКУНД).	
ВВОД В СЕКУНДАХ СР.КВ.ПОГР.ИЗМЕРЕННОГО УГЛА ?	10
ЧИСЛО ТРЕУГОЛЬНИКОВ В СИСТЕМЕ РАВНО 4.	
ВВОД ФАКТИЧЕСКИХ НАЗВАНИЙ ПУНКТОВ (КАЖДОЕ ДО 10 БУКВ).	
УСЛОВНЫЙ НОМЕР НАЧАЛА БАЗИСА — 1, ДАЛЕЕ ПО ХОДУ ЧАС. СТРЕЛКИ.	
УСЛОВНЫЙ НОМЕР 1, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	АВИЛОВО
УСЛОВНЫЙ НОМЕР 2, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	ДУБКИ
УСЛОВНЫЙ НОМЕР 3, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	КАМЕНКА
УСЛОВНЫЙ НОМЕР 4, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	АРХИПОВКА
УСЛОВНЫЙ НОМЕР 5, ФАКТИЧЕСКОЕ НАЗВАНИЕ ?	ЮЖНАЯ
ВВОД ЧИСЛОВЫХ ЗНАЧЕНИЙ УГЛОВ ПО ТРЕУГОЛЬНИКАМ.	
УГЛЫ: 1—A1 — ПРОТИВ ИСХОДНОЙ СТОРОНЫ: 2—B1 — ПРОТИВ ОПРЕДЕЛЯЕМОЙ: 3—C1 — ВСЕГДА ПРИ ПЯТОМ ПУНКТЕ	
ФОРМАТ ВВОДА ЗНАЧЕНИЙ УГЛОВ: ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ И НАЖАТЬ ВК.	
ДОПУСТИМАЯ УГЛОВАЯ НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 40 СЕКУНД.	
ТРЕУГОЛЬНИК 1:	
УГОЛ A1 ?	68,27,27
УГОЛ B1 ?	58,39,28
УГОЛ C1 ?	52,53,14
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 1 РАВНА 9 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 2:	
УГОЛ A2 ?	66,30,49
УГОЛ B2 ?	62,47,04
УГОЛ C2 ?	50,42,04
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 2 РАВНА —3 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 3:	
УГОЛ A3 ?	45,38,48
УГОЛ B3 ?	95,08,12
УГОЛ C3 ?	39,12,54
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 3 РАВНА —6 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 4:	
УГОЛ A4 ?	20,39,42
УГОЛ B4 ?	16,31,54
УГОЛ C4 ?	142,48,12
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 4 РАВНА —12 СЕКУНДАМ.	
УСЛОВНЫЙ НОМЕР ПУНКТА НАЧАЛА БАЗИСА ВСЕГДА 1, КОНЦА — 4.	
ВВЕДИТЕ ИСХОДНЫЙ ДИРЕКЦИОННЫЙ УГОЛ ЛИНИИ 1—4 ?	89,12,36
ВВЕДИТЕ ДЛИНУ БАЗИСА В МЕТРАХ?	2180.47
ВВОД КООРДИНАТ НАЧАЛЬНОГО И КОНЕЧНОГО ПУНКТОВ БАЗИСА.	
ВВЕДИТЕ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	1
X1 ?	7323.41
Y1 ?	678.30

Продолжение табл. 37

Сообщение или запрос программы	Ответ пользователя
ВВЕДИТЕ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	4
X4 ?	7353,48
Y4 ?	2858.56
ВВЕДИТЕ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777) ?	777
БОКОВОЕ УСЛОВИЕ —15 СЕКУНД.	
КОНТР.СМЕЩ. ПУНКТА 1 (В МЕТРАХ) —.01	

Таблица 38
Результаты решения тестовой задачи по программе "Веер"

"ВЕЕР" – ВСТАВКА ТРЕХ ПУНКТОВ НА СТОРОНЕ

КАТАЛОГ КООРДИНАТ ПУНКТОВ (СКП ИЗМ. УГЛА 10 СЕКУНД)

НАЗВАНИЕ ПУНКТА	КООРДИНАТЫ, М		ДЛИНА СТОРОНЫ, М	ДИРЕКЦИОННЫЙ УГОЛ, ГРАДУСЫ, МИН, С	ПУНКТ НАПРАВЛЕНИЯ
	X	Y			
1. АВИЛОВО	7323.41	678.30	2180.47	89.12.36	АРХИПОВКА
			1188.39	37 17 7	ДУБКИ
2. ДУБКИ	8268.93	1398.21	1206.22	92 6 54	КАМЕНКА
3. КАМЕНКА	8224.41	2603.61	907.48	163 41 0	АРХИПОВКА
4. АРХИПОВКА	7353.48	2858.56	1026.33	248 32 48	ЮЖНАЯ
5. ЮЖНАЯ	6978.11	1903.34	1272.77	285 44 30	АВИЛОВО
			1386.14	338 37 43	ДУБКИ
			1429.56	29 19 50	КАМЕНКА
			1026.33	68 32 48	АРХИПОВКА

2.5.5. Программа "Вставка пунктов в угол"

На схеме сети (рис. 32) условный номер 0 присваивается пункту в вершине угла. Далее нумерация идет по ходу часовой стрелки.

Для решения задачи необходимо ввести следующие исходные данные:

1) среднюю квадратическую погрешность M измерения угла;

2) число треугольников системы N;

3) фактические названия пунктов вместо условных номеров, выдаваемых программой;

4) значения углов A(I), B(I), C(I), где I=1, 2, ... N;

5) дирекционный угол α_1 линии 0—1;

6) дирекционный угол α_2 линии 0—N+1;

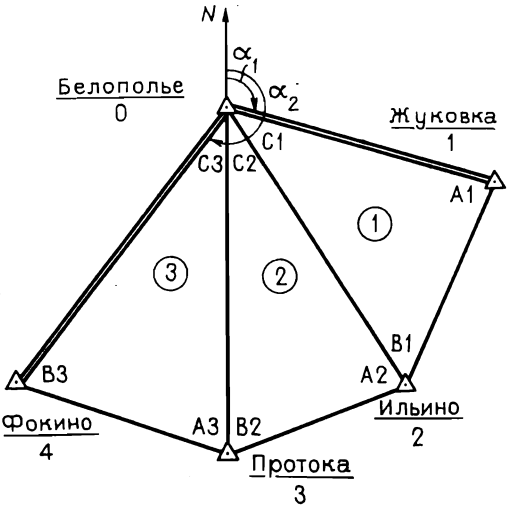
7) длину базиса 0—1 (начального);

8) длину базиса 0—N+1 (конечного);

9) координаты пунктов концов двух базисов.

