

МАТЕМАТИКА КИБЕРНЕТИКА

Новое
в жизни,
науке,
технике

Подписная
научно-
популярная
серия

Издается
ежемесячно
с 1967 г.

А.Б. Бойко
Игры
с микро-
калькулятором



1987/11

Новое
в жизни,
науке,
технике

МАТЕМАТИКА КИБЕРНЕТИКА

Подписная
научно-
популярная
серия

11/1987

Издается
ежемесячно
с 1967 г.

А. Б. Бойко

ИГРЫ С МИКРОКАЛЬКУЛЯТОРОМ

СОДЕРЖАНИЕ

От автора	3
Введение	3
В поисках утраченного порядка	5
«Магический квадрат»*	5
«Одномерный кубик Рубика»	8
«Кольцо с секретом»*	11
Поединки со случаем	14
«Опасный рейс»*	15
«Электронные городки»	18
«Жонглер и шарик»	20
Большие гонки	24
«Поездка по городу»	24
«Гонки на зимнем озере»*	28
«Регата»*	30
Совершенный стратег	33
«Три шашки»*	34
«Африканское сафари»*	35
«Охота на лис»	38
Быстрота и точность*	40
«Кто первый»*	41
«Восемь цифр в уме»*	42
«Учебный полет»*	44
Заключение	48
Список литературы	48



Издательство
«Знание»
Москва
1987

ББК 32.974
Б72

БОЙКО Алексей Борисович — заведующий отделом журнала «Наука и жизнь», автор ряда публикаций по вычислительной технике в журналах «Наука и жизнь», «Химия и жизнь», «Радио», один из авторов сборника «Микрокалькуляторы в играх и задачах».

Рецензент: Пухначев Ю. В., — кандидат физико-математических наук.

Бойко А. Б.

Б72 Игры с микрокалькулятором.— М.: Знание, 1987.— 48 с.— (Новое в жизни, науке, технике. Сер. «Математика, кибернетика»; № 11).

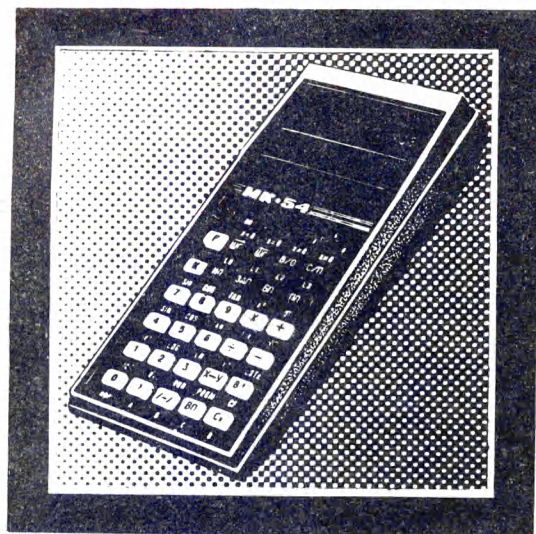
11 к.

Брошюра, содержащая пятнадцать игровых программ для одного из самых массовых отечественных программируемых микрокалькуляторов «Электроника БЗ-34», поможет не только увлекательно провести время. Внимательный читатель найдет на ее страницах немало интересных приемов программирования. Алгоритмы и комментарии к программам помогут разобраться в их структуре и самостоятельно составить программу игры для микро-ЭВМ «Электроника МК-85», «Микроша», «Электроника БК-0010» и др.

Предназначена широкому кругу читателей — всем, кто хочет углубить свои знания в области микровычислительной техники.

2302020200

ББК 32.974



ОТ АВТОРА

Название этой книги располагает к увлекательному времяпровождению. Безусловно, все описанные здесь игры можно использовать для того, чтобы отдохнуть в часы досуга.

Однако назначение книги и всех собранных в ней игровых программ не только в этом. Гораздо важнее то, что, развлекаясь с программируемым микрокалькулятором, человек, имевший прежде о вычислительной технике самое отдаленное представление, постепенно привыкает к ее использованию, как мы привыкли разговаривать по телефону или смотреть телевизор.

Книга адресована не только начинающим. Читатель, осваивающий основы программирования, многое почерпнет, разбирая тексты программ. Обладатель микро-ЭВМ, таких, как «Электроника БК-0010», «Микроша» или «Электроника МК-85», «Электроника МК-72», сможет на основе алгоритмов, приведенных в книге, создать программы для своего компьютера.

Вдобавок ко всему этому собранные в книге игры помогут читателю развить такие жизненно необходимые качества, как память, логическое мышление, быстрота реакции, способность приспосабливаться к быстро изменяющейся ситуации.

Бурный процесс микроминиатюризации в электронике привел к созданию больших интегральных схем, а затем и появлению на их основе миниатюрных ЭВМ — микрокалькуляторов. В 1978 году в продаже появился первый отечественный программируемый микрокалькулятор «Электроника БЗ-21». На его основе вскоре были разработаны две настольные модели «Электроника» МК-46 и МК-64, которые можно использовать не только для расчетов, но и для автоматизации электрических измерений и контроля технологических процессов. А еще через некоторое время появились первые представители нового семейства программируемых микрокалькуляторов. Базовой моделью нового семейства стала «Электроника БЗ-34». Ее карманный вариант получил название «Электроника МК-54», настольный — «Электроника МК-56». Семейство этих калькуляторов получило большое распространение, поэтому именно БЗ-34 выбран в качестве той базовой модели, для которой составлены игровые программы.

Основными характеристиками программируемого микрокалькулятора считают объем программной памяти, число адресуемых регистров, наличие перепрограммируемого устройства памяти и его емкость. Объем программной памяти калькулятора измеряется в шагах программы. Число адресуемых регистров показывает, сколько чисел, необходимых для вычислений по программе, способен хранить микрокалькулятор. Перепрограммируемое запоминающее устройство (ППЗУ) позволяет машинке в течение длительного времени помнить введенную программу, т. е. существенно упрощает работу с калькулятором, оборудованным подобным устройством. Не менее важной характеристикой является число возможных различных команд. Основные характеристики выпускаемых микрокалькуляторов собраны в табл. 1.

В 1985 году появились еще две модели программируемых микрокалькуляторов «Электроника»: МК-61 и МК-52, которые можно считать дальнейшим развитием семейства БЗ-34.

На смену этому семейству идут модели, подобные микро-ЭВМ «Электрони-

Таблица 1

Модель ПМК	Объем памяти (шаги)	Число регистров	Емкость ППЗУ	Число различных программируемых команд	Питание (кроме сетевого)
БЗ-21	60	6	—	33	Аккумуляторы
БЗ-34	98	14	—	58	Аккумуляторы
МК-54	98	14	—	58	Элементы
МК-56	98	14	—	58	—
МК-61	105	15	—	65	Элементы
МК-52	105	15	512	65	Элементы

ка МК-85». Эта микро-ЭВМ программируется на алгоритмическом языке БЕЙСИК и, по-видимому, в скором времени заменит традиционные программируемые микрокалькуляторы.

Игры, собранные в книге, предназначены для калькуляторов различных марок. Чтобы читателю легче было ориентироваться в программах, использованы следующие обозначения. Если справа от названия игры не стоит никаких дополнительных знаков, то программу можно использовать на калькуляторах «Электроника БЗ-34», «Электроника МК-54», «Электроника МК-56».

Обозначения на клавишах этих машинок немного различаются, поэтому для удобства читателя в книге введена единая система обозначений (табл. 2).

Таблица 2

Обозначения на клавиатуре БЗ-34	Обозначения на клавиатурах МК-54, МК-56, МК-61, МК-52	Обозначения, принятые в книге
ИП	П→х	ИП
П	х→П	П
$\frac{xy}{\downarrow}$	\longleftrightarrow	\longleftrightarrow
Farcsin	$F\sin^{-1}$	Farcsin
Farccos	$F\cos^{-1}$	Farccos
Farctg	Ftg^{-1}	Farctg
\div	\div	:
F ∅	F ∅	F ∅

Если же около названия помещен значок «*», это значит, что программу без малейших изменений можно вводить не только в те три калькулятора, о которых говорилось выше, но и в калькуляторы новых моделей — «Электроника МК-61», «Электроника МК-52».

Что же делать, если в распоряжении читателя только «Электроника МК-61» или «Электроника МК-52», а

звездочка у названия программы отсутствует? Оказывается, и в этом случае есть возможность запустить программу. Первый способ самый простой. Он заключается в том, что команды КИП↑ и КП↑ заменяют последовательностью других команд, например, ИПО ПЕ F ∅ КИПЕ и ИПО ПЕ F ∅ КПЕ соответственно. Фрагмент замены можно выносить в подпрограмму или встраивать в программу. Если фрагмент встраивается в программу, необходимо одновременно скорректировать адреса всех условных и безусловных переходов, включая адреса переходов после команд FLO, FL1, FL2, FL3. Иногда удобнее использовать замену ИПО 1+ПО F ∅ КИПО и ИПО 1+ПО F ∅ КПО соответственно.

Несмотря на всю простоту первого способа, читателю лучше пользоваться вторым. Он заключается в том, чтобы, разобравшись в программе и алгоритме, самостоятельно составить новую программу, используя все возможности своего микрокалькулятора. Этим способом придется пользоваться и владельцам микро-ЭВМ «Электроника МК-72», «Электроника МК-85», «Электроника БК-0010» и «Микроша».

В книге часто будет использоваться обозначение O,ЧЧММ. Оно означает, что требуется ввести исходное случайное число из диапазона от нуля до единицы, например, такое, где первые две цифры после запятой будут соответствовать текущему часу, а две последующие цифры — числу минут к моменту старта.

Еще одно обозначение — это кавычки. Так в книге обозначены те числа, которые надлежит считать с индикатора. Если число, напротив, не заключено в кавычки, это означает, что его нужно вводить с клавиатуры.

В заключение хочется сообщить имена авторов игр, включенных в сборник.

«Магический квадрат» — М. Вахтеров из Москвы, «Жонглер и шарик» — М. Шарий, Н. Узденников из Донецка. «Одномерный кубик Рубика», «Кольцо с секретом», «Поездка по городу» — Г. Славин из Тарту, «Регата» — Г. Гурьянов из Москвы, «Три шашки» — В. Рыбинский из Москвы, «Африканское сафари» — В. Кононов из Курска, «Кто первый» — Г. Белавин из Риги, «Учебный полет» — С. Чуприна из Новороссийска. В. Кононов написал также новый вариант игровой программы для игры А. Несчетного из Ленинграда — «Охота на лис».

В ПОИСКАХ УТРАЧЕННОГО ПОРЯДКА

Эта глава для тех, кто любит головоломки. Запаситесь терпением, ведь найти последовательность действий, ведущую к успеху, не очень просто. Наградой за долгие размышления будет не только успех в игре, но и отточенное умение мыслить логически.

Все игры, собранные в этой главе, составлены так, чтобы можно было играть одному. Но, конечно, интереснее устроить соревнование с друзьями, собирая, например, одномерный кубик Рубика. Каждый пусть вводит одинаковые исходные данные и программу, а соревноваться можно в том, кто быстрее справится с решением или добьется успеха меньшим числом ходов.

Каждую из игр нетрудно запрограммировать на БЕЙСИКе или ФОКАЛе для любой микро-ЭВМ, даже такой простой, как «Электроника МК-85». Сделать это можно, переводя помещенную здесь программу с языка калькулятора на язык имеющегося у вас компьютера, но полезнее составить свою собственную программу, используя описание игры и блок-схему.

«Магический квадрат»*

В этой игре девять чисел от единицы до девяти по определенным правилам могут перемещаться по полям разграфленного квадрата 3×3 . Игра напоминает

7	8	9
4	5	6
1	2	3

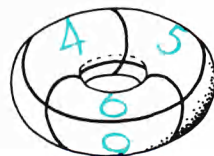
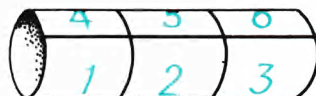


Рис. 1

сборку кубика Рубика. Правда, если бы для нее понадобилась объемная интерпретация, то на роль модели подошел бы не куб, а тор, поверхность которого разделена на девять пронумерованных сегментов.

Получить такой тор несложно. Мысленно изогнем наш квадрат и склеим верхнюю сторону с нижней. Получим трубочку (рис. 1). Изогнем трубочку и склеим ее концы, т. е. бывшие левую и правую стороны квадрата. Получится тор.

Образ тора позволяет проще понять, каким правилам подчиняются перемещения чисел по полям нашего квадрата. Будем считать, что при сдвиге какого-либо столбца цифр сверху вниз содержимое верхнего поля передвигается на среднее, содержимое среднего — на нижнее, содержимое нижнего — на верхнее. Продолжая аналогию с тором, скажем, что этот сдвиг равносителен повороту кольца тора.

Аналогично при сдвиге какой-либо строки квадрата слева направо содержимое левого поля попадает в среднее поле, содержимое среднего — в правое, содержимое правого — в левое. Впрочем, программа игры позволяет сдвигать лишь нижнюю строку, но это — не ограничение. В самом деле, сдвигая столбцы,



Каждый из желаемых сдвигов выполняется нажатием определенной цифровой клавиши (табл. 3) и вслед за нею клавиши С/П. После каждого перемещения на индикаторе демонстрируется нижняя строка квадрата. На индикатор можно вызвать и другие строки, нажимая цифровую клавишу и вслед за нею клавишу С/П. Расшифровка остальных цифр также указана в таблице. Разобравшись в программе, можно упростить игру, если ввести дополнительную команду перемещения, например сдвига средней строки.

Таблица 3

Операция	Нажимаемые клавиши	
Поворот нижнего ряда вправо	0	С/П
Сдвиг левого столбца вниз	1	С/П
Сдвиг среднего столбца вниз	2	С/П
Сдвиг правого столбца вниз	3	С/П
Вызов на индикатор нижнего ряда	5	С/П
Вызов на индикатор среднего ряда	7	С/П
Вызов на индикатор верхнего ряда	9	С/П

Работа с программой требует ввода исходных данных.

* Структура программы — совокупность ее отдельных фрагментов с указанием взаимосвязи между ними. Поясняя структуру программы, мы будем указывать адреса отдельных команд или единого по назначению фрагмента программы, а также приводить комментарий — для чего служат эти команды или фрагменты.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	КБПА	8—	28	1	01	46	FO	25
01	КБПВ	8L	29	0	00	57	П2	42
02	КБПС	8C	30	:	13	58	FO	25
03	БП	51	31	П2	42	59	П8	48
04	61	61	32	FO	25	60	В/О	52
05	БП	51	33	1	01	61	ИП3	63
06	17	17	34	0	00	62	ИП6	66
07	БП	51	35	:	13	63	ИП9	69
08	70	70	36	П3	43	64	П6	46
09	БП	51	37	FO	25	65	FO	25
10	77	77	38	2	02	66	П3	43
11	ПД	4Г	39	F10 ^x	15	67	FO	25
12	ИПО	60	40	X	12	68	П9	49
13	1	01	41	П1	41	69	В/О	52
14	+	10	42	В/О	52	70	ИП4	64
15	ПО	40	43	ИП1	61	71	ИП5	65
16	КППД	—Г	44	ИП4	64	72	ИП6	66
17	ИП1	61	45	ИП7	67	73	+	10
18	ИП2	62	46	П4	44	74	+	10
19	+	10	47	FO	25	75	БП	51
20	ИП3	63	48	П1	41	76	22	22
21	+	10	49	FO	25	77	ИП9	69
22	С/П	50	50	П7	47	78	ИП8	68
23	БП	51	51	В/О	52	79	+	10
24	11	11	52	ИП2	62	80	ИП7	67
25	ИП3	63	53	ИП5	65	81	+	10
26	ИП2	62	54	ИП8	68	82	БП	51
27	ИП1	61	55	П5	45	83	22	22

Рис. 3

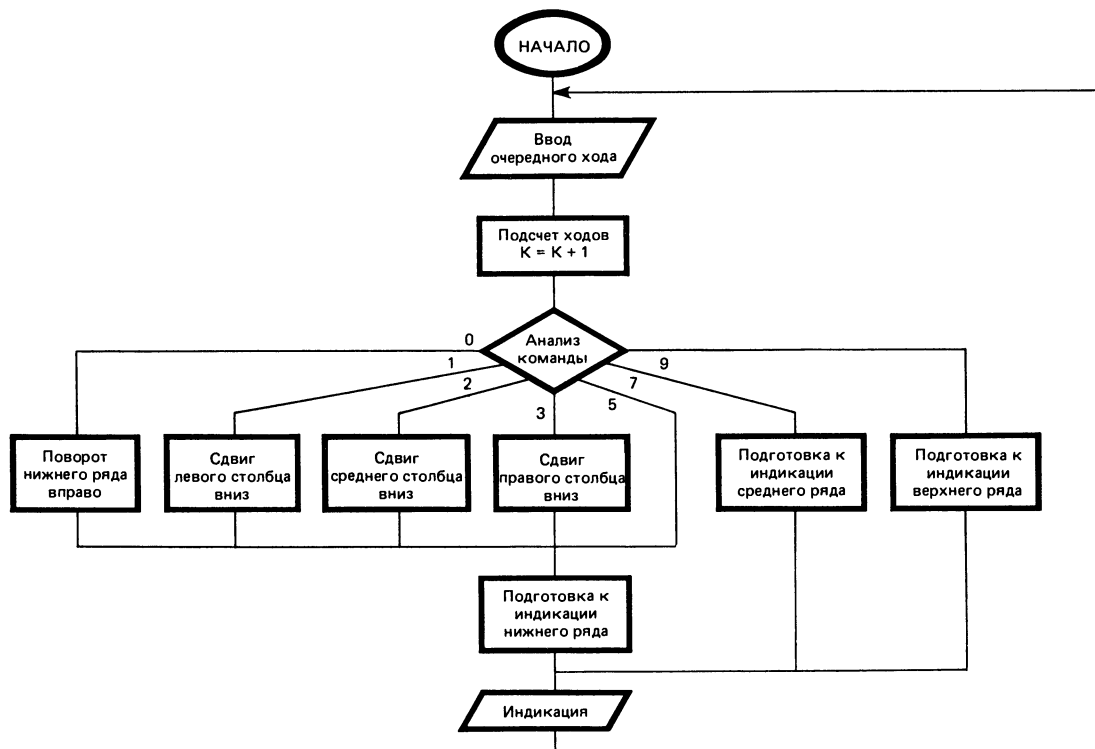


Рис. 4

07—08, 70—76: вызов на индикатор среднего ряда.

09—10, 77—83: вызов на индикатор верхнего ряда.

11, 16: анализ очередного хода.

12—15: счетчик ходов.

23—24: переход после индикации на блок анализа очередного хода.

Внутренние связи программы хорошо видны на блок-схеме (рис. 4).

Первый овальный блок означает ввод исходных данных и переход на одиннадцатый адрес по команде БП 11. Следующий блок «запоминает» код выбранной операции, т. е. ход игрока. Предположим для определенности, что нажата клавиша 7. Семерка отправится в регистр Д, за это отвечает команда ПД. Следующий блок — «подсчет ходов». Каждый новый ход добавляет единицу к содержимому регистра 0. По команде ИПО в любой момент можно узнать, сколько ходов уже сделано.

Основной блок в программе — «анализ команды». При всей своей важности он предельно краток — состоит из одной лишь команды КППД. Это команда косвенного обращения к подпрограмме. В зависимости от содержимого регистра Д, т. е. от того, какую мы нажимали клавишу, делая ход, управление переходит к одному из квадратов, что помещены под ромбом. Команды с 70 по 77 соберут на индикаторе изображение среднего ряда магического квадрата, а команда С/П, обозначенная на блок-схеме параллелограммом, остановит вычисления. Здесь то и можно обдумать следующий ход.

Пользуясь блок-схемой, нетрудно написать программу для любого микрокомпьютера, например для БК-0010.

«Одномерный кубик Рубика»

«Одномерный кубик Рубика» — это пластинка, разделенная на зоны. Каждая зона окрашена в свой цвет, всего зон восемь. Цель игры, как и в случае с традиционным кубиком Рубика, — упорядочить цветовую комбинацию, выстроить семь цветов так, как они расположены в спектре.

Разрешается только один вид перестановок: поворот блока, состоящего из

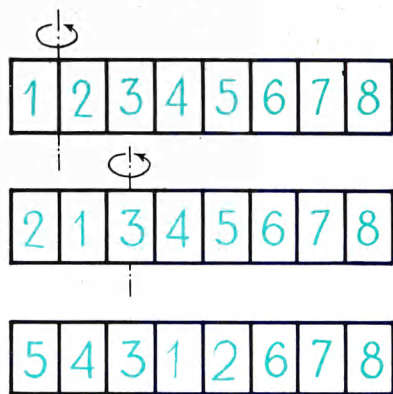


Рис. 5

любого числа зон, на 180° относительно плоскости пластинки. Блоком условимся называть несколько соседних зон, начиная от левого края пластинки. Поворот блока, состоящего из одной зоны, не имеет смысла, так как такой ход не меняет расположения цветов. Итак, в распоряжении играющего остаются возможные повороты блоков из двух, трех, ... восьми зон.

В варианте этой игры, разработанном для программируемого микрокалькулятора, цвета заменены цифрами от единицы до восьми. Изображение пластинки до и после каждого хода высвечивается на индикаторе. Поворачивать можно блоки, примыкающие к левому краю индикатора. Возьмем, например, в качестве исходной комбинации «12345678» (рис. 5), повернем блок из двух цифр и получим «21345678». Если поворачивается блок из нечетного числа цифр, то центральная цифра остается на месте. Так, в комбинации «21345678» поворот пяти цифр приводит к комбинации «54312678».

Исходную комбинацию придумывает сам игрок. Вводить ее в микрокалькулятор следует цифра за цифрой, перемежая ввод цифр нажатиями клавиши С/П. Пусть нужно ввести комбинацию «12345678». Ввод начинается с цифры 1, далее нажимаем клавишу С/П, затем 2 С/П и т. д. Наконец, вводим 8 С/П, и через 13 с на индикаторе появляется изображение пластинки.

Теперь задаем количество зон, которые необходимо повернуть, — размер

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	П8	48	33	БП	51	66	76	76
01	7	07	34	23	23	67	ИП3	63
02	ПО	40	35	Сх	ОГ	68	ИП4	64
03	↔	14	36	Сх	ОГ	69	ИП5	65
04	С/П	50	37	ИП7	67	70	ИП6	66
05	КП↑	LE	38	ИП8	68	71	П3	43
06	FLO	5Г	39	БП	51	72	FO	25
07	04	04	40	93	93	73	П4	44
08	1	01	41	Сх	ОГ	74	FO	25
09	0	00	42	ИП6	66	75	П5	45
10	ПС	4С	43	ИП8	68	76	↔	14
11	7	07	44	П6	46	77	П6	46
12	ПО	40	45	БП	51	78	9	09
13	ИПС	6С	46	94	94	79	ИПД	6Г
14	↑	OE	47	БП	51	80	—	11
15	↑	OE	48	78	78	81	ПА	4—
16	ИП8	68	49	Сх	ОГ	82	1	01
17	Х	12	50	Сх	ОГ	83	+	10
18	КИП↑	ГЕ	51	Сх	ОГ	84	ПВ	4L
19	+	10	52	БП	51	85	КИПА	Г—
20	FLO	5Г	53	78	78	86	КИПВ	ГL
21	17	17	54	Сх	ОГ	87	ИП7	67
22	П9	49	55	Сх	ОГ	88	ИП8	68
23	С/П	50	56	Сх	ОГ	89	КПА	L—
24	ПД	4Г	57	ИП5	65	90	FO	25
25	5	05	58	ИП6	66	91	КПВ	LL
26	Х	12	59	БП	51	92	FO	25
27	2	02	60	75	75	93	П7	47
28	7	07	61	Сх	ОГ	94	↔	14
29	+	10	62	ИП4	64	95	П8	48
30	ПО	40	63	ИП6	66	96	БП	51
31	КБП↑	8Е	64	П4	44	97	11	11
32	ИП9	69	65	БП	51			

Рис. 6

блока, нажимаем клавишу С/П, и через 15—25 с на индикаторе появляется результат перестановки, т. е. новая исходная ситуация.

Перед началом игры необходимо решить, какая цифровая комбинация должна получиться в итоге. Например, из возрастающей последовательности «12345678» необходимо получить убывающую «87654321». Стоит разом повернуть все восемь сегментов, и задача будет решена. Гораздо труднее было бы от приведенной исходной ситуации прийти, например, к такой: «18273645».

Если игра с восемью различными цифрами покажется трудной, можно упростить задачу — в качестве исходного числа выбрать комбинацию с повторяющимися цифрами, например, «13122312», и расставить цифры в порядке возрастания или убывания.

Инструкция:

1. Ввести программу (рис. 6).

2. Ввести исходную комбинацию: первая цифра В/О С/П «первая цифра» вторая цифра С/П «вторая цифра»...

восьмая цифра С/П «исходная комбинация».

3. Ввести размер блока, который нужно повернуть, С/П «результат перестановки». Выполнять п. 3 до успешного завершения игры.

4. Для новой партии перейти к п. 2.

Если вы захотите сыграть в эту увлекательную игру с друзьями, то придется подсчитать число сделанных ходов или записывать эти ходы. Победителем окажется тот, кому удастся решить головоломку за меньшее число ходов.

Контрольный пример поможет быстро проверить программу. Вводим исходную комбинацию «19781982»:

1 С/П «1» 9 С/П «9» 7 С/П «7» 8 С/П «8»
1 С/П «1» 9 С/П «9» 8 С/П «8» 2 С/П «19781982».

Начинаем упорядочивать комбинацию:

4 С/П «87911982»

2 С/П «78911982»

3 С/П «98711982»

5 С/П «11789982»

7 С/П «89987112»

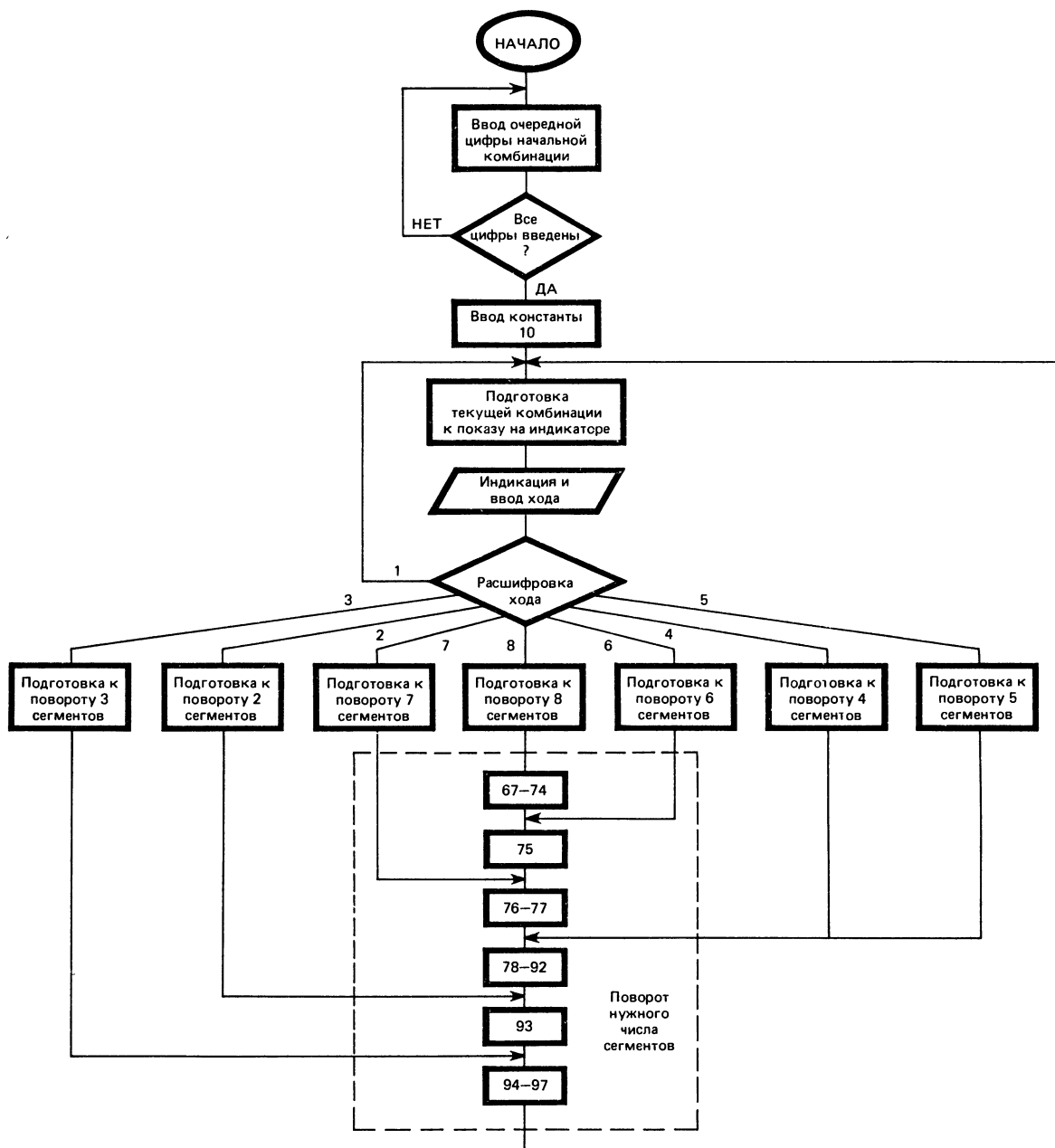


Рис. 7

3 С/П «99887112»

8 С/П «21178899»

3 С/П «11278899»

Игра закончена за восемь ходов. Может быть, вам удастся отыскать более короткий путь к победе?

Программа этой игры может удивить

не только новичка. Здесь и неописанные в руководстве калькулятора команды КБП↑, КП↑, КИП↑, и вроде бы нелепые сочетания БП 78 Сх Сх Сх БП 78 Сх Сх Сх. Тем интереснее будет самостоятельно разобраться в программе, не заглядывая в блок-схему (рис. 7).

Структура программы:

00—07: ввод исходной комбинации.

08—10: ввод константы 10.

11—23: подготовка текущей комбинации к показу на индикаторе и остановка для индикации.

24: ввод кода хода.

25—31: расшифровка кода.

32—34: переход на блок 11—23, так как поворот одного сегмента не имеет смысла.

35—36: «заплата», необходимая для того, чтобы следующий фрагмент начался с нужного адреса, вычисленного в блоке 25—31.

37—40, 93—97: поворот двух сегментов.

41: «заплата».

42—46, 94—97: поворот трех сегментов.

47—48, 78—97: поворот четырех сегментов.

49—51: «заплата».

52—53, 78—97: поворот пяти сегментов.

54—56: «заплата».

57—60, 75—97: поворот шести сегментов.

61: «заплата».

62—66, 76—97: поворот семи сегментов.

67—97: поворот восьми сегментов.

Блок-схема (см. рис. 7) предназначена для тех, кто решит написать программу этой игры для микрокомпьютера. В программе на БЕЙСИКе блок расшифровки кода хода удобно реализовать командами GOTO A или GOSUB A, где A — это выражение, величина которого, т. е. адрес перехода, определяется сделанным ходом.

Непростая задача стоит и перед владельцами калькуляторов новых моделей. В системах команд МК-61 и МК-52 команда КИП↑ и ей подобные отсутствуют. Прежде чем приниматься за головоломку, придется немало поломать голову над тем, какие изменения в программе нужно сделать, чтобы она заработала на новых калькуляторах.

«Кольцо с секретом»*

Представьте прямоугольное гнездо — желоб, в котором разместились восемь цветных шариков. Некоторые из этих шариков, а возможно и все восемь, окрашены в разные цвета. Противоположные концы гнезда соединяет круговой желобок. По нему крайний левый

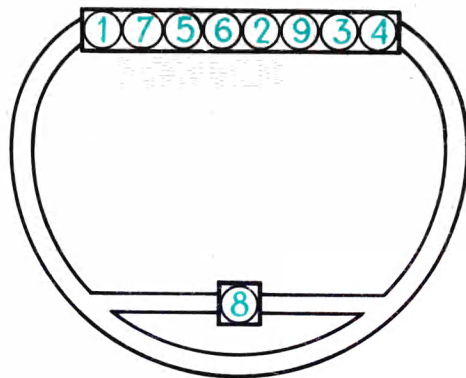


Рис. 8

шарик без труда можно перекатить так, чтобы он стал крайним, но уже правым. Можно вернуть его обратно, а вслед за этим проделать то же с крайним правым шариком. Ясно, что сколь бы долго мы ни перекачивали шарики по полученному кольцу — слева-направо или справа-налево, порядок их расположения не нарушится. Для того чтобы шарики можно было менять местами, на кольцевой дорожке устроена хитроумная ловушка. Как она действует — показано на рис. 8. Шарик, катящийся слева или справа, можно поймать ловушкой, при этом тот шарик, что скрывался в ловушке до этого, будет продолжать движение в заданном направлении вместо пойманного. Одновременно в ловушке может находиться лишь один шарик.

Умело используя ловушку и круговой желобок, можно поменять местами любые два шарика. В нашем распоряжении четыре варианта возможных перемещений: слева-направо, слева-направо через ловушку, справа-налево и справа-налево через ловушку.