Тестовая задача и диалог. Исходные данные для решения тестовой задачи соответствуют схеме, приведенной на рис. 32:

Рис. 32. Схема сети триангуляции к программе «Вставка пунктов в угол»



1) M = 10;
2) N = 3;
3) 0 — Белополье, 1 — Жуковка, 2 — Ильино, 3 — Протока, 4 — Фокино;
4) A1 = 77,01,15 B1 = 66,44,27 C1 = 36,14,07
A2 = 69,09,40 B2 = 72,38,55 C2 = 38,11,30
A3 = 52,05,18 B3 = 93,18,48 C3 = 34,36,06
(градусы, минуты, секунды отделены запятыми для удобства при вводе);
5) дирекционный угол $\alpha_1 = 126,59,26$;
6) дирекционный угол $\alpha_2 = 236,00,59$;
7) длина базиса 0—1 (начального) 2823.33;
8) длина базиса 0—4 (конечного) 2317.05;
9) X(0) = 8167.18 Y(0) = 4944.92
X(1) = 6468.43 Y(1) = 7200.01
X(4) = 6872,05 Y(4) = 3023.63

Диалог в процессе решения тестовой задачи приведен в табл. 39, результаты решения — в табл. 40.

Таблица 39
Диалог с программой "Вставка пунктов в угол"

Сообщение или запрос программы	Ответ пользователя
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН. КУДА?	1
ВСТАВКА ПУНКТОВ ТРИАНГУЛЯЦИИ В УГОЛ (М НЕ МЕНЕЕ 10 СЕКУНД). ВВОД В СЕКУНДАХ СР. КВ. ПОГР. УГЛА?	10
ВВЕДИТЕ ЧИСЛО ТРЕУГОЛЬНИКОВ СИСТЕМЫ?	3
ПРОГРАММА ИСПОЛЬЗУЕТ УСЛОВНЫЕ НОМЕРА ПУНКТОВ.	

Продолжение табл. 39

Сообщение или запрос программы	Ответ пользователя
ВВОД ФАКТИЧЕСКИХ НАЗВАНИЙ ПУНКТОВ (КАЖДОЕ ДО 10 БУКВ).	
УСЛОВНЫЙ НОМЕР ВЕРШИНЫ УГЛА — 0, ДАЛЕЕ ПО ХОДУ ЧАС. СТРЕЛКИ.	
УСЛОВНЫЙ НОМЕР 0, ФАКТИЧЕСКОЕ НАЗВАНИЕ?	БЕЛОПОЛЬЕ
УСЛОВНЫЙ НОМЕР 1, ФАКТИЧЕСКОЕ НАЗВАНИЕ?	ЖУКОВКА
УСЛОВНЫЙ НОМЕР 2, ФАКТИЧЕСКОЕ НАЗВАНИЕ?	ИЛЬИНО
УСЛОВНЫЙ НОМЕР 3, ФАКТИЧЕСКОЕ НАЗВАНИЕ?	ПРОТОКА
УСЛОВНЫЙ НОМЕР 4, ФАКТИЧЕСКОЕ НАЗВАНИЕ?	ФОКИНО
ВВОД ЧИСЛОВЫХ ЗНАЧЕНИЙ УГЛОВ ПО ТРЕУГОЛЬНИКАМ.	
УГЛЫ ПРОТИВ СТОРОН: 1—А1—ПРОТИВ ОПРЕДЕЛЯЕМОЙ, 2—В1—ПРОТИВ ИСХОДНОЙ, 3—С1—ПРОТИВ ПРОМЕЖУТОЧНОЙ.	
ФОРМАТ ВВОДА ЗНАЧЕНИЙ УГЛОВ:	
ГРАДУСЫ, МИНУТЫ, СЕКУНДЫ И НАЖАТЬ ВК.	
ДОПУСТИМАЯ УГЛОВАЯ НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 40 СЕКУНД.	
ТРЕУГОЛЬНИК 1:	
УГОЛ А1?	77,01,15
УГОЛ В1?	66,44,27
УГОЛ С1?	36,14,07
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 1 РАВНА —11 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 2:	
УГОЛ А2?	69,09,40
УГОЛ В2?	72,38,55
УГОЛ С2?	38,11,30
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 2 РАВНА 5 СЕКУНДАМ.	
ТРЕУГОЛЬНИК 3:	
УГОЛ А3?	52,05,18
УГОЛ В3?	93,18,48
УГОЛ С3?	34,36,06
НЕВЯЗКА В ТРЕУГОЛЬНИКЕ 3 РАВНА 12 СЕКУНДАМ.	
СР. КВ. ПОГР. ИЗМЕР. УГЛА 6 СЕКУНД.	
ВВЕДИТЕ ДИРЕКЦ. УГОЛ ЛИНИИ 0—1?	126,59,26
ВВЕДИТЕ ДИРЕКЦ. УГОЛ ЛИНИИ 0—4?	236,00,59
ВВЕДИТЕ В МЕТРАХ ДЛИНУ НАЧАЛЬНОГО БАЗИСА?	2823.33
ВВЕДИТЕ В МЕТРАХ ДЛИНУ КОНЕЧНОГО БАЗИСА?	2317.05
ВВОД КООРДИНАТ ИСХОДНЫХ ПУНКТОВ.	
В КООРДИНАТАХ ПОВТОРЯЮЩИЕСЯ ЦИФРЫ СЛЕВА ОТБРОСИТЬ.	
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777)?	0
X0?	8167,18
Y0?	4944,92
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777)?	1
X1?	6468.43
Y1?	7200.01
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777)?	4
X4?	6872.05
Y4?	3023.63
ВВЕДИТЕ УСЛОВНЫЙ НОМЕР ПУНКТА (ЕСЛИ ВВОД ЗАКОНЧЕН, ТО 777)?	777
КОНТР. СМЕЩ. ПУНКТА (В МЕТРАХ) —.01.	

Таблица 40
Результаты решения тестовой задачи по программе "Вставка пунктов в угол"

ВСТАВКА ПУНКТОВ В УГОЛ

КАТАЛОГ КООРДИНАТ ПУНКТОВ (СКП ИЗМ. УГЛА 10 СЕКУНД)

НАЗВАНИЕ ПУНКТА	КООРДИНАТЫ, М		ДЛИНА СТОРОНЫ, М	ДИРЕКЦИОННЫЙ УГОЛ, ГРАДУСЫ, МИН, С	ПУНКТ НАПРАВЛЕНИЯ
	X	Y			
1. ЖУКОВКА	6468.43	7200.01	1816.48	229 58 9	ИЛЬИНО
2. ИЛЬИНО	5300.07	5809.13	1939,70	274 3 57	ПРОТОКА
3. ПРОТОКА	5437,60	3874.31	1667.72	329 19 50	ФОКИНО
4. ФОКИНО	6872.05	3023.63	2317.05	56. 0 59	БЕЛОПОЛЬЕ
О. БЕЛОПОЛЬЕ	8167.18	4944.92	2823.33	126 59 26	ЖУКОВКА
			2994.53	163 13 34	ИЛЬИНО
			2932.03	201 24 59	ПРОТОКА
			2317.05	236 0 59	ФОКИНО

3. УРАВНИВАНИЕ НИВЕЛИРНЫХ СЕТЕЙ

3.1. Назначение программы и способ решения

Программа предназначена для обработки сетей 2, 3, 4 классов и технического нивелирования. Обрабатываемые сети могут включать ходы разных классов для совместного уравнивания. По программе выполняется контроль полевых измерений, уравнивание и оценка точности нивелирования. Размеры обрабатываемых сетей определяются возможностями используемой машины.