Цель игры — привести исходное случайное расположение цветных шариков к заранее определенной цветовой последовательности. Введя в микрокалькулятор программу, например ту, что на рис. 9, можно устроить турнир, победителем которого окажется игрок, потративший на решение задачи меньшее число ходов. Конечно, на индикаторе машинки нельзя получить изображения шариков, но и без этого игра получается увлекательной, нужно только договориться, что каждая цифра (от единицы до девяти) обозначает свой цвет.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	FV	21	33	БП	51	66	ПО	40
01	3	03	34	16	16	67	0	00
02	—	11	35	ИП7	67	68	ИПВ	6L
03	2	02	36	ПП	53	69	8	08
04	X	12	37	21	21	70	X	12
05	ПС	4C	38	ИП8	68	71	Ел	20
06	КБПС	8C	39	X	12	72	+	10
07	ИП7	67	40	ИП2	62	73	ПП	53
08	БП	51	41	БП	51	74	24	24
09	11	11	42	30	30	75	ПВ	4L
10	ИП1	61	43	ИП1	61	76	9	09
11	ПП	53	44	↑	OE	77	X	12
12	21	21	45	ПП	53	78	1	01
13	ИП8	68	46	21	21	79	+	10
14	X	12	47	X	12	80	ПС	4C
15	ИПД	6Г	48	ИП2	62	81	КИПС	ГС
16	+	10	49	↔	14	82	FO	25
17	П5	45	50	П2	42	83	FO	25
18	С/П	50	51	↔	14	84	ИП1	61
19	БП	51	52	ИП7	67	85	X	12
20	00	00	53	БП	51	86	ИПС	6C
21	ИП5	65	54	14	14	87	+	10
22	↔	14	55	ПВ	4L	88	FLO	5Г
23	:	13	56	Сх	OG	89	68	68
24	ПД	4Г	57	ВП	OC	90	П5	45
25	КИПД	ГГ	58	7	07	91	ИПД	6Г
26	FO	25	59	П7	47	92	П2	42
27	ИПД	6Г	60	1	01	93	КИП2	Г2
28	—	11	61	0	00	94	КИП2	Г2
29	В/О	52	62	П1	41	95	ИП5	65
30	ИПД	6Г	63	X	12	96	БП	51
31	П2	42	64	П8	48	97	18	18
32	FO	25	65	8	08			

Рис. 9

Итак, всего шариков-цифр девять. Восемь разместились в гнезде — на индикаторе, и одна цифра — в ловушке — в регистре 2. Из исходной случайной последовательности цифр надо получить упорядоченную. Например, из последовательности (рис. 10) нужно получить ту, что на рис. 11, расположив цифры на индикаторе в порядке возрастания, а в ловушку поместить наибольшую — в данном случае 9.

Для того чтобы двигать шарики-цифры, достаточно запомнить четыре команды. Если обозначить левый край последовательности цифр в гнезде цифрой 4, а правый край — 6, то команды движения запомнить очень легко: 46 — слева-направо, 64 — справа-налево. Если помнить, что для ловушки выделен регистр 2, то несложно разобраться в назначении следующих двух команд 426 и 624. Команда 426 означает, что самый левый шарик из гнезда, т. е. старший разряд числа на индикаторе, попадет в регистр 2, а цифра из регистра 2 приписется к числу из гнезда справа, восста-

6 1 3 5 7 9 5 5

5

Рис. 10

1 3 5 5 5 5 6 7

9

Рис. 11

навливая исходную разрядность числа на индикаторе. Командой 624, наоборот, можно приписать цифру из ловушки к числу на индикаторе слева, тогда как

младший разряд отправится на хранение в ловушку. Запомнить команды помогает несложное мнемоническое правило: представьте, что гнездо находится под клавишами 4—5—6, а ловушка — под клавишей 2.

Микрокалькулятор выполняет команды в строгом соответствии с заложенными в программу формулами, например команде 426 соответствуют формулы:

$$\begin{aligned}T &= [ЧГ/10^7] \\ ЧГ &= 10^8 \cdot \{ЧГ/10^7\} + R2 \\ R2 &= T.\end{aligned}$$

Здесь R2 — число из ловушки, ЧГ — число в гнезде. Квадратные скобки означают операцию отделения целой части числа, а фигурные — дробной части.

Каждый раз перед игрой калькулятор заново выдает случайную начальную комбинацию, иначе головоломка быстро наскучит. Чтобы каждая новая игра начиналась с новой цифровой комбинации, перед игрой требуется набрать стартовое число в форме О, ЧЧММ. Если программа (см. рис. 9) уже набрана, можно запускать машинку клавишами БП 55 С/П. Меньше чем через 2 мин на индикаторе появится изображение гнезда. При необходимости нетрудно заглянуть в ловушку — нажать клавиши ИП2. В этом случае на индикаторе тотчас же появится недостающее девятое число. Если требуется еще раз взглянуть на гнездо, можно нажать клавиши ИП5 или ↵.

Теперь, манипулируя клавишами 4, 6 и 2, начинаем приводить последовательность к необходимому порядку: вслед за кодом выбранного действия нажимаем клавишу С/П. Каждый раз через 8 с на индикаторе появляется результат операции — пора делать новый ход.

Головоломка не очень проста, иногда может казаться, что выхода из создавшейся ситуации нет, однако решение имеется в любом случае.

Инструкция:

1. Ввести программу (см. рис. 9).
 2. О, ЧЧММ БП 55 С/П «гнездо» код команды С/П «гнездо» код С/П...
 3. Для новой игры БП 6 5 С/П.
- Структура программы:

00—06: запись вычисленного адреса перехода в регистр С и переход по этому адресу. Если была введена команда 46, то переход осуществляется на адрес 07,

если 64, то на адрес 10, 426 — на 35, 624 — на адрес 43.

07—09: подготовка константы $1 \cdot 10^7$, необходимой для вычислений по формулам, и передача управления на команды обращения к подпрограмме 11, 12.

18—20: команда останова для индикации гнезда и переход к новым действиям по введенной команде.

21—23: подготовка содержимого гнезда к перестановке.

24—29: подпрограмма выделения целой и дробной частей числа.

30—34: подготовка нового содержимого для ловушки и передача управления на команду формирования нового содержимого гнезда.

35—42: формирование нового содержимого гнезда и ловушки по команде 426.

43—54: формирование нового содержимого гнезда и ловушки по команде 624.

55: запоминание стартового случайного числа.

56—59: формирование константы $1 \cdot 10^7$ и запись ее в регистр 7.

60—62: формирование константы 10 и запись ее в регистр 1.

63—64: формирование константы 10^8 и запись ее в регистр 8.

65—66: начальная установка счетчика цифр в гнезде на 8.

68—75: формирование псевдослучайного числа ξ по формуле $\xi = \{8\xi + \pi\}$, где фигурные скобки означают выделение дробной части числа.

76—86: формирование целого псевдослучайного числа от единицы до девяти.

87, 67: формирование числа — содержимого гнезда.

88—89: счетчик цифр в гнезде.

90: число в гнезде сформировано и записывается в регистр 5.

91—94: формирование случайного целого числа от единицы до девяти — содержимого ловушки.

95—97: вызов в регистр X содержимого гнезда и останов для индикации.

Контрольный запуск программы необходим для того, чтобы убедиться, правильно ли набран текст программы. Выбираем стартовое число, например О, ЧЧММ=0,1835.

0,1835 БП 55 С/П «61357955». Нажимаем ИП2 «5» — итак, в гнезде «61357955» и в ловушке — «5». Цель игры — преобразовать исходное число

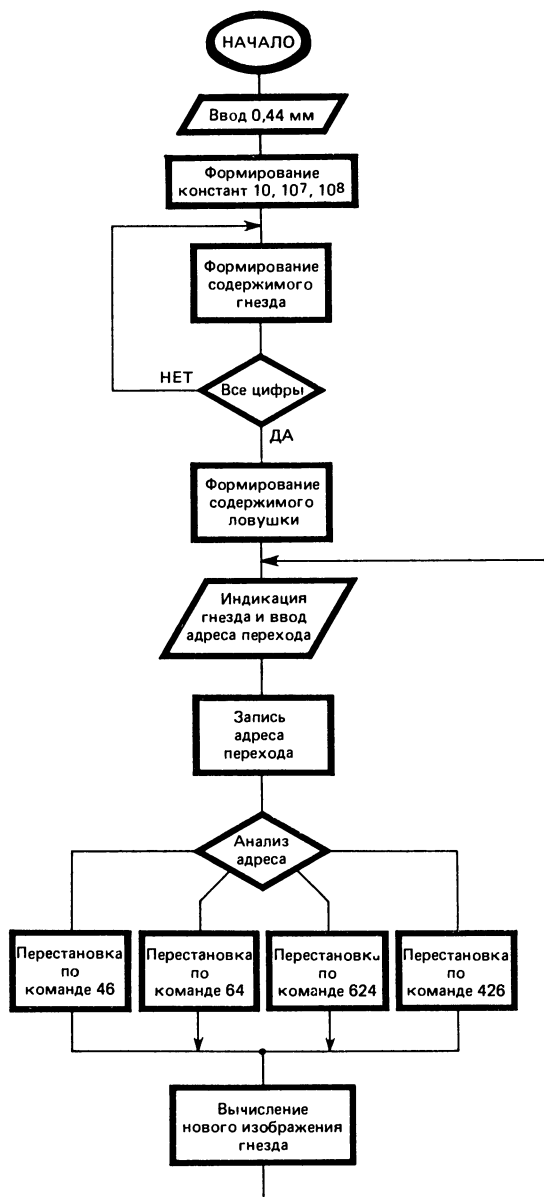


Рис. 12

к виду «13555567», а в ловушке оставить «9».

Начинаем: 64 С/П «56135795» ИП2 «5» 64 С/П «55613579» ИП2 «5» 624 С/П «55561357» ИП2 «9» ... и т. д.

Если через восемнадцать ходов цель достигнута, значит, можно смело бросить вызов товарищам: пусть самого искусного игрока выявит состязание.

Тем, кто решил переложить програм-

му на язык микрокомпьютера, советуем воспользоваться иным методом представления гнезда.

Дело в том, что в некоторых микрокомпьютерах ошибки округления могут порой изменять цифры, составляющие гнездо. Лучше представлять гнездо не как переменную, а как одномерный массив из восьми ячеек. Тогда не нужно будет использовать формулы, достаточно передвигать нужным образом сами цифры — содержимое ячеек подобно тому, как это делается в стеке микрокалькулятора по команде F0. Блок-схема программы изображена на рис. 12.

ПОЕДИНКИ СО СЛУЧАЕМ

В этой главе описаны игры, исход которых — победа или поражение — зависит не только от мастерства игрока. В ход игры то и дело вмешиваются случайности. Они могут помочь неопытному, но удачливому игроку, но точно так же они способны и расстроить самые надежные планы хитроумного несчастливца.

Такие игры, где ходы калькулятора не вычисляются по раз и навсегда заложенным в программу формулам, а подчиняются законам случайных чисел, характерны разнообразием ситуаций, возникающих в ходе поединка. Из механизма, всецело послушного командам своего владельца, микрокалькулятор превращается в интересного соперника, который на равных ведет с человеком борьбу за победу в игре.

Для того чтобы получать случайные числа, на использовании которых строятся алгоритмы таких игр, во многих языках программирования предусмотрены специальные функции. В большинстве диалектов БЕЙСИКа случайное число можно получить с помощью функции RND. Каждое обращение к ней дает новое случайное число из диапазона от нуля до единицы. Владельцы БК-0010 первых годов выпуска смогут воспользоваться аналогичной функцией ФОКАЛа под названием FRAN. В калькуляторах «Электроника БЗ-34», «Электроника МК-54», «Электроника МК-56» встроенный датчик псевдослучайных чисел отсутствует, поэтому придется использо-

вать специальные программы для получения таких чисел. В новых моделях «Электроника МК-61», «Электроника МК-52» для этой цели предназначена команда КСч.

«Опасный рейс»*

Вообразите, что вы капитан океанского лайнера. Ваша цель — привести корабль в порт назначения, на остров, расположенный в Саргассовом море. На пути корабль подстерегают опасности — плотные скопления водорослей, буйно разросшихся под тропическим солнцем. Это настоящие плавучие острова, едва различимые на поверхности океана. Чаше всего скопления водорослей образуют круг, пересесть границу которого — значит рисковать повредить винт. Дело осложняется тем, что такие плавучие острова медленно дрейфуют, а потому не значатся даже на самых лучших картах.

Для прокладки курса понадобится миллиметровка. Не обойтись и без компаса (его можно нарисовать). Порт расположен в центре карты (рис. 13). Ему соответствуют координаты $(0,0)$. По данным последней воздушной разведки известно, что вокруг острова имеется свободная от водорослей акватория с центром на острове и диаметром шесть миль. Также известно, что плавучих островов всего три — каждый диаметром 18 миль. Где именно они дрейфуют, в настоящий момент неизвестно, однако компьютерная модель состояния моря показала, что центр каждого острова расположен не ближе, чем в 12 милях от точки $(0,0)$.

Карта разделена координатными осями на четыре квадрата, они пронумерованы цифрами 1, 2, 3, 4. Плавучие острова могут оказаться в разных квадратах или в одном из них. Не исключено и то, что опасные зоны могут частично или полностью перекрываться. В этом случае задача существенно усложнится.

В игре нельзя обойтись без помощника, который должен незаметно для капитана ввести координаты центров плавучих островов. Если в распоряжении капитана «Электроника МК-52»,

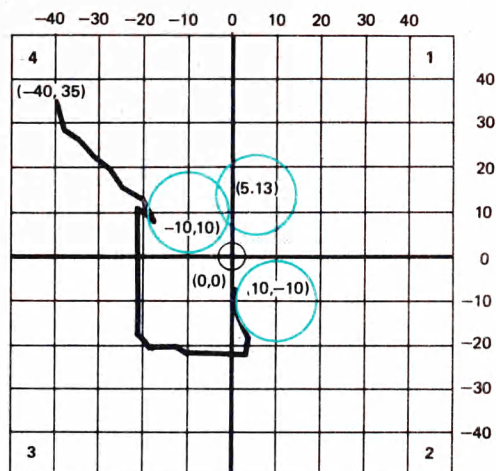


Рис. 13

то помощник сможет заменить специальная программа.

Поясним, как необходимо разместить координаты островов в памяти микрокалькулятора. Предположим, что центр первого острова должен оказаться в точке $(10, -10)$. Тогда координату $X_1=10$ следует записать в регистр Р8, а $Y_1=-10$ в регистр Р7. Аналогично записываются и координаты X_2, Y_2 и X_3, Y_3 — соответственно (-10) в Р6, (10) в Р5, (5) в Р4, (13) в Р3.

Необходимо задать начальные координаты корабля. Помощник выбирает точку на карте, но не ближе, чем на расстоянии 40 миль от точки $(0,0)$. Второе условие — выбранная точка с координатами X, Y не должна быть ближе пяти миль к любому из плавучих островов.

Вводим X -координату (-40) в РА, Y -координату (35) в РВ. Осталась лишь одна цифра — скорость корабля. Когда игра начинается, его скорость составляет пять узлов. Это значит, что за каждый ход корабль преодолевает пять миль пути. Пятёрку следует записать в Р1 (5 П1).

Если корабль попал в опасный район, он неминуемо получит повреждения. Каждый раз, когда это случается, скорость корабля падает на один узел. Ясно, что после пятой аварии корабль не сможет продолжать движение. Калькулятор сообщит об этом, высветив на индикаторе сообщение ЕГГОГ.

Таблица 4

Координаты островов			Скорость корабля	Координаты корабля
первого	второго	третьего		
10 —10 П8 П7	5 13 П6 П5	—10 10 П4 П3	5 П1	—40 35 ПА ПВ

Программа введена? Проверьте соответствие исходных данных (табл. 4).

Можно приступить к игре. Для этого передвинем в положение Г переключатель Р-Г и нажмем клавиши В/О, С/П. Через 10 с на индикаторе появится сообщение о текущей позиции корабля в виде ХХ, УУОООК, ХХ — это координата «запад-восток» всегда со знаком «+», УУ — координата «север-юг» также всегда положительная, а К — указывает номер квадрата, в котором находится корабль. Предположим, на индикаторе светится 40,350004. Это соответствует примеру ввода исходных данных (см. табл. 4) и карте. Таким образом, корабль расположен в северо-западном районе.

Можно вводить курс, которым будет следовать корабль. Вводить его необходимо в градусах; от 0 до 360. Направимся, к примеру, на юго-юго-восток: 165 С/П (рис. 14). Если корабль не попал в опасный район, на индикаторе через минуту появятся новые координаты. В нашем случае это позиция 39,30 в четвертом квадрате. Если вам не повезло, и корабль, оказавшись в опасной зоне, получил повреждения, то на индикаторе появится число от 0 до 9, характеризующее расстояние от корабля до центра плавучего острова. Нажмите клавишу С/П, и через 20 с индикатор высветит текущие координаты.

При особо неудачном стечении обстоятельств корабль может оказаться в районе, где накладываются друг на друга смежные опасные зоны. В этом случае индикатор сначала сообщит о расстоянии до центра первого острова, а после нажатия С/П можно будет узнать о расстоянии до другого. Удвоится, конечно, и число разрушений.

Что же предпринять, если корабль оказался в аварийной ситуации? Такой

Х	У	Квадрат	Курс
40	35	4	165
39	30	4	135
35	27	4	135
32	23	4	135
28	20	4	135
25	16	4	135
21	13	4	135
	7.56 — авария, С/П		
18	09	4	300
21	11	4	180
21	07	4	180
21	03	4	180
21	01	3	180
21	05	3	180
21	09	3	180
21	13	3	180
21	17	3	130
18	20	3	100
14	20	3	100
10	21	3	92
6	21	3	92
2	21	3	92
2	21	2	15
3	18	2	350
	8.49 — авария, С/П		
2	14	2	340
	8.74 — авария, С/П		
1	11	2	350
1	09	2	350
1	07	2	350
0	05	2	360
	3.1415926 — победа!		

Рис. 14

вопрос может возникнуть у неопытного капитана. Есть две возможности: либо попытаться вернуться и обойти плавучий остров, либо прорываться напрямик в надежде, что корабль не получит слишком тяжелых повреждений и сохранит плавучесть и ход. Как только корабль достигнет свободной от опасности зоны вокруг острова, можете считать, что удача сопутствовала вам и вы с честью выполнили трудное задание.

Калькулятор сообщит о победе, высветив на индикаторе число 3,1415926. Для новой игры повторите ввод исходных данных согласно таблице.

Полезным будет разбор итогов игры.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИПА	6—	33	↔	14	66	ИПД	6Г
01	ПП	53	34	ИП1	61	67	С/П	50
02	73	73	35	X	12	68	FLO	5Г
03	ИПВ	6L	36	ИПВ	6L	69	48	48
04	ПП	53	37	+	10	70	КБПО	80
05	73	73	38	ПВ	4L	71	Fπ	20
06	2	02	39	Fx ²	22	72	С/П	50
07	F10x	15	40	+	10	73	Fx ²	22
08	:	13	41	9	09	74	F√	21
09	+	10	42	Π2	42	75	7	07
10	ПД	4Г	43	—	11	76	F10x	15
11	ПП	53	44	Fx ≥ 0	59	77	+	10
12	81	81	45	71	71	78	FBx	0
13	6	06	46	3	03	79	—	11
14	F10x	15	47	ΠO	40	80	В/О	52
15	:	13	48	ИПА	6—	81	ИПА	6—
16	ИПД	6Г	49	КИП2	Г2	82	Fx < 0	5С
17	+	10	50	—	11	83	91	91
18	С/П	50	51	Fx ²	22	84	ИПВ	6L
19	↑	OE	52	ИПВ	6L	85	Fx < 0	5С
20	9	09	53	КИП2	Г2	86	89	89
21	0	00	54	—	11	87	3	03
22	↔	14	55	Fx ²	22	88	В/О	52
23	—	11	56	+	10	89	4	04
24	Fsin	1C	57	F√	21	90	В/О	52
25	FBx	0	58	ПД	4Г	91	ИПВ	6L
26	Fcos	1Г	59	9	09	92	Fx < 0	5С
27	ИП1	61	60	—	11	93	96	96
28	X	12	61	Fx < 0	5С	94	2	02
29	ИПА	6—	62	68	68	95	В/О	52
30	+	10	63	FL1	5L	96	1	01
31	ПА	4—	64	66	66	97	В/О	52
32	Fx ²	22	65	K—	27			

Рис. 15

Попробуйте отметить на карте координаты плавучих островов, с какими «почастливилось» встретиться кораблю. Сравните их с истинными координатами, записанными в РЗ—Р8.

Не запрещено и произвольно изменять правила игры. Например, можно потребовать, чтобы помощник предлагал четыре варианта исходного положения корабля в разных квадратах на выбор капитана. Когда будут введены координаты островов, пусть капитан сам выберет, из какого квадрата отправится в путь корабль. В общем, все зависит от договоренности игроков.

Структура программы (рис. 15):

00—18: синтез сообщения о координатах корабля и квадрате.

18—38: запрос курса корабля и вычисление его новых координат.

39—45: проверка «не достигнут ли порт назначения?».

46—62: проверка «нет ли аварийной ситуации?».

63—65: уменьшение скорости корабля и проверка «не закончена ли игра?».

66—67: индикация расстояния до центра «острова», с которым произошло столкновение.

68—69: проверка «возможны ли еще аварийные ситуации?».

70: переход к началу программы.

71—72: индикация успешного завершения игры.

73—80: вычисление целой части абсолютного значения числа.

81—97: определение местоположения корабля.

«Где же обещанные случайные числа?» — спросит иной читатель. Роль датчика, который не поместился в программу, пришлось на этот раз исполнять помощнику игрока.

Можно заметить, что программа для калькулятора не совсем соответствует блок-схеме алгоритма (рис. 16), отличающаяся от нее некоторыми фрагментами. Блок-схема составлена с тем расчетом, чтобы по ней можно было написать программу для любого программируемого микрокалькулятора или микро-ЭВМ. Например, блоки, соответствующие коман-

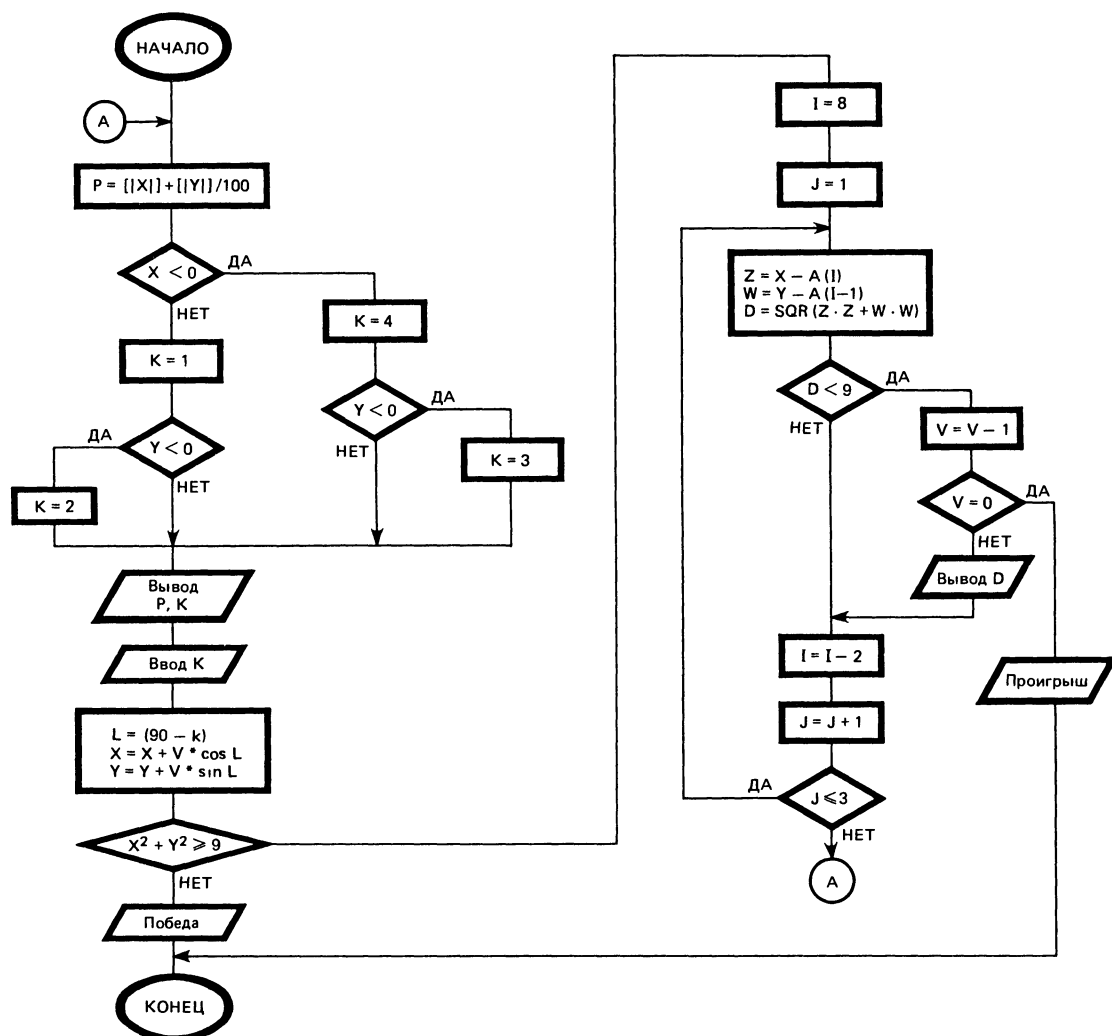


Рис. 16

```

30 P = INT ABS X + (INT ABS Y) / 100
40 IF X < 0 GOTO 80
50 K = 1
60 IF Y > 0 THEN K = 2
70 GOTO 100
80 K = 4
90 IF Y < 0 THEN K = 3

```

Рис. 17

дам 00—18, на БЕЙСИКе могут выглядеть так (рис. 17).

Нетрудно было записать программу на ФОКАЛе для БК-0010, но лучше оставим эту задачу в качестве упражнения для начинающих.

Поскольку возможности микро-ЭВМ шире, чем у самого сложного программируемого микрокалькулятора, программе стоит дополнить несколькими командами, которые создадут на экране карту, отметят пройденный путь и аварийные ситуации.

Впрочем, как всегда, дело за фантазией играющего.

«Электронные городки»

Электронные городки не совсем похожи на обычные. В чем же отличие? Если в традиционной игре броски произ-

водят с определенных расстояний: 13 м — «кон», 6,5 м — «полу-кон», то в нашем случае расстояние определяется игроком и может быть любым от 11 до 20 м. Вместо квадратного «города» размерами 2×2 м используется линия, на которой расставляются городки. Что происходит в «городе», показывает индикатор. Вместо пяти городков на линии расставляются от одного до восьми. Незанятые позиции обозначаются единицами, а те, где установлен городок, — двойками. Таким образом, на индикаторе в начале игры может быть число 22222222, а в конце должно быть 11111111.

Длина биты выбрана такой, чтобы в случае удачного броска она сшибала пять стоящих рядом городков. Удачным броском условимся называть такой, когда, несмотря на вращение в полете, бита приходит к городу параллельно линии, а не под углом — иными словами, длина проекции биты на линию совпадает с длиной биты. Наименее удачным — бросок, если бита пролетает сквозь город торцом, не задев ни одного городка.

Цель такова же, как и в обычной игре, — наименьшим числом ударов выбить все городки с линии.

Теперь коротко о правилах диалога (а ведь именно в этом режиме проходят все игры с микрокалькулятором).

После того как программа введена, требуется ввести признак окончания игры — число из восьми единиц — в Р9. Это следует сделать лишь раз — перед началом игры.

Городки-двойки располагают в регистрах с восьмого по первый. Незаполненные позиции на линии отмечают единицей в соответствующем разряде (регистре). О количестве фигур и их конфигурациях заранее договариваются с партнером. Обычное число фигур — пятнадцать.

Пусть выбрана такая фигура: два городка посередине линии. Чтобы ее установить, записываем двойку в Р4 и Р5, а в регистры с Р1 по Р3 и в регистры с Р6 по Р8 помещаем единицу.

Чтобы вычисления происходили в градусах, установите в нужное положение переключатель Р-Г.

Коль скоро городки расставлены по местам, можно запускать программу кла-

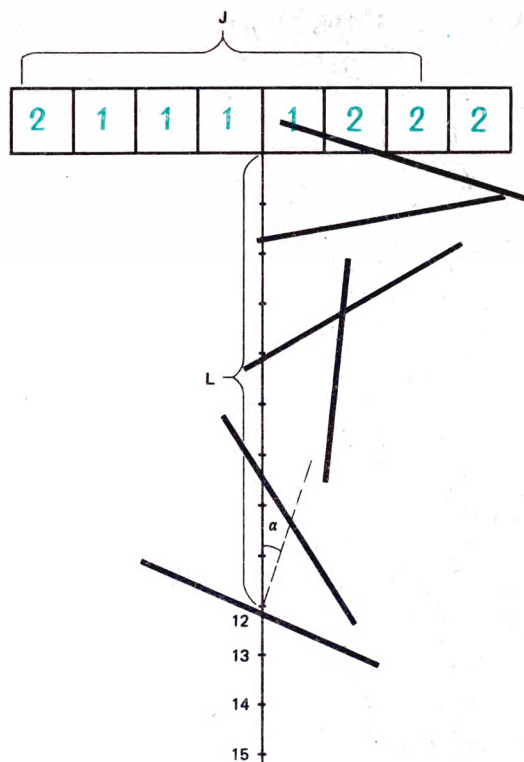


Рис. 18

вишами В/О, С/П. Через 15 с на индикаторе появится изображение города. Выберем дистанцию для броска, например, 12 м (12 С/П) и угол α (рис. 18) — от -30° до 30° , нажмем С/П.

Менее чем через минуту на индикаторе появится изображение города после броска, и можно будет оценить, насколько удачен был бросок. Если на индикаторе появится сообщение ЕГГОГ, это означает, что цель достигнута и можно расставлять следующую фигуру.

Пояснения в программе (рис. 19): 00: команда ↑ готовит регистр Х для приема адреса старшего регистра, начиная с которого предстоит формировать изображение города.

01—02: записывают в РО адрес старшего регистра. В нашем случае речь идет о восьмом регистре.

03—12: синтезируют изображение города, причем каждый регистр соответствует одному разряду индикатора.

13: дублирует изображение в регистр У.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	↑	0E	33	$F\sqrt{\quad}$	21	66	ИПС	6C
01	8	08	34	5	05	67	3	03
02	ПО	40	35	—	11	68	+	10
03	Cx	0Г	36	Fx^2	22	69	ПС	4C
04	КИП↑	ГЕ	37	$F\sqrt{\quad}$	21	70	5	05
05	+	10	38	1	01	71	ПП	53
06	1	01	39	+	10	72	80	80
07	0	00	40	ПО	40	73	ИПС	6C
08	X	12	41	КИПО	ГО	74	4	04
09	FLO	5Г	42	ИПА	6—	75	—	11
10	04	04	43	Ftg	1E	76	ПС	4C
11	FBx	0	44	ИПВ	6L	77	ПП	53
12	:	13	45	X	12	78	86	86
13	↑	0E	46	5	05	79	8	08
14	ИП9	69	47	+	10	80	ИПО	60
15	—	11	48	ПС	4C	81	↔	14
16	FIn	17	49	КИПС	ГC	82	—	11
17	↔	14	50	1	01	83	$Fx \geq 0$	59
18	C/П	50	51	ПП	53	84	01	01
19	ПВ	4L	52	80	80	85	ИПС	6C
20	C/П	50	53	ИПС	6C	86	1	01
21	/-/	0L	54	1	01	87	—	11
22	ПА	4—	55	+	10	88	$Fx \geq 0$	59
23	2	02	56	ПС	4C	89	97	97
24	:	13	57	3	03	90	7	07
25	Fx^2	22	58	ПП	53	91	↔	14
26	$F\sqrt{\quad}$	21	59	80	80	92	—	11
27	ИПВ	6L	60	ИПС	6C	93	$Fx \geq 0$	59
28	+	10	61	2	02	94	97	97
29	2	02	62	—	11	95	1	01
30	0	00	63	ПС	4C	96	КПС	LC
31	—	11	64	ПП	53	97	B/O	52
32	Fx^2	22	65	86	86			

Рис. 19

14—16: проверка «игра окончена?».

17—18: если игра не закончена, эти команды выводят на индикатор изображение города.

20—22: ввод в PB и PA хода игрока — L вводится в PB, а угол в PA.

23—41: формирование длины проекции биты L' по формуле $L' = \left[\left| \left| \frac{\alpha}{2} \right| + L - \right. \right.$

$\left. - 20 \right| - 5 \left| \right]$. Прямые скобки в выражении означают операцию вычисления абсолютного значения числа, а квадратные — выделение целой части числа. Формула имитирует влияние вращения биты на длину проекции L' в зависимости от выбранной дистанции L и угла α . 42—49: вычисление положения центра проекции относительно левого края города.

50—98: команды с 50 по 80 и подпрограмма с 80 по 98 проверяют, какие городки задеты битой, и заменяют двойки в соответствующих разрядах на единицы.

Блок-схема алгоритма приведена на рис. 20.

Программу для компьютера можно строить по этому же алгоритму, но игра станет интереснее, если ввести блоки звуковой имитации полета биты и удара. На ФОКАЛе БК-0010 программа будет работать медленно, поэтому лучше писать ее или отдельные ее части непосредственно в машинных кодах, т. е. на языке компьютера.

Потренировавшись, даже самую сложную фигуру можно поразить двумя точными бросками. Однако добиться этого не так уж просто.