Способ решения. По высотам твердых точек и измеренным превышениям вычисляют приближенные высоты определяемых узловых точек: $H_j = H_i + h_{i-j}$, где H_j — вычисляемая приближенная высота точки j; H_i — твердая (или приближенная) высота точки на другом конце хода $i-j$, h_{i-j} — превышение по ходу $i-j$.

Средние квадратические погрешности вычисленных приближенных высот находят по формуле

$$m_j = \sqrt{m_i^2 + \mu_{i-j}^2 L_{i-j}},$$

где m_j — средняя квадратическая погрешность приближенной высоты точки j; $m_j = 0$, если точка j твердая; μ_{i-j} — средняя квадратическая погрешность нивелирования на 1 км данного класса; L_{i-j} — длина хода $i-j$ (в км).

Для контроля исходных данных вычисляют невязки f_h ходов по формуле

$$f_h = H_i - H_j + h_{i-j}$$

и допустимые невязки

$$f_{h_{\text{доп}}} = 2\sqrt{m_i^2 + m_j^2 + \mu_{i-j}^2 L_{i-j}}.$$

Если в сети обнаружены ходы с недопустимыми невязками, то выдаются соответствующие сообщения и делается запрос о целесообразности продолжения вычислений с полученными невязками. В зависимости от принятого пользователем решения программа либо прекращает, либо продолжает работу.

Уравнивание высот узловых точек выполняется параметрическим способом. Неизвестными при этом являются поправки к приближенным высотам узловых точек, а измеренными величинами — превышения по ходам. Уравнения поправок в превышении имеют простой вид:

$$v_{i-j} = \delta H_j - \delta H_i + l_{i-j}, \text{ с весом } p_{i-j},$$

где i, j — номера начальной и конечной точек хода; δH — поправка к приближенному значению высоты; H — приближенная высота; l — свободный член, вычисляемый по формуле

$$l_{i-j} = H_j - H_i - h_{i-j}.$$

Веса превышений вычисляют при работе программы по формуле $p=\lambda/L$ или $p=\lambda/\sum_{k=1}^{m} n_k$, где L — длина хода (в км); m — число секций в ходе; n_k — число штативов в k-й секции; λ — коэффициент. Способ вычисления весов превышений выбирает пользователь. Значение коэффициента λ зависит от класса хода:

Класс хода	2	3	4	Технич. нивелир.
λ	40	10	2,5	0,4

По уравнениям поправок обычным путем составляют нормальные уравнения, решение которых выполняется путем вычисления обратной матрицы и умножения ее на вектор свободных членов.

Если максимальная поправка к значениям приближенных высот не превышает допуска (в программе он принят равным 0,001 м), то уравнивание считается законченным. В противном случае оно повторяется с использованием вычисленных высот точек в качестве приближенных.

Уравненные высоты находят путем сложения приближенных с полученными поправками.

По окончании уравнивания выполняется оценка точности сети. Для каждого хода вычисляют средние квадратические погрешности уравненных высот начальной и конечной точек хода, а также среднюю квадратическую погрешность взаимного положения по высоте. Для этого используют следующие формулы:

$$m_i=c_1\sqrt{q_{i,i}};\quad m_{i,j}=c_1\sqrt{q_{i,i}+q_{j,j}-2q_{i,j}},$$

где $q_{i,j}$ — элементы матрицы обратных весов; c_1 — средняя квадратическая погрешность единицы веса.

Описанная выше оценка точности может быть выполнена (по желанию пользователя) для любой пары узловых пунктов, не принадлежащих одному и тому же ходу.

Результаты уравнивания выдаются в виде каталога высот, в котором, кроме измеренных и уравненных превышений и уравненных высот узловых точек и конечных точек секций, по каждому ходу выдается следующая информация: название хода, его класс и длина, средняя квадратическая погрешность нивелирования на 1 км (или на 1 штатив) хода данного класса, поправка в превышение для данного хода и величина допустимой невязки.

3.2. Подготовка исходных данных

Составляют схему сети, на которой показывают твердые и определяемые узловые точки и ходы между ними с указанием их направления. Точки в конце висячих ходов считают узловыми (рис. 33).

Твердые и определяемые узловые точки нумеруют на схеме в произвольном порядке. Высоты твердых точек выписывают на схему. Если в сети нет твердой точки, необходимо принять за твердую одну из узловых, задав для нее условную высоту.

Для каждого хода на схеме указывают его класс, длину и число секций, а для каждой секции через дробь выписывают измеренное превышение и число штативов в секции.

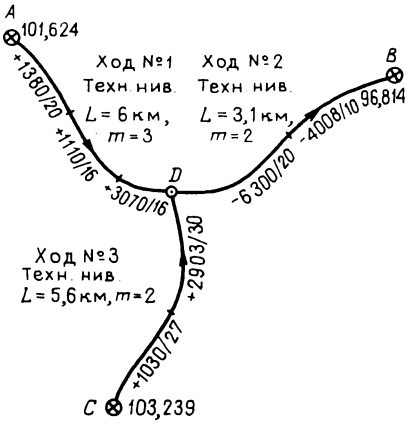


Рис. 33. Схема сети нивелирных ходов с одной узловой точкой

Для работы программы необходимо подготовить четыре массива информации.

1-й массив содержит общую информацию о задаче, а именно:

число твердых точек в сети;

число определяемых узловых точек;

число ходов;

число пар точек, не принадлежащих одному ходу, для которых требуется выполнить оценку точности их взаимного положения по высоте;

признак способа вычисления весов ходов: если признак равен 0, то веса считаются прямо пропорциональными длинам ходов; если ризнак равен 1, то веса считаются прямо пропорциональными числу штативов.

2-й массив информации составляют номера и высоты твердых точек и номера определяемых точек.

3-й массив содержит информацию о ходах. Массив состоит из t подмассивов, соответствующих числу ходов в сети. Каждый подмассив включает номер хода (ходы должны быть пронумерованы подряд числами от 1 до t) и следующие пять чисел, характеризующие ход:

номер начальной точки хода;

номер конечной точки хода;

класс хода (техническое нивелирование условно обозначается пятым классом);

число секций в ходе;

длина хода в км.

За этими данными следуют результаты измерений по ходу, а именно: измеренные превышения по секциям и число штативов в секциях.

Подмассивы могут следовать друг за другом в произвольном порядке, а не по порядку номеров ходов.

Признаком конца информации 3-го массива является 0, вводимый в ответ на очередной запрос о номере хода.

4-й массив информации готовится только тогда, когда требуется

оценка точности взаимного положения точек по высоте, не принадлежащих одному ходу. Этот массив включает в себя номера точек, взаимное положение которых следует оценить.

3.3. Исходные данные тестовой задачи

Исходные данные соответствуют схеме, изображенной на рис. 33:

1-й массив: 3 — число твердых точек; 1 — число определяемых узловых точек; 3 — число ходов; 0 — оценивать точность взаимного положения точек, не принадлежащих одному ходу, не требуется; 1 — веса прямо пропорциональны числу штативов.

2-й массив: A — номер твердой точки; 101624 — высота твердой точки (в мм): B — номер твердой точки; 96814 — высота твердой точки (в мм); C — номер твердой точки; 103239 — высота твердой точки (в мм); D — номер определяемой точки.

3-й массив: 1 — номер хода; A — номер начальной точки хода; D — номер конечной точки хода; 5 — класс хода (ход технического нивелирования); 3 — число секций в ходе; 6 — длина хода (в км); 1380 — превышение в 1-й секции; 20 — число штативов в 1-й секции; 1110 — превышение во 2-й секции; 16 — число штативов во 2-й секции; 3070 — превышение в 3-й секции; 16 — число штативов в 3-й секции.

Далее в аналогичной последовательности информация по ходу № 2:

2 D B 5 2 3.1 —6300 20 —4008 10

Информация по ходу № 3:

3 C D 5 2 5.6 1030 27 2903 30

0 — признак конца массива 3.

3.4. Диалог с программой и результаты решения тестовой задачи

Диалог в процессе решения тестовой задачи приведен в табл. 41, результаты решения — в табл. 42, 43.

Таблица 41

Диалог с программой «Уравнивание нивелирных сетей»

Сообщение или запрос программы	Ответ пользователя
УРАВНИВАНИЕ НИВЕЛИРНЫХ СЕТЕЙ.	
ВВЕДИТЕ:	
ЧИСЛО ТВЕРДЫХ ТОЧЕК	3
ЧИСЛО ОПРЕДЕЛЯЕМЫХ УЗЛОВЫХ ТОЧЕК	1
ЧИСЛО ХОДОВ	3
ЧИСЛО ПАР ОЦЕНИВАЕМЫХ ТОЧЕК, НЕ ПРИНАДЛЕЖАЩИХ ОДНОМУ ХОДУ	0
ПРОВЕРЬТЕ ВВОД:	
1) ТВЕРДЫХ ТОЧЕК — 3	
2) ОПРЕДЕЛЯЕМЫХ УЗЛОВЫХ ТОЧЕК — 1	
3) ХОДОВ — 3	

Продолжение табл. 41

Сообщение или запрос программы	Ответ пользователя
4) ПАР ОЦЕНИВАЕМЫХ ТОЧЕК, НЕ ПРИНАДЛЕЖАЩИХ ОДНОМУ ХОДУ,— 0	
БУДЕТЕ ИСПРАВЛЯТЬ ОШИБКИ? (ДА/НЕТ)	НЕТ
ВВЕДИТЕ Ø ИЛИ 1 (ПРИЗНАК СПОСОБА ВЫЧИСЛЕНИЯ ВЕСОВ ХОДОВ):	
ЕСЛИ P=C/L, ВВЕДИТЕ Ø: ЕСЛИ P=C/N,	
ВВЕДИТЕ 1.	
ЗДЕСЬ: L — ДЛИНА ХОДА В КМ, N — ЧИСЛО ШТАТИВОВ В ХОДЕ	1
ПРИ ВВОДЕ НАЗВАНИЙ И ВЫСОТ ТВЕРДЫХ ТОЧЕК И НАЗВАНИЙ ОПРЕДЕЛЯЕМЫХ ТОЧЕК СЛЕДУЕТ:	
ЛИБО ВВЕСТИ НАЗВАНИЕ, ЛИБО ВВЕСТИ СЛОВО «ПРОПУСК», ЕСЛИ НАЗВАНИЯ И ВЫСОТЫ УЖЕ БЫЛИ ВВЕДЕНЫ:	
ВВЕДИТЕ НАЗВАНИЕ ТВЕРДОЙ ТОЧКИ	A
ВВЕДИТЕ ВЫСОТУ ТВЕРДОЙ ТОЧКИ В ММ	101 624
ВВЕДИТЕ НАЗВАНИЕ ТВЕРДОЙ ТОЧКИ	B
ВВЕДИТЕ ВЫСОТУ ТВЕРДОЙ ТОЧКИ В ММ	96 814
ВВЕДИТЕ НАЗВАНИЕ ТВЕРДОЙ ТОЧКИ	C
ВВЕДИТЕ ВЫСОТУ ТВЕРДОЙ ТОЧКИ В ММ	103 239
ВВЕДИТЕ НАЗВАНИЕ ОПРЕДЕЛЯЕМОЙ ТОЧКИ	D
ПРОВЕРЬТЕ:	

НОМЕР П/П	НАЗВАНИЕ ТОЧКИ	ВЫСОТА, ММ
1.	A	101 624
2.	B	96 814
3.	C	103 239
4.	D	

БУДЕТЕ ИСПРАВЛЯТЬ ОШИБКИ? (ДА/НЕТ)	НЕТ
ВВОДИТЕ ИНФОРМАЦИЮ О ХОДАХ: ТЕХНИЧЕСКОЕ НИВЕЛИРОВАНИЕ	
НАЗЫВАЙТЕ 5 КЛАССОМ	
* ВВЕДИТЕ НОМЕР ХОДА (ИЛИ Ø, КОГДА ВСЯ ИНФОРМАЦИЯ О ХОДАХ УЖЕ ВВЕДЕНА)	1
НОМЕР НАЧАЛЬНОЙ ТОЧКИ	A
НОМЕР КОНЕЧНОЙ ТОЧКИ	D
КЛАСС ХОДА	5
ЧИСЛО СЕКЦИЙ В ХОДЕ	3
ДЛИНА ХОДА, В КМ	6
ВВЕДИТЕ ДАННЫЕ ПО СЕКЦИЯМ ХОДА.	
ВВЕДИТЕ ПРЕВЫШЕНИЕ (В ММ) И ЧИСЛО ШТАТИВОВ В СЕКЦИИ	1380
?	20
ВВЕДИТЕ ПРЕВЫШЕНИЕ (В ММ) И ЧИСЛО ШТАТИВОВ В СЕКЦИИ	1110
?	16
ВВЕДИТЕ ПРЕВЫШЕНИЕ (В ММ) И ЧИСЛО ШТАТИВОВ В СЕКЦИИ	3070
?	16
ИНФОРМАЦИЯ ПО ХОДУ 1 ВВЕДЕНА.	
ОШИБКИ ИСПРАВИТЕ ПО ОКОНЧАНИИ ВВОДА ВСЕЙ ИНФОРМАЦИИ.	
ПРОДОЛЖАЙТЕ ВВОД.	