«Жонглер и шарик»

На доске размерами 99×99 см лежит теннисный шарик. Задача жонглера — наклоняя доску, удержать шарик и не дать ему скатиться с доски. Игроку мешает порывистый ветер, направление и сила которого все время изменяются. Ветер так и норовит сдуть шарик с доски, и только умение уловить закономерности управления доской

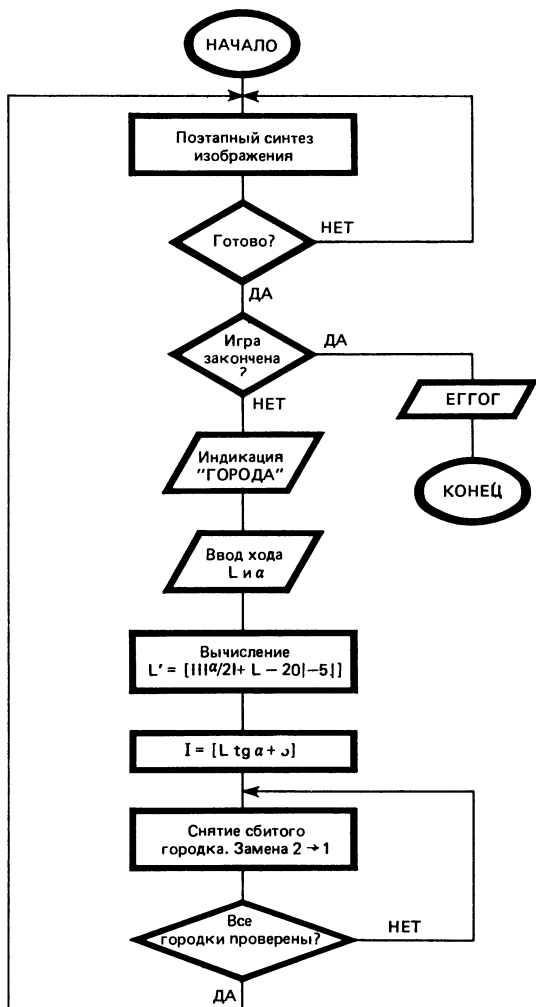


Рис. 20

позволяет жонглеру не ронять шарик.

Наклон доски характеризуется углом α между одним ребром и горизонтальной плоскостью и углом β между другим ребром и той же горизонтальной плоскостью. Углы измеряются в градусах, а направление отсчета их таково, что если за начало отсчета принять левый нижний угол доски, то при положительном значении угла шарик катится в сторону увеличения соответствующей координаты.

Игра имеет девять уровней сложности в зависимости от выбранной силы ветра в баллах от одного до девяти. Задавшись силой ветра ноль баллов,

начинающий может попрактиковаться в катании шарика без всяких помех от ветра. Перемещения шарика можно отмечать на листе бумаги в клеточку. Увеличивая сложность игры, интересно следить за тем, как уменьшается время, в течение которого удастся удерживать шарик на доске.

Прежде чем вводить программу, переключатель Р-Г необходимо поставить в положение Г, поскольку величины углов игроку удобнее вводить в градусах. Чтобы ветер от игры к игре «не повторялся», запишите в регистр Д стартовое время 0,4ЧММ.

Инструкция:

1. Ввести программу (рис. 21) и подготовить ее к работе нажатием клавиш 980 П2 В/О С/П. Через 3 с шарик займет исходную позицию в центре доски — на индикаторе это будет выглядеть как 50.

2. Набрать $\alpha \uparrow \beta$ С/П, т. е. наклонить доску так, чтобы шарик сдвинулся с места, но не скатился с доски.

3. Через 55 с на индикаторе высветится новая позиция шарика: две последние цифры — одна координата, цифры перед ними — другая координата. Если, например, на индикаторе появилось число 6348, это означает, что координаты шарика: (63,48). Число 200 расшифровывается как (2,00), число 74 как (0,74).

4. Сообщение ЕГГОГ означает, что шарик все же упал с доски. Чтобы узнать, где оказался бы шарик, если бы доска была больше, необходимо нажать С/П. На индикаторе появится одна координата, вторая находится в регистре Y. Ее значение нетрудно узнать, нажав клавишу \leftarrow .

Жонглер сможет прочесть на индикаторе количество секунд, в течение которых удалось удерживать шарик на доске, если вновь нажмет клавишу С/П.

Не удалась игра — следует обратиться к началу инструкции, а если жонглер захочет изменить силу ветра, следует нажать клавиши БП 37, войти в программу (F ПРГ), ввести балл и вернуться в режим вычислений (F АВТ).

Говоря о математической модели игры и обсуждая устройство программы, следует отметить, что все формулы, которые будут приведены, относятся лишь

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	5	05	32	—	11	64	КП↑	LE
01	0	00	33	ПД	4Г	65	Fx ≥ 0	59
02	ПА	4—	34	2	02	66	73	73
03	П7	47	35	F1/x	23	67	9	09
04	ИП6	66	36	—	11	68	9	09
05	0	00	37	4	04	69	↔	14
06	ПС	4С	38	:	13	70	—	11
07	ПВ	4L	39	балл	06	71	Fx < 0	5С
08	П9	49	40	X	12	72	74	74
09	П8	48	41	КИПО	ГО	73	П5	45
10	П6	46	42	+	10	74	ИП4	64
11	П5	45	43	КП↑	LE	75	П3	43
12	↔	14	44	ИП3	63	76	FL1	5L
13	С/П	50	45	+	10	77	22	22
14	П4	44	46	Fsin	1С	78	ИП5	65
15	↔	14	47	ИП2	62	79	Fx = 0	5E
16	П3	43	48	X	12	80	89	89
17	1	01	49	П3	43	81	КИП6	Г6
18	3	03	50	КИПО	ГО	82	ИПА	6—
19	ПО	40	51	+	10	83	ВП	ОС
20	2	02	52	КП↑	LE	84	2	02
21	П1	41	53	ИП3	63	85	ИП7	67
22	ИПД	6Г	54	2	02	86	+	10
23	1	01	55	:	13	87	БП	51
24	1	01	56	—	11	88	13	13
25	X	12	57	7	07	89	ВП	ОС
26	Fπ	20	58	F10x	15	90	,	0—
27	+	10	59	+	10	91	ИП7	67
28	ПД	4Г	60	FBx	0	92	ИПА	6—
29	КИПД	ГГ	61	—	11	93	С/П	50
30	↔	14	62	КИПО	ГО	94	БП	51
31	ИПД	6Г	63	+	10	95	00	00

Рис. 21

к одной из составляющих. На самом деле наклон доски, скорость, ускорение, перемещение шарика, воздействие ветра раскладываются на две составляющих и рассчитываются по одинаковым формулам. Чтобы не вводить в память машинки одинаковые формулы дважды, в программе предусмотрен цикл, охватывающий команды с адресами 22—77. Будем приводить формулы только для одной из составляющих.

Структура программы:

22—43: воздействие ветра на шарик предполагается эквивалентным дополнительному наклону доски на углы α' и β' , с которыми суммируются заданные нам углы α и β . Начальное значение $\alpha' = 0$, а каждое следующее определяется по формуле $\alpha'_{i+1} = \alpha'_i + (\xi_i - 0,5) B/4$, где ξ_i , в свою очередь, вычисляется как $\xi_i = \{11 \cdot \xi_i + \pi\}$.

Фигурные скобки в данном случае означают операцию отделения дробной части числа (команды 28—33). B — это сила ветра в баллах. Отметим, что случайным числом определяется приращение ветра, а не его величина, благодаря чему ветер изменяется сравнительно плавно.

44—64: команды вычисления ускорения, скорости, координаты по формулам: $a_{xi} = g \cdot \sin(\alpha_i + \alpha'_i)$; $v_{xi} = v_{xi-1} + a_{xi} t$; $x_i = x_{i-1} + \frac{a_{xi} t^2}{2}$. У нас $t = 1$ с, поэтому формулы упрощаются:

$$v_{xi} = v_{xi-1} + a_{xi}; x_i = x_{xi-1} + v_{xi-1} + \frac{a_{xi}}{2} = x_{i-1} + v_{xi} - \frac{a_{xi}}{2}.$$

Значение $g = 981$ см/с² извлекается из регистра 2, куда оно было записано перед началом игры.

Координаты шарика округляются до целого значения командами с 57 до 61. 65—73: проверка «не вышел ли шарик за пределы доски?». Если шарик скатился с доски, то в регистр 5 записывается число, отличное от нуля, или, как говорят программисты, устанавливается флаг.

74—77: эти команды завершают цикл,

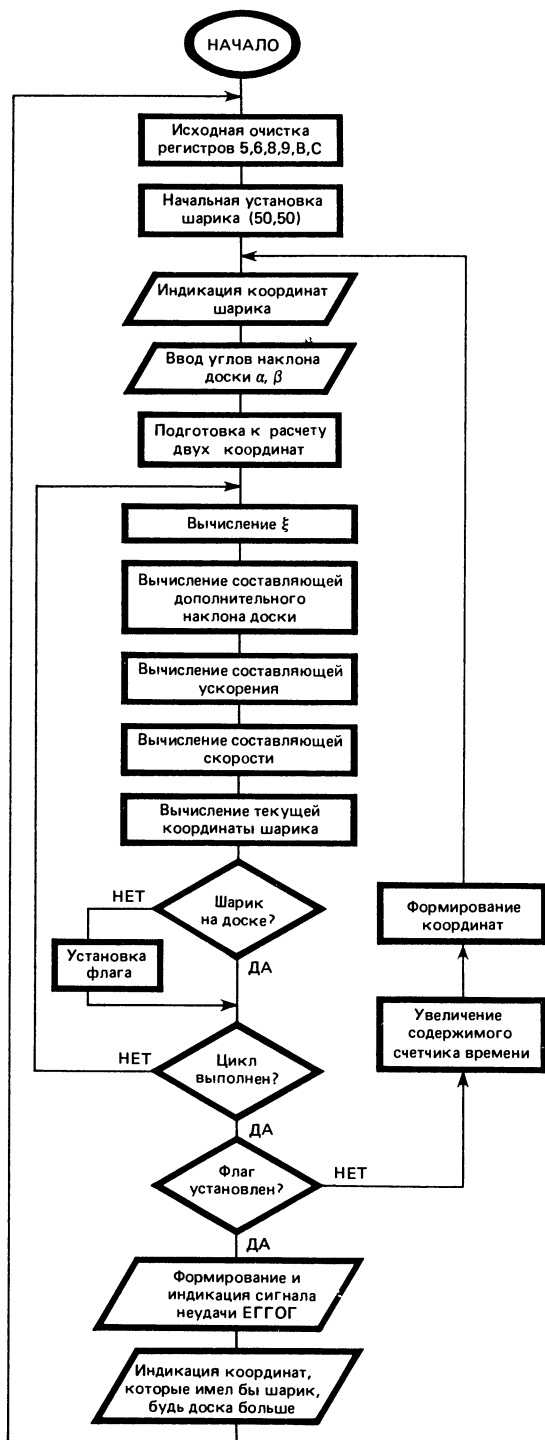


Рис. 22

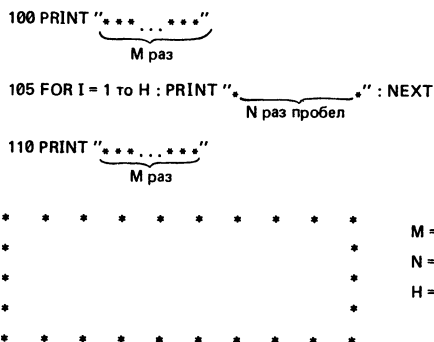


Рис. 23

и, если рассчитаны обе координаты, управление передается на следующую группу команд.

78—80: проверка «был ли установлен флаг?».

81—88: в тех случаях, когда флаг установлен не был, к счетчику времени добавляется единица и индицируются координаты шарика.

85—95 и 00—13: если флаг был установлен, то калькулятор останавливается и выдает на индикатор сообщение ЕГГОГ. После запуска клавишей С/П игрок узнает координаты шарика, а затем регистры приводятся в исходное состояние и индицируется время, в течение которого шарик находился на доске.

По блок-схеме алгоритма (рис. 22) нетрудно написать программу для компьютера. Вычисление составляющих удобно оформить в виде подпрограммы. Чтобы играть было интереснее, на экран дисплея или телевизора, подключенного к микро-ЭВМ, можно вывести изображение доски с шариком. Изображение доски формируется, например, так, как на рис. 23.

Положение шарика будет зависеть от вычисленных координат. Предположим, блок вычислений разместился с 200 по 400 адрес. Тогда команда 410 PRINT AT X, Y; «0» выведет изображение шарика в нужную позицию доски. X и Y — координаты шарика на доске, имеющей размеры $M \times N$. M и N определяются графическими возможностями, которые предоставляет компьютер.

Перед тем как вычислять новые координаты шарика, нужно стереть его

прежнее изображение. Это сделает команда

610 PRINT AT X,Y; « »

Программа «Жонглер» может иметь варианты, поиск которых — увлекательное занятие для начинающих программистов.

БОЛЬШИЕ ГОНКИ

В гонках, как ни в каких других играх, проявляется способность игрока приспосабливаться к постоянно изменяющейся ситуации. Но не только этим интересны игры, собранные в данной главе. Освоив их, читатель в подобных ситуациях научится выбирать оптимальную стратегию: иногда необходимо выиграть в скорости, а иногда важнее позиционный успех.

Гонки рассчитаны на многих участников, поэтому для игры можно пригласить друзей, располагающих любыми моделями программируемых микрокалькуляторов. Впрочем, можно обойтись и одним калькулятором, играя по очереди.

Немного фантазии, и можно перенести гонки в космос или глубины океана — стоит только составить собственные программы для микрокалькулятора или микро-ЭВМ.

«Поездка по городу»

Для этой игры удобно воспользоваться планом вашего родного города, однако подойдет и тетрадный лист в клетку. В таком случае клетки заменят городские кварталы, а разделяющие их линии станут улицами.

На плане отмечаем два традиционных пункта: А и Б. Итак, из пункта А нам предстоит доехать до пункта Б, неукоснительно соблюдая предписания и запрещения дорожных знаков и руководствуясь сигналами светофора.

В путь мы отправимся на машине, очень похожей на настоящую: она может двигаться вперед, поворачивать направо или налево, притормаживать. Управлять такой машиной совсем легко — от перекрестка до перекрестка она движется автоматически, если только водитель соблюдает правила движения.

Пульт управления — клавиатура микрокалькулятора. Разобраться в пульте поможет рис. 24.

Какой именно знак управляет движением на данном перекрестке, сообщит индикатор. Знак может быть любым из тех десяти знаков, что приведены на рис. 25. Поскольку на индикаторе высвечивается только цифровая информация, то каждому знаку поставлено в соответствие некоторое число. Если на перекрестке нас встретит красный сигнал светофора: «121», то следует остановиться; 0 С/П «11» — загорелся желтый свет, подождем еще 0 С/П «0» — путь свободен.

Если мы нарушим правила движения, например, не заметив знака «7», предписывающего движение прямо, повернем направо: 6 С/П, то сразу же нам преградит путь «милицейский жезл», указывающий на нарушенный знак: «7,1111111». После этого нам останется лишь проехать в разрешенном знаком направлении, однако к содержимому регистра 5 — «счетчику нарушений» — добавляется единица.

Конечно, наш путь к намеченной цели по городским улицам будет извилист, но это неизбежно.

Цель игры — как можно быстрее добраться до пункта Б, не совершив нарушений. Тем, кто хочет проверить свою память, стоит попробовать не пользоваться во время игры рисунком с расшифровкой, где указано, какой цифре какой знак соответствует.

Инструкция:

1. Наберите число 0,4ЧММ, нажмите В/О С/П «0».

2. Ноль на индикаторе говорит, что путь свободен, мы в исходном пункте А, на светофоре — зеленый сигнал. Набираем цифру — код направления движения, С/П. Проезжаем квартал до следующего перекрестка, и на индикаторе появляется новое число — код дорожного знака или светофора.

3. После прибытия в конечный пункт Б загляните в Р4 и Р5: ИП4 — «длина маршрута или количество кварталов, мимо которых пролегал маршрут», ИП5 — «число допущенных нарушений».

Чтобы проверить точность набора программы и освоить управление машиной, необходим контрольный пример.

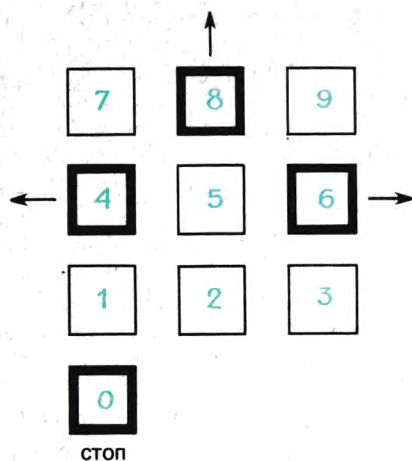


Рис. 24



Рис. 25

На плане города отмечаем произвольные пункты А и Б, набираем 0,1612 В/О С/П «0». В путь! 8 С/П

«2» — поворот налево запрещен, значит, снова едем прямо: 8 С/П «121». Светофор. Остановимся: 0 С/П «11» 0 С/П «0», путь свободен, повернем налево: 4 С/П «3». Знак «поворот направо запрещен», тогда вперед: 8 С/П «5». Еще вперед: 8 С/П «5,111111». Что такое? «5» — означает «движение разрешено только налево». Мы нарушили правила! Исправимся: 4 С/П «5». Придется сделать круг: 4 С/П «9» 4 С/П «8». Теперь полный вперед: 8 С/П «1» 4 С/П «1,11111111»! Почему? Мы не притормозили у знака «Стоп», 0 С/П «0» — путь свободен, 4 С/П «2» 6 С/П «9» 8 С/П «9». Приближаемся к цели 4 С/П «121». Светофор, но мы уже на месте! Из регистров Р4 и Р5 узнаем, что проехали двенадцать кварталов и допустили при этом два нарушения.

Структура программы (рис. 26):

00: после того как игрок ввел время в регистр Х и запустил программу на счет, эта команда копирует случайное число в регистр Д.