Продолжение табл. 41

Сообщение или запрос программы	Ответ пользователя
Далее диалог повторяется еще два раза, начиная с запроса, отмеченного звездочкой, давая возможность ввести информацию о 2-м и 3-м ходах. Затем следует такое продолжение:	
ВВЕДИТЕ НОМЕР ХОДА (ИЛИ Ø, КОГДА ВСЯ ИНФОРМАЦИЯ О ХОДАХ УЖЕ ВВЕДЕНА)	Ø
ПРОВЕРЬТЕ ИНФОРМАЦИЮ О ХОДАХ.	
На экран (или печать) выдается информация 3-го массива исходных данных, после чего следует:	
БУДЕТЕ ВНОСИТЬ ИСПРАВЛЕНИЯ В ИНФОРМАЦИЮ О ХОДАХ? (ДА/НЕТ)	НЕТ
БУДЕТЕ ИСПРАВЛЯТЬ ОШИБКИ В ОБЩЕЙ ИНФОРМАЦИИ О СЕТИ? (ДА/НЕТ)	НЕТ

Таблица 42
Результаты решения тестовой задачи по программе «Уравнивание нивелирных сетей» (оценка точности)

ОЦЕНКА ТОЧНОСТИ			
ХОД I — J	MI, ММ	MJ, ММ	MIJ, ММ
A — D	0.00	20.11	20.11
D — B	20.11	0.00	20.11
C — D	0.00	20.11	20.11

Таблица 43
Результаты решения тестовой задачи по программе «Уравнивание нивелирных сетей» (каталог высот)

КАТАЛОГ ВЫСОТ				
НАЗВАНИЕ ТОЧКИ	ЧИСЛО СТАНЦИЙ	ПРЕВЫШЕНИЯ, М		ВЫСОТА, М
		ИЗМЕРЕННОЕ	УРАВНЕННОЕ	
ХОД A — D, 5 КЛАСС; L=6.000 КМ C=5,3 ММ; V=−32 ММ; $F_{доп}$=122 ММ				
A				101.624
	20	1.380	1.368	
1				102.992
	16	1.110	1.100	
2				104.092
	16	3.070	3.060	
D				107.152
ХОД D — B, 5 КЛАСС; L=3.100 КМ C=5.3; V=−30 ММ; $F_{доп}$=88 ММ				
D				107.152
	20	−6.300	−6.320	
1				100.832

Продолжение табл. 43

КАТАЛОГ ВЫСОТ				
НАЗВАНИЕ ТОЧКИ	ЧИСЛО СТАНЦИЙ	ПРЕВЫШЕНИЯ, М		ВЫСОТА, М
		ИЗМЕРЕННОЕ	УРАВНЕННОЕ	
B	10	—4.008	—4.018	96,814
ХОД C — D; 5 КЛАСС; L=5.600 КМ C=5.3 ММ; V=—20 ММ; $F_{доп}$=118 ММ				
C				103.239
	27	1.030	1.020	
1				104.259
	30	2.903	2.893	
D				107.152

4. АНАЛИТИЧЕСКАЯ ПОДГОТОВКА ДАННЫХ ДЛЯ ГЕОДЕЗИЧЕСКИХ РАЗБИВОЧНЫХ РАБОТ

Для переноса проекта сооружения в натуру (разбивки) требуется предварительно выполнить подготовку геодезических данных. Она состоит в нахождении угловых и линейных величин, используемых для определения положения на местности отдельных точек сооружения. В некоторых случаях возникает задача перенесения в натуру и точек запроектированной геодезической основы. Для решения таких задач можно использовать программы, приведенные в данной главе.

4.1. Вычисление углов и линий проектного теодолитного хода

Программа предназначена для расчета значений угловых и линейных элементов проектного теодолитного хода по заданным проектным координатам его поворотных точек.

Способ решения. По проектным координатам решают обратные геодезические задачи между смежными поворотными точками проектного теодолитного хода. Полученные значения угловых и линейных элементов используют при выносе и закреплении проектного теодолитного хода на местности. Программа позволяет выполнить обработку проектных теодолитных ходов с числом вершин до 50.

В программе предусмотрен ввод исходных данных как в режиме диалога, так и по имени подготовленного на диске файла с исходными данными.

Найденные по программе значения углов выдаются с округлением до целых секунд, а расстояния — с округлением до см и числом верных (значащих) цифр не более шести. При подготовке исходных данных следует иметь в виду, что в данные должны быть включены и координаты четырех существующих (двух в начале и двух в конце хода) на местности пунктов, необходимых для привязки проектного теодолитного хода к пунктам плановой геодезической сети (рис. 34).

Если предполагается ввод исходных данных с магнитного диска, то их необходимо предварительно подготовить с помощью редактора текстов. Поскольку в программе при вводе с магнитного диска используются операторы: INPUT N FOR I=1,TO N INPUT X(I), Y (I) NEXT I, то исходные данные при их подготовке в редакторе текстов следует располагать на экране следующим образом:

```
N
X(1), Y(2)
X(2), Y(2)
.....
X(N), Y(N)
```

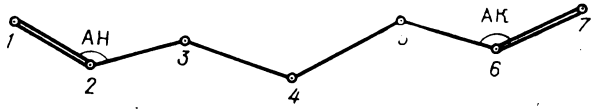


Рис. 34. Схема проектного теодолитного хода:
1, 2, 6, 7 — существующие на местности пункты: *3, 4, 5,* — проектируемые пункты: *АН, АК* — примычные углы

Тестовая задача и диалог. Для решения тестовой задачи необходимо ввести следующие значения исходных данных:

число вершин 5;

координаты вершин:

X(1) = 8487·54 Y(1) = 1283.32
X(2) = 7986.02 Y(2) = 1784.94
X(3) = 8316.53 Y(3) = 2015.40
X(4) = 8511.34 Y(4) = 1285.30
X(5) = 8133.18 Y(5) = 801.36

Диалог в процессе решения тестовой задачи в режиме ввода исходных данных с пульта приведен в табл. 44, результаты решения — в табл. 45.