01—03: число 11 используется при каждом обращении к датчику псевдослучайных чисел 09—19. Чтобы не тратить время на многократное получение этого числа, его записывают в Р1.

04—06: для подсчета пройденного пути служит регистр Р4. За каждое нарушение электронный арбитр станет добавлять единицу к содержимому регистра 5 — счетчика нарушений. Пока что автомобиль стоит на старте, поэтому в Р4 и Р5 записывается ноль.

07: предыдущие команды очистили не только Р4 и Р5, но и РХ, поэтому команда 07 выводит на индикатор 0 — сигнал, означающий, что путь свободен. 08: к содержимому регистра 4 — счетчика пройденного пути — добавляется единица.

09—19: вычисление очередного случайного числа по формуле $D = \{11 \cdot D + \pi\}$, где фигурные скобки означают отделение дробной части числа.

20—21: определение дорожного знака на основе целой части случайного числа. Если результат определения знака — число 11 или 12, то на перекрестке встретится не знак, а светофор.

22—26: анализ вычисленного числа, которое может быть в диапазоне от единицы до двенадцати. В зависимости

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ПД	4Г	30	ИПО	60	60	БП	51
01	1	01	31	1	01	61	70	70
02	1	01	32	—	11	62	5	05
03	П1	41	33	$F_x = 0$	5Е	63	+	10
04	Сх	ОГ	34	40	40	64	2	02
05	П4	44	35	+	10	65	Х	12
06	П5	45	36	$F_x = 0$	5Е	66	ИПС	6С
07	С/П	50	37	70	70	67	—	11
08	КИП4	Г4	38	БП	51	68	$F_x \neq 0$	57
09	ИПД	6Г	39	07	07	69	08	08
1С	ИП1	61	40	ИПО	60	70	КИП5	Г5
11	Х	12	41	5	05	71	ИПО	60
12	Фπ	20	42	—	11	72	9	09
13	+	10	43	$F_x < 0$	5С	73	F1/x	23
14	ПО	40	44	48	48	74	+	10
15	КИП†	ГЕ	45	ИПО	60	75	БП	51
16	≥	14	46	БП	51	76	28	28
17	ИПО	60	47	54	54	77	КИП5	Г5
18	—	11	48	3	03	78	ИП1	61
19	ПД	4Г	49	—	11	79	F_x^2	22
20	КИПО	ГО	50	$F_x \geq 0$	59	80	С/П	50
21	КИПО	ГО	51	62	62	81	$F_x = 0$	5Е
22	ИПО	60	52	2	02	82	77	77
23	ИП1	61	53	+	10	83	ИП1	61
24	—	11	54	2	02	84	С/П	50
25	$F_x < 0$	5С	55	Х	12	85	$F_x \neq 0$	57
26	78	78	56	ИПС	6С	86	07	07
27	ИПО	60	57	—	11	87	КИП5	Г5
28	С/П	50	58	$F_x = 0$	5Е	88	БП	51
29	ПС	4С	59	08	08	89	83	83

Рис. 26

от его величины управление переходит на блок «Светофор», либо на блок «Индикация знака».

27—28: индикация знака. Команда 27 выводит знак в регистр Х, а команда 28 прерывает выполнение программы для того, чтобы игрок мог увидеть знак и ввести код движения.

29—30: ввод хода. Код движения запоминается в РС.

31—34: анализ дорожного знака, который управляет движением на данном перекрестке.

35—39: на этот блок управление переходит во всех случаях, когда на перекрестке установлен знак «Стоп». Напротив, когда знак «Стоп» на перекрестке отсутствует, будет выполняться блок «Проверка выполнения указаний знаков». Наконец, в том случае, если, несмотря на знак «Стоп», машина не остановлена, начнет выполняться блок «Сообщение об ошибке». Остался лишь один блок, куда можно передать управление — это «Путь свободен». К нему мы отправимся только в том случае, когда предписание знака будет выполнено.

40—44: проверяется, идет ли речь о запрещающем или предписывающем знаке. В любом случае управление переходит на блок проверки корректности хода. Однако в случае если знак оказался запрещающим, в работу включатся команды 45—47, 54—61, а если предписывающим, то 48—69.

45—69: проверка корректности хода. В зависимости от результата проверки отправляемся на блок «Сообщение об ошибке», когда правила нарушены, и на «Ввод хода», если водитель следовал указаниям знаков.

70—76: блок «Сообщение об ошибке». Команда 70 начисляет штрафное очко, а команды с 71 по 74 нужны для того, чтобы подготовить изображение милицейского жезла, «указывающего» на нарушенный знак. Оставшиеся команды 75—76 передают нарушителя на блок «Индикация знака».

77—85: последний блок команд — это «Светофор». Подобно «проверке выполнения знака», этот блок состоит из нескольких команд. Команды с 78 по 80 выводят на индикатор «красный сигнал» светофора — «121». Фрагмент 81, 82

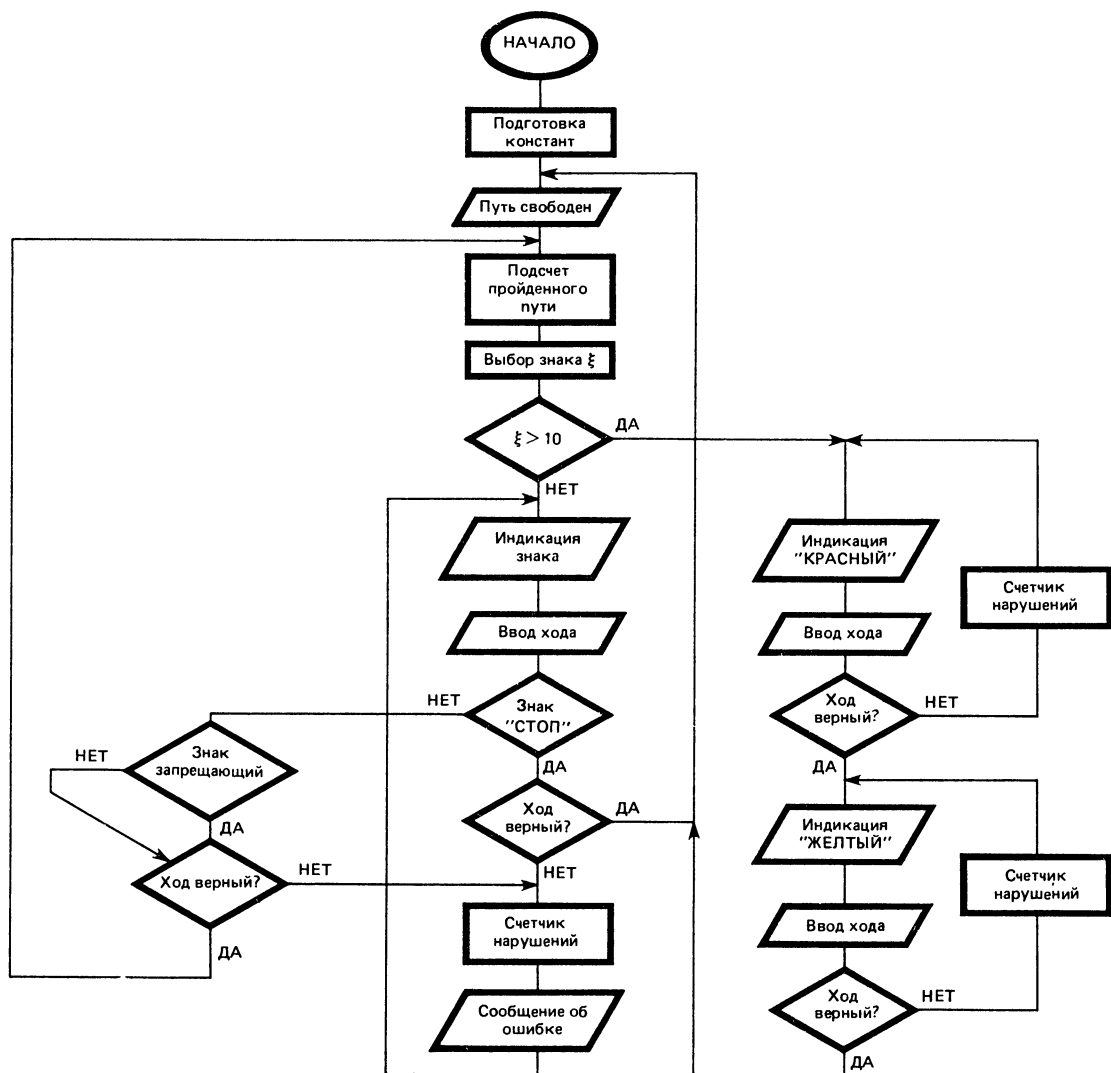


Рис. 27

анализирует, правильно ли отреагировал водитель на сигнал светофора, и если нарушение не зафиксировано, то управление передается командам 83—84, которые выводят на индикатор «желтый сигнал» светофора — «12». В противном случае содержимое счетчика нарушений увеличивается на единицу командой 77. Назначение команд 85, 86 такое же, как у команд 81, 82, однако на этот раз нарушение будет зафиксировано командами 87—89. Если же предписание светофора будет выполнено верно, управление передается на команду 07, и машина продолжает движение.

Блок-схема игры приведена на рис. 27. Особого внимания заслуживает фрагмент, изображенный слева в нижней части блок-схемы. Непривычный для программистов на языках высокого уровня — Бейсике, Фортране, ПЛ-1, этот блок нередко можно встретить в программах для микрокалькулятора, где одни и те же команды могут выполнять различные функции в зависимости от того, откуда передано управление.

Программа для микро-ЭВМ должна выдавать на экран не только знак и число нарушений, но и изображение карты пройденного маршрута, длины

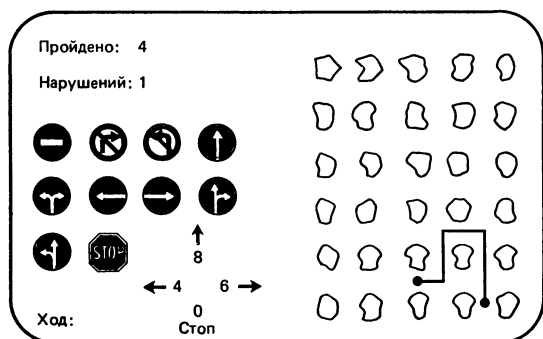


Рис. 28

пройденного пути. Картинка может быть, например, такой, как показано на рис. 28.

«Гонки на зимнем озере»

В игре принимают участие несколько человек, соревнуясь друг с другом. У каждого — микрокалькулятор, который по программе моделирует движение аэросаней по ледяной дорожке, расчищенной в снегу на поверхности замерзшего озера. Справа дует порывистый ветер. Особенно сильные его порывы сносят машину к левой обочине. Компенсировать снос можно, поворачивая руль от -90° до $+90^\circ$. Во время игры переключатель Р-Г должен находиться в положении Г.

Ширина дорожки равна шести единицам, и в момент старта сани установлены на осевой линии. О длине дорожки играющие договариваются заранее. Чаще всего для соревнования выбирают дистанцию от 20 до 50 единиц. Условленную величину следует набрать с клавиатуры каждого микрокалькулятора перед началом игры. После нажатия клавиш В/О, С/П калькулятор, закончив все подготовительные операции, останавливается примерно через 4 с. Тройка на индикаторе показывает расстояние от саней до левой обочины. Зная ширину дорожки, легко определить расстояние до правой обочины. Скорость саней узнать также нетрудно, стоит лишь нажать клавишу \rightleftarrows . Калькулятор готов принять информацию для очередного хода.

Гонщик вводит приращение скорости саней Δv и угол поворота руля φ (как правило, положительный, поскольку ветер дует справа): $\Delta v \uparrow \varphi$. Величина Δv может равняться либо единице, либо нулю, либо минус единице — таково условие игры.

Задача каждого гонщика — достичь финиша за наименьшее время, в нашем случае — за наименьшее число ходов, заработав при этом минимальное число штрафных очков. Случаев, когда начисляется штрафное очко, всего три.

Во-первых, одно штрафное очко гонщик получает, если из-за его ошибки или нерасторопности скорость саней упала до нуля.

Во-вторых, гонщика наказывают также в том случае, если абсолютная величина приращения скорости превышает единицу (в обоих случаях после начисления штрафа надо ввести приращение заново).

В-третьих, управляемые неопытным гонщиком сани могут вылететь с трассы: пересечь левую или правую обочину, врезаться в снег (тогда скорость саней гасится до нуля, и они начинают разбег с той же точки).

После начисления штрафного очка, о чем сигнализирует сообщение ЕГГОГ на индикаторе, движение саней начинается с осевой линии трассы. К счетчику времени прибавляется единица. Увидев сигнал аварийного останова, следует нажать клавишу С/П, чтобы узнать причину штрафа. Если после этого на индикаторе загорается число л, значит, слишком резко возросла скорость саней, если нуль — скорость упала до нуля, если число меньше нуля или больше шести — сани заехали на обочину. Еще раз нажимаем клавишу С/П, и игра продолжается. А вот если во время останова на индикаторе высвечивается длина дистанции, значит сани достигли финиша.

Сколько времени t ушло на прохождение дистанции, можно узнать, нажав клавиши ИП6.

Счет полученным штрафным очкам ведется в регистре Р5. Нажимаем ИП5, и на индикаторе высвечивается их суммарное число.

Зная распределение остальных регистров, нетрудно контролировать все па-

Адрес	Команд	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	П1	41	30	П9	49	60	П7	47
01	3	03	31	П0	40	61	2	02
02	П8	48	32	ИП1	61	62	:	13
03	Сх	0Г	33	ИП2	62	63	—	11
04	П2	42	34	ИП3	63	64	П8	48
05	П5	45	35	F cos	1Г	65	ПД	4Г
06	П6	46	36	+	10	66	$F_x \geq 0$	59
07	П7	47	37	П2	42	67	77	77
08	П9	49	38	—	11	68	6	06
09	КИП6	Г6	39	$F_x \geq 0$	59	69	ИП8	68
10	ИП9	69	40	84	84	70	—	11
11	ИП8	68	41	ИП3	63	71	$F_x \geq 0$	59
12	С/П	50	42	F sin	1С	72	77	77
13	П3	43	43	ИП8	68	73	FL0	5Г
14	\rightarrow	14	44	+	10	74	32	32
15	П4	44	45	ИП9	69	75	БП	51
16	F π	20	46	9	09	76	09	09
17	ПД	4Г	47	:	13	77	Сх	0Г
18	1	01	48	—	11	78	П9	49
19	ИП4	64	49	ИП7	67	79	3	03
20	F_x^2	22	50	1	01	80	П8	48
21	—	11	51	1	01	81	КИП5	Г5
22	$F_x \geq 0$	59	52	X	12	82	ИПД	6Г
23	81	81	53	F π	20	83	K—	27
24	ИП4	64	54	+	10	84	ИП1	61
25	ИП9	69	55	П7	47	85	С/П	50
26	+	10	56	КИП7	Г7	86	БП	51
27	ПД	4Г	57	FO	25	87	09	09
28	$F_x \neq 0$	57	58	ИП7	67			
29	81	81	59	—	11			

Рис. 29

раметры непосредственно в ходе игры. Р1 — длина дистанции L , Р2 — пройденный путь S , Р3 — угол поворота руля φ , Р4 — приращение скорости Δv , Р7 — случайное число ξ , используемое для вычисления порывов ветра, Р8 — расстояние d от саней до левой обочины, Р9 — скорость саней v , Р0 — счетчик.

Перед расчетом каждого очередного хода в регистр 0 заносится значение скорости v . Вычисления происходят циклически, поэтому чем выше скорость, тем дольше приходится ожидать результатов. С каждым из v прохождений цикла к d прибавляется $\sin \varphi = (v/9 - (\xi/2))$, к S прибавляется $\cos \varphi$, к t — единица, а также проверяется, не оказались ли сани на обочине. Поскольку их снос зависит от случайных порывов ветра, то в особенно неудачном положении оказывается лихач, задающий саням слишком большую скорость.

В программе игры сигнал ЕГГОГ вызывается командой К—, не описанной в руководстве по эксплуатации микрокалькулятора. Останов по ней происходит с пропуском одной команды: пропуск используется, чтобы по достижении фи-

ниша на индикаторе появилась длина дистанции — сигнал об окончании гонок.

Программа игры показана на рис. 29. Краткий комментарий поможет разобраться в назначении отдельных групп команд:

00—08: команды засылки введенного числа L в Р1, тройки в Р8 и очистка, т. е. ввод нуля в Р2, Р5, Р6, Р7, Р9.

09—12: добавление единицы к счетчику времени, в регистры Y и X вызываются величины v и d , чтобы их можно было увидеть на индикаторе. Команда С/П — останов, во время которого происходит ввод $\Delta v \uparrow \varphi$, а затем пуск калькулятора. 13—15: рассылка величин Δv и φ по своим местам.

16—23: проверка, соблюдается ли условие « $|\Delta v| \leq 1$ ».

24—29: вычисление и проверка « $v=0$ ». 30—31: запись величины v в регистры Р9 и Р0.

32—40: расчет продвижения саней вдоль дистанции $v \cos \varphi$ и проверка « $S-L \leq 0$ ».