Таблица 44
Диалог с программой вычисления углов и линий проектного теодолитного хода

Сообщение или запрос программы	Ответ пользователя
ВВОД 1 — ПЕЧАТЬ НА БУМАГУ, 0 — ВЫДАЧА НА ЭКРАН, КУДА?	1
УКАЖИТЕ РЕЖИМ ВВОДА ИСХОДНЫХ ДАННЫХ.	
ВВОД 1 — ДИАЛОГОВЫЙ, 0 — ЧТЕНИЕ С ДИСКА. КАКОЙ?	1
ВВОД ИСХОДНЫХ ДАННЫХ В ДИАЛОГЕ.	
ВВОД ЧИСЛА ВЕРШИН ПРОЕКТНОГО ТЕОДОЛИТНОГО ХОДА?	5
ВЕРШИНЫ ХОДА ДОЛЖНЫ ИМЕТЬ ПОРЯДКОВЫЕ НОМЕРА ОТ 1 ДО 5.	
В КООРДИНАТАХ ПОВТОРЯЮЩИЕСЯ ЦИФРЫ СЛЕВА ОТБРОСИТЬ.	
ВВОД КООРДИНАТ (ЗАВЕРШАЕТСЯ ВВОДОМ ЧИСЛА 777).	
ПОРЯДОК ВВОДА (ИЛИ ИСПРАВЛЕНИЙ): НОМЕР ТОЧКИ I; X(I); Y(I)?	
ВВОД НОМЕРА ОЧЕРЕДНОЙ ТОЧКИ ХОДА I?	1
X(1)?	8487.54
Y(1)?	1283.32
ВВОД НОМЕРА ОЧЕРЕДНОЙ ТОЧКИ ХОДА I?	2
X(2)?	7986.02
Y(2)?	1784.94
ВВОД НОМЕРА ОЧЕРЕДНОЙ ТОЧКИ ХОДА I?	3
X(3)?	8316.53
Y(3)?	2015.40
ВВОД НОМЕРА ОЧЕРЕДНОЙ ТОЧКИ ХОДА I?	4
X(4)?	8511.34
Y(4)?	1285.30

Продолжение табл. 44

Сообщение или запрос программы	Ответ пользователя
ВВОД НОМЕРА ОЧЕРЕДНОЙ ТОЧКИ ХОДА 1? X(5)? Y(5)? ВВОД НОМЕРА ОЧЕРЕДНОЙ ТОЧКИ ХОДА 1?	5 8133.18 801.36 777

Таблица 45
Результаты решения тестовой задачи по программе «Проектный теодолитный ход»

УГЛЫ И ЛИНИИ ПРОЕКТНОГО ТЕОДОЛИТНОГО ХОДА				
НОМЕР ТОЧКИ	КООРДИНАТЫ ПРОЕКТНЫЕ), М		УГОЛ (ЛЕВЫЙ), ГРАДУСЫ, МИН, С	ГОРИЗОНТАЛЬНОЕ ПОЛОЖЕНИЕ, М
	X	Y		
1.	8487.54	1283.32		709.33
2.	7986.02	1784.94	79 53 36	402.92
3.	8316.53	2015.40	70 3 8	755.64
4.	8511.34	1285.30	127 3 19	614.17
5.	8133.18	801.36		

4.2. Расчет разбивочных элементов

Назначение программы. По программе вычисляют разбивочные углы и расстояния, используемые при выносе в натуру точек способами полярных координат, прямоугольных координат, линейной засечки и прямой угловой засечки. Программа выполняет оценку точности для каждого конкретного варианта разбивки, предоставляя тем самым возможность выбора оптимального варианта.

Способ решения. По заданным координатам точек геодезической основы и проектным координатам выносимых в натуру точек решают обратные геодезические задачи. Результаты их решения позволяют определить все необходимые разбивочные элементы. Расстояния получают непосредственно из решения обратных задач, а разбивочные углы вычисляют как разности соответствующих дирекционных углов.

Оценка точности способов разбивки выполняется по формулам:

$$M_{\text{пол. коор}} = \sqrt{(m_\beta S/\rho)^2 + m_S^2};$$

$$M_{\text{прям. коор}} = \sqrt{(m_\beta S/\rho)^2 + m_{S_1}^2 + m_{S_2}^2};$$

$$M_{\text{лин. зас}} = \sqrt{m_{S_1}^2 + m_{S_2}^2}/\sin\gamma;$$

$$M_{\text{прям. зас}} = m_\beta \sqrt{S_1^2 + S_2^2}/(\rho \sin\gamma).$$

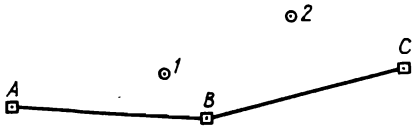


Рис. 35. Схема геодезической сети к расчету разбивочных элементов

Подготовка исходных данных. Составляется схема сети, на которой показывают точки геодезической основы и выносимые в натуру проектные точки. Нумерация точек произвольная, но не должно быть точки с номером, равным нулю (рис. 35).

1-й массив исходных данных составляет общая информация о задаче — это число точек геодезической основы и число проектных точек.

2-й массив информации включает в себя номера и прямоугольные координаты точек основы и проектных точек, причем, сначала должны быть перечислены все точки основы, затем — проектные.

3-й массив содержит информацию о предполагаемой точности угловых и линейных измерений. Данные этого массива используются для оценки точности вариантов разбивки. Массив включает четыре величины: m_β, μ_S, m_{S_1} и m_{S_2}. Пояснения о порядке задания этих величин см. на с. 105, 106.

4-й массив дает ответы на вопросы, какие способы разбивки предполагается использовать. Если тот или иной способ планируется, то в ответ на запрос машины следует вводить 1 (единицу), если нет,— то 0 (ноль).

5-й массив составляют исходные данные для разбивки способом прямоугольных координат. Если разбивка данным способом не предполагается, то вводится нуль, в противном случае вводится единица и далее готовится следующая информация:

а) название стороны геодезической основы (номера точек на концах стороны), с которой возможна разбивка способом прямоугольных координат;

в) число проектных точек, разбиваемых от этой стороны;

с) номера проектных точек, число которых указано выше.

Далее идет аналогичная информация для следующей стороны основы и т. д.

0 (нуль) — признак конца информации 5-го массива.

6-й массив содержит две величины: предельное расстояние от точки основы до проектной точки для линейных измерений; предельное расстояние для угловых измерений.

7-й массив. Порядок дальнейшей подготовки информации зависит от принятого способа ограничения числа просчитываемых вариантов разбивки. Просчет всех возможных вариантов вряд ли целесообразен из-за большого их числа, однако программа предоставляет возможность просчета и всех вариантов. В таком случае массив 7 отсутствует.

Для уменьшения числа возможных вариантов разбивки предлагается выбрать один из двух способов:

1. Наметить, так называемые, базисы разбивки — стороны геоде-

зической основы, с концов которых будут выполняться угловые и линейные измерения, причем, при угловых измерениях всегда одной стороной угла будет базис, второй — направление на проектную точку; с точек, не являющихся концами базисов, разбивки проектироваться не будут.

2. Указать, с каких именно точек основы возможен вынос в натуру каждой проектной точки.

При выборе 1-го способа 7-й массив информации содержит следующие данные: число базисов разбивки и названия базисов (указывают номерами двух точек на концах базиса).