41—44: расчет бокового сноса саней, вызванного отклонением от курса.

45—48: расчет постоянной составляющей ветрового сноса $v^2/9$.

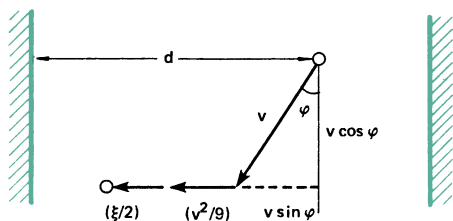


Рис. 30

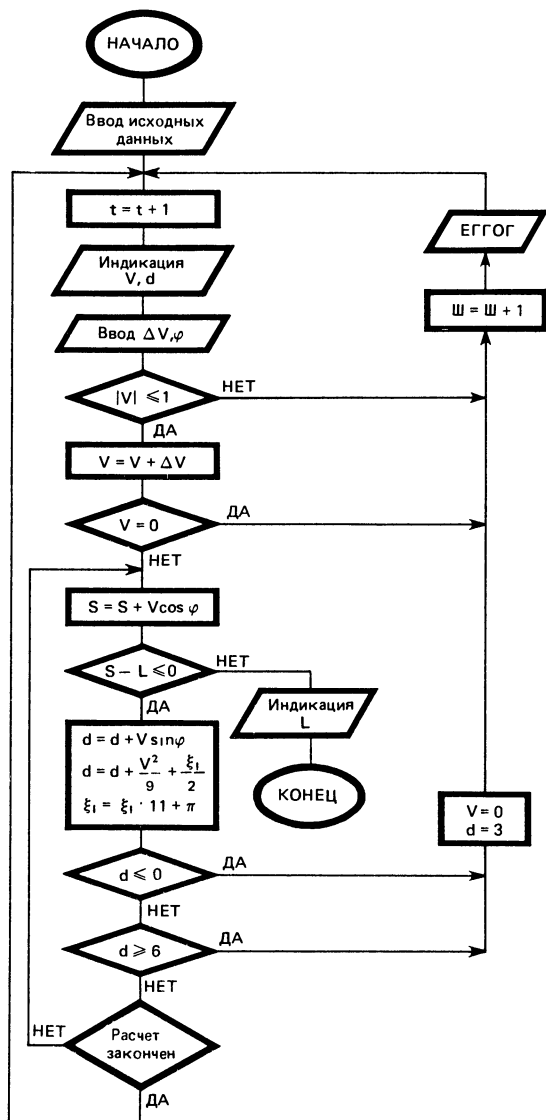


Рис. 31

49—65: расчет случайной составляющей ветрового сноса (рис. 30), при которой к ней за каждое i -е прохождение цикла прибавляется величина $(\xi_i/2)$, рассчитываемая согласно формуле $\xi_i = \{11 \cdot \xi_{i-1} + \pi\}$, где фигурные скобки означают операцию отделения дробной части заключенного в них числа.

66—72: проверка, соблюдаются ли условия « $d \leq 0$ », « $d \geq 6$ ».

73—74: возврат к началу цикла.

75—76: возврат к началу расчета очередного хода.

77—80: уменьшение до нуля скорости саней, заехавших на обочину, возврат машины на середину дорожки.

81—87: блок начисления штрафных очков.

Блок-схема программы приведена на рис. 31.

Приводим ради примера протокол одной из гонок (табл. 5): длина дистанции — 20, штрафное очко начислено на седьмом ходу из-за пересечения санями левой обочины ($d = -0,01$).

Таблица 5

d	v	Δv	φ°
3,0	0	1	0
2,8	1	1	3
1,7	2	1	45
2,0	3	1	20
0,5	4	1	70
1,1	5	1	80
1,0	8	1	85
0,7	7	1	87
ЕГГОГ —0,01	т. е.	$d < 0$	С/П С/П
3,0	0	1	0
2,8	1	1	5
1,9	2	1	10
0,5	3	1	15

На индикаторе: 20 (финиш)
ИП6: $t = 12$ ИП5: $n = 1$

«Регата»*

Для этой игры в калькулятор будет введена математическая модель небольшой яхты с бермудским парусным вооружением — одной мачтой и косым тре-

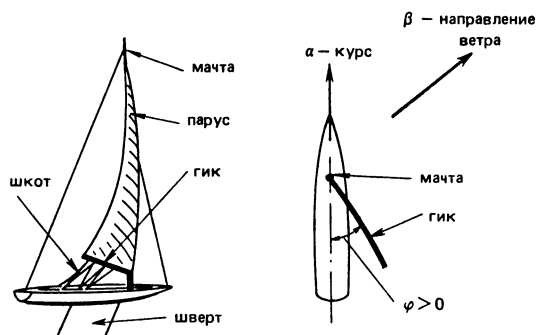


Рис. 32

Рис. 33

угольным парусом (рис. 32). Нижней кромкой парус крепится к гику, который подвижно скреплен с мачтой и растягивает нижнюю часть паруса. К гику, в свою очередь, крепится шкот, необходимый для управления яхтой. Чтобы яхту не сносило в сторону, есть и плоский киль, или, говоря на морском языке, шверт.

Коль скоро яхту не сносит в сторону, то и направление скорости яхты совпадает с α — курсом судна. Величина скорости определяется из уравнения $\frac{dv}{dt} = F_v - kv$, где $F_v = F \sin \varphi \cdot \sin(\alpha + \varphi - \beta)$ зависит от силы ветра и углов: угла курса — α , направления ветра — β и φ — угла между парусом и осью судна. Функция F_v имеет положительное значение, если парус отклоняется на правый борт, отрицательное — если на левый (рис. 33).

Поскольку все величины, кроме скорости v чаще всего постоянны, уравнение $\frac{dv}{dt} = F_v - kv$ легко разрешается. Сложнее дело обстоит при повороте яхты. По правилам игры считается, что во время поворота координаты X и Y не меняются, а на поворот затрачивается некоторое время τ . Величина τ зависит от угла поворота δ и скорости судна: $\tau = \frac{\delta}{v \cdot c}$.

Скорость судна, выполнившего поворот, определяется формулой $v = v_0 \cos \delta$, где v_0 — скорость до поворота.

Перед тем как вводить программу, установите в положение Г переключатель Р-Г. На листе миллиметровки нужно нарисовать озеро или залив побольше, например, 200×200 мм, причем берега не должны быть прямолинейными. Опреде-

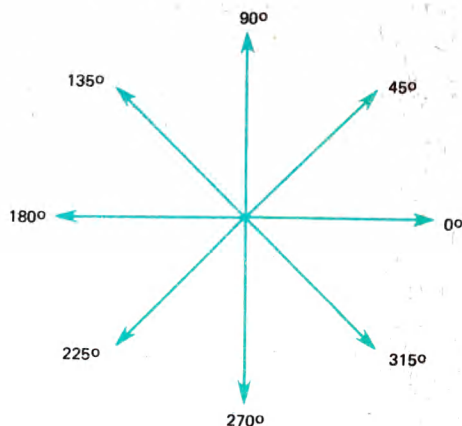


Рис. 34

лив направление ветра по рис. 34 (пусть для начала ветер будет попутным), занесите в РА. Начальные координаты X , Y записываются в Р1 и Р2. За единицу масштаба удобно принять 1 мм.

Теперь можно ввести программу. Проверив ее правильность по кодам или по тесту, разверните яхту (введите начальное значение курса α в Р4), отпустите шкот (угол φ в Р5) и оттолкнитесь от берега, задав небольшую положительную скорость ($v=0,1$ в Р3). Проверьте еще раз по табл. 6, все ли начальные координаты заданы верно, и нажимайте клавиши В/О, С/П.

Таблица 6

Регистр	Величина, обозначение	Начальное значение
0	Время, t	0
1	x } координаты	0
2		0
3	Скорость, v	0,1
4	Курс яхты, α	30
5	Угол отклонения паруса, φ	10
A	Направление ветра, β	80
B	Сила ветра, F	2
C	Коэффициент сопротивления, K	1
D	Коэффициент, C	80

На индикаторе засветился курс — угол α . Теперь можно нажать С/П, тогда курс останется прежним, или ввести новое значение курса α С/П, и яхта повернет. Не стоит поворачивать сразу больше чем на 90° , к тому же каждый поворот требует времени и снижает скорость.

Новое число на индикаторе — это угол ориентации паруса относительно оси судна φ . Если $\varphi > 0$, то парус отклонен в сторону правого борта, в противном случае — в сторону левого. То, в какую сторону отклонен парус, «решает» не игрок, а ветер. Игрок задает лишь «длину» шкота, численно равную величине угла φ , а уж знак этой величины выбирает микрокалькулятор. Ему можно помочь, задавая нужный знак, однако программа в любом случае дополнительно проверит правильность выбора.

Следует помнить, что при определенных условиях парус начинает полоскаться на ветру. В этом случае нужно либо изменить немного курс, либо укоротить шкот. Итак, приняв решение о величине угла φ , вводите ее и нажимайте С/П. Как и в случае с α , если φ остается прежним, можно просто нажать С/П.

При следующей остановке калькулятора на экране высвечивается время, прошедшее с момента отплытия. Необходимо задать величину следующего промежутка времени, в течение которого курс меняться не будет. По условиям игры максимальный интервал времени равен 230. Нажав опять клавишу С/П, через некоторое время увидим текущие координаты яхты, точнее, координату X . Координату Y нетрудно определить, нажав клавишу \leftrightarrow .

Отметьте на карте новые координаты яхты и решайте, куда плыть дальше. Когда решение будет принято, снова нажимайте С/П. На индикаторе вновь высветится значение курса — угол α , и можно продолжать плавание.

Прежде чем участвовать в гонках, сначала «выйдите в море» в одиночку, сделайте несколько поворотов неподалеку от берега, научитесь управлять яхтой, возвращаться к пристани, причем возвращаться не на слишком большой скорости. Ну, а овладев этим искусством, можно отправляться в «большое» плавание. Это лучше сделать с товарищем, у которого тоже есть калькулятор. Устройте гонки — регату: кто скорее пройдет дистанцию от пункта А до пункта Б, разделенных островами и проливами. Секундомер есть у каждого — время, прошедшее со старта, суммируется в РО.

Победить в регате будет достаточно сложно, даже если ветер на протяжении

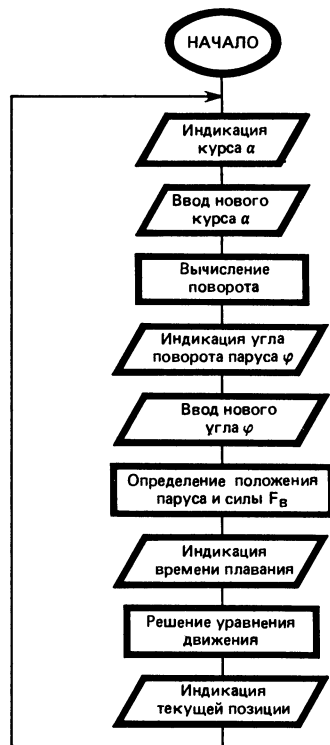


Рис. 35

всех гонок будет оставаться постоянным по силе и направлению, однако чтобы дополнительно усложнить задачу, опытные яхтсмены могут договориться об изменениях в силе и направлении ветра: в такой-то момент ветер станет таким-то.

Прежде чем разбираться в структуре программы, попробуйте сыграть в игру. Чтобы проверить, не ошиблись ли вы при вводе программы в калькулятор, воспользуйтесь контрольным примером.

Контрольный пример. Введем в память калькулятора начальные условия (распределение регистров памяти — в табл. 4): $t=0$, $X=0$, $Y=0$, $\beta=80^\circ$, $F=2$, $K=1$, $C=80$, $v=0,1$, $\alpha=30^\circ$, $\varphi=10^\circ$.

Старт: В/О С/П « $\alpha=30^\circ$ ». Не будем менять курс — С/П « $\varphi=10^\circ$ » С/П « $t=0$ ». Будем выдерживать заданный курс две единицы времени: 2 С/П « $X=0,37060552$ » \leftrightarrow « $Y=0,2139692$ » — координаты яхты С/П « $\alpha=30^\circ$ » С/П « $\varphi=-10^\circ$ », оказывается $+10$ было задано неверно, но калькулятор заметил ошибку и автоматически исправил ее. Увеличим длину шкота: 20 С/П « $t=2$ » — прошло две

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИП4	64	31	ИПО	60	62	ИП2	62
01	С/П	50	32	С/П	50	63	+	10
02	ИП4	64	33	П8	48	64	П2	42
03	↗	14	34	ИПО	60	65	ИП9	69
04	П4	44	35	+	10	66	ИП4	64
05	—	11	36	ПО	40	67	Fcos	1Г
06	Fcos	1Г	37	ИП9	69	68	X	12
07	П8	48	38	ИП8	68	69	ИП1	61
08	Farccos	1—	39	X	12	70	+	10
09	ИП3	63	40	FBx	0	71	П1	41
10	:	13	41	ИПС	6С	72	С/П	50
11	ИПД	6Г	42	X	12	73	БП	51
12	:	13	43	/—/	0L	74	00	00
13	ИПО	60	44	FeX	16	75	Fsin	1C
14	+	10	45	ИП3	63	76	ИП5	65
15	ПО	40	46	ИП9	69	77	ИП4	64
16	ИП8	68	47	—	11	78	+	10
17	ИП3	63	48	X	12	79	ИПА	6—
18	X	12	49	ИП9	69	80	—	11
19	П3	43	50	+	10	81	Fsin	1C
20	ИП5	65	51	ИП3	63	82	X	12
21	С/П	50	52	↗	14	83	ИПВ	6L
22	П5	45	53	П3	43	84	X	12
23	ПП	53	54	—	11	85	Fx < 0	5C
24	75	75	55	ИПС	6С	86	28	28
25	ПП	53	56	:	13	87	ИП5	65
26	75	75	57	+	10	88	/—/	0L
27	Сx	0Г	58	П9	49	89	П5	45
28	ИПС	6С	59	ИП4	64	90	В/О	52
29	:	13	60	Fsin	1C			
30	П9	49	61	X	12			

Рис. 36

единицы времени, а теперь не будем менять курс еще десять единиц времени: 10 С/П «X=5,6175955» ↗ «Y=3,2433203» — новые координаты яхты С/П «a=30». На этот раз изменим курс: 0 С/П «φ=—20». Положение паруса оставим прежним С/П «t=12,583412». Времени прошло немного больше 12 единиц, так как некоторое время было потрачено на поворот. 10 С/П «X=12,23709» ↗ «Y=—3,2433203» — координаты яхты.

Алгоритм игры изображен на рис. 35. Линейная структура алгоритма обеспечивает сравнительно высокое быстродействие. Еще интереснее реализовать эту программу на микро-ЭВМ. Алгоритм настолько несложен, что на этот раз не понадобятся никакие комментарии. Опытный программист сможет создать на экране компьютера изображение акватории, островов, яхт и пристаней.

Структура программы (рис. 36):

00—01: индикация курса яхты.
02—19: вычисление поворота.
22—31: определение положения паруса и силы F_B .

32: индикация времени, прошедшего с отплытия.

33—71: интегрируется уравнение движения.

72—74: индикация текущей позиции яхты и переход к началу программы.

75—80: подпрограмма вычисления силы F_B .

СОВЕРШЕННЫЙ СТРАТЕГ

Исход многих игр, с которыми читатель знакомился в предыдущих главах, зависел от случайностей, поэтому даже опытный, дальновидный игрок может иногда проиграть новичку. В гонках, например, важнее было выбрать верную тактику движения, нежели пытаться предвидеть повороты игры на много ходов вперед.

В этой главе — игры, для успеха в которых важно не только вести умелые тактические действия, но и тщательно продумывать стратегию. Такие игры помогут развить наблюдательность, память и, конечно, логическое мышление.

Новичку, играющему в «Охоту на лис», может показаться, что добиться успеха невозможно, однако мастера справляются с пятью-шестью «лисами» всего за пятнадцать ходов.

То же можно сказать и об охоте на жирафа из игры «Африканское сафари». Только наблюдательный охотник, склонный к логическому анализу, сможет быстро разобраться в «повадках» электронного жирафа и научится ловить его, почти не делая лишних ходов.

«Три шашки» — наиболее простая из игр этой главы, однако и она доставит немало трудностей тому, кто пытается добиться победы, не разработав сначала выигрышного алгоритма.

«Три шашки»

Игровое поле отображается на индикаторе. Калькулятор оперирует числами, имеющими восьмиразрядную мантиссу, а следовательно, и поле состоит из восьми клеток. На поле размещаются три шашки. Они изображаются единицами в соответствующих разрядах. Нули в остальных разрядах показывают незанятые позиции. Следует учитывать, что нули, стоящие слева от старшей единицы, являются незначащими и на индикатор не выводятся. Соответственно этому «10110» на индикаторе следует понимать как 00010110. Клетки игрового поля нумеруются слева направо цифрами от 1 до 8.

Цель игры заключается в том, чтобы, передвигая по определенным правилам шашки в строгой очередности с микрокалькулятором, сделать последний, победный ход. Разрешается при этом ставить одну шашку на другую или даже собирать три шашки на одной клетке. В итоге все шашки должны собраться на последней, восьмой клетке игрового поля. На индикаторе в этом случае появится как бы «сумма» всех изображений — число 3.