При выборе 2-го способа 7-й массив информации формируется в таком порядке:

а) номер проектной точки;

в) число *l* точек основы, с которых возможна разбивка данной точки;

с) номера точек основы.

Далее идет аналогичная информация (а, в, с) для следующей проектной точки и т. д.

0 (нуль) — признак конца информации 7-го массива при этом способе.

Для точек, разбивка которых предполагается со всех точек основы или не предполагается ни с каких, вводится информация а и в.

Тестовая задача и диалог. Исходные данные для решения тестовой задачи соответствуют схеме, приведенной на рис. 35:

1-й массив: 3 — число точек геодезической основы, 2 — число проектных точек.

2-й массив — номера и координаты точек:

A 1033.56 2035.08 B 1015 2145.73 C 1054.38 2265.96

1 1040 2120 2 1100 2180.

3-й массив: 10 0 0 5000.

4-й массив — способы разбивки, предполагаемые для использования: полярный — 1; линейная засечка — 1; прямая засечка — 1.

5-й массив: 1 — способ перпендикуляров планируется; A B — название стороны для способа перпендикуляров (каждый номер вводится отдельно); 1 — число проектных точек, разбиваемых от этой стороны; 1 — номер проектной точки; 0 — признак конца массива.

6-й массив: 150 — предел для линий; 300 — предел для углов.

7-й массив: предполагается наметить базисы разбивки; 2 — число базисов; A B B C — названия двух базисов (A—B и B—C).

Диалог в процессе решения тестовой задачи приведен в табл. 46, результаты решения для точки *1*, выданные машиной, в табл. 47—49.

Таблица 46
Диалог с программой «Подготовка данных для разбивки»

Сообщение или запрос программы	Ответ пользователя
ПОДГОТОВКА ДАННЫХ ДЛЯ РАЗБИВКИ.	
ВВЕДИТЕ ЧИСЛО ТОЧЕК ОСНОВЫ	3
ВВЕДИТЕ ЧИСЛО ПРОЕКТНЫХ ТОЧЕК	2
ХОТИТЕ ИСПРАВИТЬ? (ДА/НЕТ)	НЕТ

Продолжение табл. 46

Сообщение или запрос программы	Ответ пользователя
ВВОД НОМЕРОВ И КООРДИНАТ ТОЧЕК:	
ВВЕДИТЕ НОМЕР ТОЧКИ ОСНОВЫ	A
ВВЕДИТЕ X И Y ТОЧКИ	1033.56
?	2035.08
ВВЕДИТЕ НОМЕР ТОЧКИ ОСНОВЫ	B
ВВЕДИТЕ X И Y ТОЧКИ	1015
?	2145.73
ВВЕДИТЕ НОМЕР ТОЧКИ ОСНОВЫ	C
ВВЕДИТЕ X И Y ТОЧКИ	1054.38
?	2265.96
ВВЕДИТЕ НОМЕР ПРОЕКТНОЙ ТОЧКИ	1
ВВЕДИТЕ X И Y ТОЧКИ	1040
?	2120
ВВЕДИТЕ НОМЕР ПРОЕКТНОЙ ТОЧКИ	2
ВВЕДИТЕ X И Y ТОЧКИ	1100
?	2180
ПРОВЕРЬТЕ:	
На экран (или печать) выдается массив номеров и координат точек.	
ХОТИТЕ ИСПРАВИТЬ? (ДА/НЕТ)	НЕТ
ВВЕДИТЕ ХАРАКТЕРИСТИКИ ТОЧНОСТИ — 4 ВЕЛИЧИНЫ	10
?	0
?	0
?	5000
ХОТИТЕ ПОВТОРИТЬ ВВОД? (ДА/НЕТ)	НЕТ
КАКИЕ СПОСОБЫ РАЗБИВКИ ПРЕДПОЛАГАЕТСЯ ИСПОЛЬЗОВАТЬ?	
ВВОДИТЕ 1, ЕСЛИ СПОСОБ ПРЕДПОЛАГАЕТСЯ ИСПОЛЬЗОВАТЬ, И Ø, ЕСЛИ НЕТ	
ПОЛЯРНЫЙ? (1/Ø)	1
ЛИНЕЙНАЯ ЗАСЕЧКА? (1/0)	1
ПРЯМАЯ ЗАСЕЧКА? (1/Ø)	1
ПРЕДПОЛАГАЕТСЯ ЛИ РАЗБИВКА СПОСОБОМ ПЕРПЕНДИКУЛЯРОВ?	
(ВВЕДИТЕ 1, ЕСЛИ ДА, И Ø, ЕСЛИ НЕТ)	1
ВВЕДИТЕ НАЗВАНИЕ СТОРОНЫ ДЛЯ СПОСОБА ПЕРПЕНДИКУЛЯРОВ (ИЛИ Ø, ЕСЛИ ВСЕ НАЗВАНИЯ СТОРОН УЖЕ ВВЕДЕНЫ)	A
?	B
ВВЕДИТЕ ЧИСЛО ПРОЕКТНЫХ ТОЧЕК, РАЗБИВАЕМЫХ ОТ ЭТОЙ СТОРОНЫ	1
ВВЕДИТЕ НОМЕР ПРОЕКТНОЙ ТОЧКИ	1
ВВЕДИТЕ НАЗВАНИЕ СТОРОНЫ ДЛЯ СПОСОБА ПЕРПЕНДИКУЛЯРОВ (ИЛИ Ø, ЕСЛИ ВСЕ НАЗВАНИЯ СТОРОН УЖЕ ВВЕДЕНЫ)	Ø
ВВЕДИТЕ ПРЕДЕЛЬНЫЕ РАССТОЯНИЯ ДЛЯ ЛИНЕЙНЫХ, ЗАТЕМ ДЛЯ УГЛОВЫХ ИЗМЕРЕНИЙ	150
?	300
НУЖНО ЛИ РАССЧИТЫВАТЬ ВСЕ ВОЗМОЖНЫЕ ВАРИАНТЫ РАЗБИВКИ (ДА/НЕТ)	НЕТ
БУДЕТЕ ЗАДАВАТЬ БАЗИСЫ РАЗБИВКИ (ДА/НЕТ)	ДА
ВВЕДИТЕ ЧИСЛО БАЗИСОВ	2
ВВЕДИТЕ НАЗВАНИЕ БАЗИСА	A
?	B
ВВЕДИТЕ НАЗВАНИЕ БАЗИСА	B
?	C

Продолжение табл. 46

Сообщение или запрос программы	Ответ пользователя
ПРОВЕРЬТЕ ИСХОДНЫЕ ДАННЫЕ На экран (или печать) выдается вся исходная информация (7 массивов). ХОТИТЕ ИСПРАВИТЬ? (ДА/НЕТ)	НЕТ

Таблица 47
Результаты решения тестовой задачи по программе «Подготовка данных для разбивки» (способ перпендикуляров)

СПОСОБ ПЕРПЕНДИКУЛЯРОВ				
СТОРОНА ОСНОВЫ (I—J)	НОМЕР ПРОЕКТНОЙ ТОЧКИ (K)	РАССТОЯНИЕ L(I—K), М	ДЛИНА ПЕРПЕНДИКУЛЯРА, М	ТОЧНОСТЬ РАЗБИВКИ M(K), СМ
B—A	1 (II)	29.51	20.40	0.7

Таблица 48
Результаты решения тестовой задачи по программе «Подготовка данных для разбивки» (данные для разбивки точки 1)

ДАННЫЕ ДЛЯ РАЗБИВКИ ТОЧКИ 1			
ТОЧКА СТОЯНИЯ	ТОЧКА НАБЛЮДЕНИЯ	НАПРАВЛЕНИЕ, ГРАДУСЫ, МИН, С	РАССТОЯНИЕ, М
A	I	0 0 0,0	85.164
	B	13 51 31,4	
B	I	0 0 0.0	35.875
	A	325 20 46.9	
	C	117 41 19.6	
C	I	0 0 0.0	146.667
	B	347 29 27.6	

Таблица 49
Результаты решения тестовой задачи по программе «Подготовка данных для разбивки» (оценка точности способов разбивки)

ОЦЕНКА ТОЧНОСТИ СПОСОБОВ РАЗБИВКИ		
СПОСОБ РАЗБИВКИ	ВАРИАНТЫ (ТОЧКИ СТОЯНИЯ)	ТОЧНОСТЬ M(K), СМ
ПОЛЯРНЫЙ	A	1.8
	B	0.7
	C	3.0
ЛИНЕЙНАЯ ЗАСЕЧКА	B, A	2.5
	C, B	4.0
ПРЯМАЯ ЗАСЕЧКА	B, A	0.6
	C, B	1.0

ОГЛАВЛЕНИЕ

СПРАВОЧНОЕ ИЗДАНИЕ

Коробочкин Михаил Ильич
Бережнов Вениамин Сергеевич
Зайцева Наталья Семеновна
Красницкий Виктор Сергеевич

РЕШЕНИЕ МАССОВЫХ ГЕОДЕЗИЧЕСКИХ ЗАДАЧ НА МИКРОЭВМ

Заведующий редакцией *О. И. Паркани*
Редактор издательства *Т. А. Борисова*
Технические редакторы *Л. А. Мурашова, М. П. Виноградова*
Корректор *Е. В. Королева*

ИБ № 7274

Сдано в набор 13.09.90. Подписано в печать 01.03.91. Формат 60×90 1/16. Бумага офсетная № 2. Гарнитура Литературная. Печать офсетная. Усл. печ. л. 9,0. Усл. кр.-отт. 9,25. Уч.-изд. л. 9,79. Тираж 9200 экз. Заказ 127 /2188-8. Цена 60 коп.

Ордена «Знак Почета» издательство «Недра»
125047 Москва, Тверская застава, 3

Набрано в ордена Октябрьской Революции и ордена Трудового Красного Знамени МПО «Первая Образцовая типография» Государственного комитета СССР по печати. 113054, Москва, Валовая, 28

Отпечатано в московской типографии № 6 Союзполиграфпрома при Государственном комитете СССР по печати. 109088, Москва Ж-88, Южнопортовая, 24

В издательстве «Недра» в 1991 году выйдет монография

ПРОГРАММИРОВАНИЕ ГЕОДЕЗИЧЕСКИХ ЗАДАЧ НА ЯЗЫКЕ БЕЙСИК

Автор — кандидат технических наук **Г. Г. Побединский**

В монографии систематизирован и обобщен опыт решения различных геодезических задач на персональных ЭВМ. Описаны процессы создания программ составления технического задания, алгоритмизации, составления текстов; приведены программы. Изложены общие положения по созданию программ для диалогового режима работы, даны примеры их конкретной реализации. Рассмотрены основные функции языка Бейсик, приведены примеры составления алгоритмов и написания программ для решения типовых геодезических задач.

Монография заинтересует инженерно-технических работников, занимающихся обработкой геодезической информации с использованием персональных ЭВМ.

В издательстве «Недра» в 1993 году выйдет книга

СПУТНИКОВАЯ АЛЬТИМЕТРИЯ

Авторы: **В. В. Бойков,** профессор, доктор технических наук; **П. П. Медведев,** кандидат технических наук

В книге рассмотрены проблемы определения параметров гравитационного поля Земли, морской топографической поверхности, единой мировой системы высот, океанических приливов и высот волн в Мировом океане с помощью метода спутниковой альтиметрии.

Изложены конструктивные особенности высотомеров. Приведены рекомендации по учету различных возмущающих факторов, влияющих на точность решения задач.

Книга заинтересует научных работников и специалистов в области спутниковой навигации и геодезии, океанографии, геофизики. Будет полезна для широкого круга геодезистов и океанографов, а также студентам геодезических вузов.

В издательстве «Недра» в 1992 году выйдет монография

СПОСОБ ОПРЕДЕЛЕНИЯ ФИЗИЧЕСКИХ РЕДУКЦИЙ В ГЕОДЕЗИЧЕСКИХ ИЗМЕРЕНИЯХ

Автор — кандидат технических наук **О. А. Мозжухин**, около 30 лет занимается проблемой учета влияния атмосферы на результаты геодезических измерений. Им опубликовано две монографии и 70 научных статей в периодических изданиях, получено три авторских свидетельства на способ определения поправок за счет влияния рефракции и показателя преломления.

В монографии изложены методы учета вертикальной рефракции и показателя преломления при геодезических измерениях в приземном слое атмосферы, основанные на применении теории подобия и моделирования. Рассмотрены способы определения поправок в результаты одно- и двустороннего тригонометрического нивелирования, высокоточного геометрического нивелирования, при определении расстояний свето- и радиодальномерами, а также оптическими дальномерами с вертикальной базой.

Книга предназначена для специалистов, выполняющих высокоточные геодезические измерения. Будет полезна для студентов и аспирантов геодезических вузов.

Если Вас заинтересовали книги издательства «Недра», то рекомендуем Вам оформить предварительный заказ в магазинах, распространяющих научно-техническую литературу.

Своевременно оформленный заказ гарантирует приобретение нужной Вам книги.

Адреса опорных магазинов издательства «Недра»:
117334 Москва, Ленинский проспект, 40, магазин № 115 «Дом научно-технической книги».
199178 Ленинград, В. О., Средний проспект, 61, магазин № 17 «Недра»

60 коп.

РЕШЕНИЕ МАССОВЫХ ГЕОДЕЗИЧЕСКИХ ЗАДАЧ НА МИКРОЭВМ

НЕДРА