Правила игры чрезвычайно просты. Ход игрока, идет ли речь о человеке или о микрокалькуляторе, должен подчиняться следующим правилам: любую шашку можно сместить вправо на любое, неравное нулю число клеток.

Калькулятор, разумеется, не станет нарушать правила, он даже не анализи-

рует корректность ходов своего соперника. Таким образом, следует внимательно контролировать свои ходы, чтобы игра была честной и по-настоящему интересной.

Игра начинается с ввода программы и исходных данных. Предположим, программа уже записана в микрокалькулятор. Введем 3 в P0, P1, 5 в P5, 9 в P2 и 44 в PД. Электронный партнер готов к игре.

Введите исходную позицию: «10101000». Для этого укажите номера клеток, занятых шашками, разделяя их командой ↑. Нумерация идет слева направо, значит, вводим 1 ↑ 3 ↑ 5 С/П. Через 15 с на индикаторе высвечивается заданная позиция — «10101000».

Первый ход за микрокалькулятором: нажмиме С/П. Теперь придется подождать чуть больше — 25 с. Ситуация изменилась — на индикаторе «10100100». Ход за игроком.

Ввод осуществляется так же, как и в первый раз, например, 1 ↑ 4 ↑ 6 С/П. Пока индикатор затемнен, напомним, что цель игры заключается в том, чтобы своим последним ходом получить на индикаторе число 3. Новое число — «10010100», нажимаем С/П и ожидаем ответа микрокалькулятора.

«1010100»! Это напоминает исходную ситуацию, 2 ↑ 5 ↑ 6 С/П «1001100» С/П «101100». И вновь вернемся к начальному расположению 3 ↑ 5 ↑ 7 С/П — «101010». Передаем ход партнеру С/П «11010».

Постараемся сохранить ситуацию: 5 ↑ 5 ↑ 7 С/П «2010» С/П, и калькулятор делает решительный ход — «2001». Итак, первая шашка выходит из игры, 5 ↑ 7 ↑ 8 С/П «1011». Если сейчас электронный партнер выведет из игры еще одну шашку или хотя бы переставит шашку с позиции 5 в позицию 6, то победа будет за человеком. С/П... Нет, калькулятор неумолим — «21», можно надеяться только на его ошибку. 7 ↑ 8 ↑ 8 С/П «12», С/П «3». В первом раунде победа досталась электронному партнеру. Может быть, в следующем поединке ситуация сложится иначе?

Структура программы (рис. 37):

00: обращение к подпрограмме 44—67.
01—04: вычисление $A = A(1) - A(2)$, где $A(1)$, $A(2)$, $A(3)$ — обозначения ячеек, где хранятся позиции шашек.
05—06: вычисление $A - A(3)$.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	КБПД	8Г	23	ИП9	69	46	+	10
01	ИП6	66	24	П7	47	47	8	08
02	ИП7	67	25	БП	51	48	↔	14
03	—	11	26	37	37	49	—	11
04	ПА	4—	27	$F_x \geq 0$	59	50	КП2	L2
05	ИП8	68	28	35	35	51	F10x	15
06	—	11	29	ИП7	67	52	ИП3	63
07	$F_x = 0$	5E	30	ИП8	68	53	+	10
08	27	27	31	+	10	54	П3	43
09	ИП7	67	32	П6	46	55	FO	25
10	1	01	33	БП	51	56	FL1	5L
11	—	11	34	37	37	57	47	47
12	П9	49	35	ИПА	6—	58	3	03
13	ИП8	68	36	П8	48	59	ПО	40
14	—	11	37	8	08	60	П1	41
15	$F_x = 0$	5E	38	КИП5	Г5	61	5	05
16	23	23	39	—	11	62	П5	45
17	ИП6	66	40	FL0	5Г	63	9	09
18	1	01	41	37	37	64	П2	42
19	—	11	42	КППД	—Г	65	ИП3	63
20	П6	46	43	БП	51	66	С/П	50
21	БП	51	44	00	00	67	В/О	52
22	37	37	45	П3	43			

Рис. 37

```

5 REM три шашки
10 DIM A(3)
20 INPUT "Ваш ход"; A(3), A(2), A(1)
30 GOSUB 150
50 PRINT "ход ЭВМ": A=A(1)-A(2)
60 IF A-A(3)=0 GOTO 90
70 IF A-A(3)>0 THEN A(1)=A(2)+A(3): GOTO 120
80 A(3)=A: GOTO 120
90 A=A(2)-1
100 IF A-A(3)=0 THEN A(1)=A(1)-1: GOTO 120
110 A(2)=A
120 GOSUB 160
130 IF S=3 THEN PRINT "Вы проиграли": STOP
140 GOTO 20
150 A=8-A(1): A(1)=8-A(3): A(2)=8-A(2): A(3)=A
160 S=0
170 FOR I=1 TO 3: S=S+10↑A(I): NEXT
180 PRINT "— — —"; S
190 RETURN
40 IF S=3 THEN PRINT "Вы проиграли": STOP

```

Рис. 38

07—08: проверка « $A-A(3)=0$ »? Если условие выполнено, то управление переходит на команду 09, иначе на команду 27.

09—12: вычисление $A=A(2)-1$.

13—14: вычисление $A-A(3)$.

15—16: проверка « $A-A(3)=0$ »? Если условие выполнено, то управление переходит на команду 17, иначе на команду 23.

17—22: вычисление $A(1)=A(1)-1$ и переход к команде 37.

23—26: запись A в $A(2)$ и переход к команде 37.

27—28: проверка « $A-A(3)>0$ »? Если условие выполнено, то управление пере-

ходит на команду 29, иначе на команду 35.

29—34: вычисление $A(1)=A(2)+A(3)$ и переход к команде 37.

35—36: запись A в $A(3)$.

37—41: преобразование содержимого массива по правилу $A(K)=8-A(K)$ и обмен содержимым регистров $A(1)$ и $A(3)$.

42—44: обращение к подпрограмме 4—67 и переход к началу программы.

44—67: подпрограмма формирования изображения игрового поля.

Программа на Бейсике (рис. 38) составлена в соответствии с блок-схемой (рис. 39). Программа практически ничем не отличается от той, что работает на микрокалькуляторе, однако желающие смогут так изменить подпрограмму 150—190, чтобы на экране формировалось игровое поле с изображениями именно шашек, а не нулей и единиц.

«Африканское сафари»

Единственным оружием в этой охоте будут сети. Если играть будут несколько человек, то каждому понадобится свой калькулятор.

Для удобства охотников саванна разбита на перенумерованные квадраты. Каждый квадрат имеет целочисленные

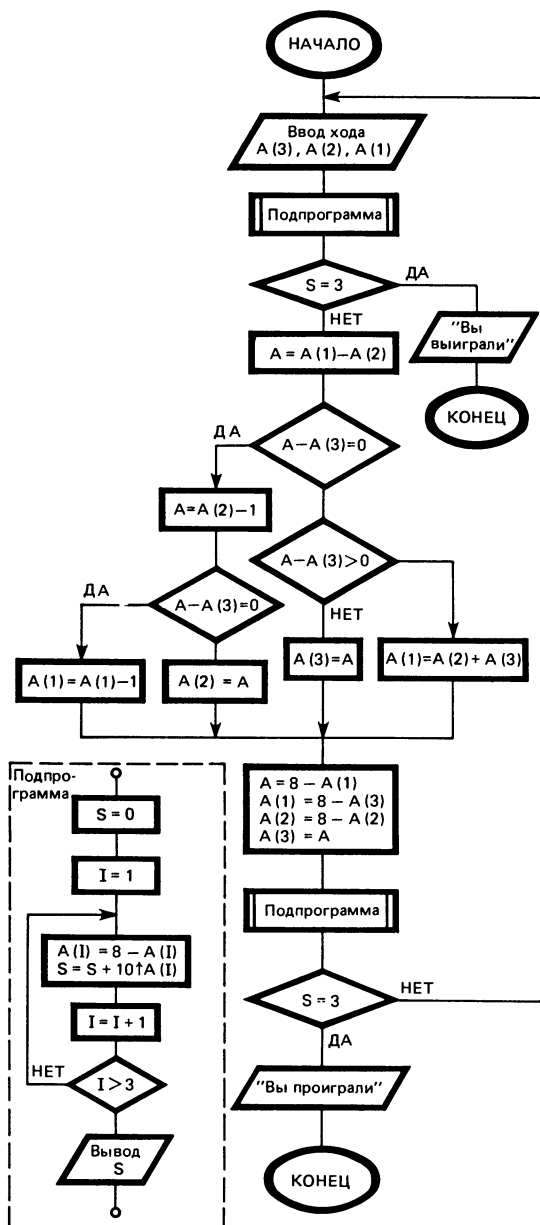


Рис. 39

координаты. Чтобы не спугнуть жирафа, который находится в квадрате с координатами 1009, 1009, вертолет высаживает охотника в квадрате 1000, 1000. Жираф пасется, перемещаясь за ход не более чем на один квадрат в одном из четырех возможных направлений: северном, восточном, южном или западном. Наблюдатели установили, что он никогда не пере-

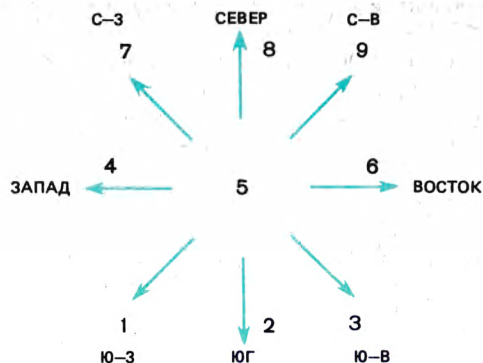


Рис. 40

мещается в диагональных направлениях. Это, конечно, значительно облегчает задачу охотника, который может перемещаться, как ему заблагорассудится. Правда, за один ход он может перейти лишь в один из восьми соседних квадратов.

Направление перемещения, задаваемое игроком, запомнить нетрудно: 8 — север, 2 — юг, 4 — запад, 6 — восток, 7 — северо-запад, 9 — северо-восток, 3 — юго-восток, 1 — юго-запад, что соответствует расположению цифровых клавиш на микрокалькуляторе, если клавиша с цифрой 8 принята за северное направление (рис. 40). Роль оставшейся без дела клавиши 5 выяснится позднее.

Цель игры, как уже говорилось, поймать жирафа. Для этого необходимо расставить сети в том квадрате, куда жираф придет следующим ходом. Если жираф будет пойман, то на индикаторе тотчас же возникнет число ходов, затраченных на охоту. Если цель игры еще не достигнута, на индикаторе появятся новые координаты местонахождения жирафа.

Иногда на пути охотника встречаются препятствия. Охотник узнает о них, когда в ответ на сделанный ход на индикаторе высветится ноль. Если теперь продолжать двигаться напрямик, то жираф обязательно заметит охотника и отбежит на три квадрата в северном или восточном направлении. Чтобы этого не случилось, можно попробовать найти обход, т. е. сделать другой ход. Впрочем, есть и третий вариант — остаться на месте. Для этого хода используется клавиша 5.

Засада может оказаться неудачной, и

тогда на индикаторе вновь появляется ноль.

Теперь, когда охотник представляет все трудности, с которыми он может столкнуться на охоте, осталось лишь познакомиться с инструкцией к программе. Чтобы отправиться в саванну, необходимо набрать на клавиатуре случайное число от 1 до 100 и запустить программу клавишами В/О, С/П. Очень скоро на индикаторе появятся координаты жирафа «10091009». Расшифровать их нетрудно. Первые четыре цифры 1009 — номер квадрата по горизонтали, вторая четверка 1009 — по вертикали.

Охотник высаживается с вертолета в квадрат с координатами 1000, 1000 и сразу же может двигаться — стоит только нажать клавишу, отвечающую выбранному направлению, и клавишу С/П. Если на индикаторе появится ноль, значит попытка неудачна, что-то мешает осуществить задуманное, необходимо изменить маршрут — нажать другую цифровую клавишу и С/П.

Есть и другой вариант. Достаточно еще раз нажать клавишу С/П, причем в этом случае на индикаторе обязательно высвечивается ноль. Такой ход означает, что охотник решил преодолеть препятствие и двигаться в направлении, выбранном последним нажатием цифровой клавиши, уже не обращая внимания на маскировку. Однако жираф, безусловно, заметит столь незадачливого охотника и отбежит в сторону.

Если в какой-то момент игры охотник потеряется или забудет, где он только что видел жирафа, то перед тем, как ввести цифру-ход, он может установить свои координаты. Для этого достаточно нажать клавиши ИП В. Координаты жирафа выяснять столь же легко, однако в этом случае нажимать следует клавиши ИП С.

Поймать жирафа сможет каждый, однако более опытному охотнику понадобится меньше времени, ведь размещение препятствий в густой траве саванны и перемещения жирафа — все имеет свои закономерности, которые откроются лишь для внимательного и настойчивого охотника. Закончить игру за одиннадцать ходов или менее — хороший результат.

Чтобы охотиться было удобно, можно



Рис. 41

нарисовать план и отмечать на нем все передвижения охотника и жирафа (рис. 41).

Структура программы (рис. 42): 00-21: по этим адресам размещены команды подготовки к игре. Исходное случайное число записывается в регистры 5 и 6. Содержимое R5, обозначим его ξ , определяет перемещение жирафа и меняется от шага к шагу. Число в R6, константа λ , определяет размещение препятствий в саванне. Кроме того, этот фрагмент программы формирует необходимые константы и координаты. Координаты жирафа записаны в RС, а координаты охотника в RВ.

22—50: этот блок команд предназначен для расшифровки хода охотника и проверки «видит ли жираф охотника?» по формуле $\sin[\lambda(2X+\Delta X)+1)] \geq 0$. Здесь X — координаты охотника в виде $1000 \cdot A+B$, A — широта, B — долгота, ΔX — приращение координат в той же форме.

51—56: переход на изменение хода или подготовка встревоженного жирафа к бегу.

57—64: формирование очередного случайного числа по формуле $\xi_i = \arcsin(\sin(10^8 \cdot \xi_{i-1}))$.

65—89: определение направления, в котором движется жираф, вычисление новых координат животного. Проверка «совпадают ли координаты жирафа с координатами охотника?»

90—93: жираф пойман, вывод на индикатор количества ходов, завершение игры.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	П5	45	32	ИПС	6С	64	П5	45
01	П6	46	33	ИП2	62	65	$F_x \geq 0$	59
02	Сх	0Г	34	ИП3	63	66	71	71
03	П4	44	35	Х	12	67	ИП1	61
04	3	03	36	—	11	68	П8	48
05	П3	43	37	ИП2	62	69	БП	51
06	F10х	15	38	ИП1	61	70	75	75
07	4	04	39	—	11	71	ИП1	61
08	F10х	15	40	ИП9	69	72	/—/	0L
09	П9	49	41	Х	12	73	П8	48
10	1	01	42	+	10	74	ИП9	69
11	П1	41	43	+	10	75	ИПД	6Г
12	ПД	4Г	44	П2	42	76	Х	12
13	+	10	45	+	10	77	ИПС	6С
14	ПА	4—	46	ИП6	66	78	+	10
15	Х	12	47	Х	12	79	ПС	4С
16	ПВ	4L	48	F_{\sin}	1C	80	ИП2	62
17	9	09	49	$F_x < 0$	5C	81	ПВ	4L
18	ИПА	6—	50	57	57	82	—	11
19	Х	12	51	Сх	0Г	83	$F_x \neq 0$	57
20	+	10	52	С/П	50	84	90	90
21	ПС	4C	53	$F_x = 0$	5E	85	ИП8	68
22	С/П	50	54	23	23	86	ПД	4Г
23	ПО	40	55	ИП3	63	87	ИПС	6С
24	2	02	56	ПД	4Г	88	БП	51
25	+	10	57	КИП2	Г2	89	22	22
26	ИП3	63	58	КИП4	Г4	90	ИП4	64
27	:	13	59	ИП5	65	91	С/П	50
28	П2	42	60	ВП	0C	92	БП	51
29	КИПО	Г0	61	8	08	93	00	00
30	КИП2	Г2	62	F_{\sin}	1C			
31	ИПВ	6L	63	Farcsin	19			

Рис. 42

Блок-схема (рис. 43) позволит еще нагляднее представить структуру программы. Дальнейшее совершенствование программы может идти по пути усложнения поведения жирафа, однако для этого придется сменить БЗ-34 с ее 98 шагами на более совершенные МК-61 или МК-52, а еще лучше на микро-ЭВМ.

«Охота на лис»

Название позаимствовано от известной спортивной игры, детали которой нетрудно узнать в нижеследующем описании.

В памяти микрокалькулятора хранится картина игрового поля размером 10×10 клеток. В произвольных клетках располагаются семь «лис» — радиопередатчики, посылающие в эфир сигнал «я здесь». Передатчики могут занимать соседствующие клетки или даже собираться по два-три на одной клетке. Впрочем, такое совпадение встречается не часто.

Игрок очередным своим ходом вводит в калькулятор положение «охотника»,

набирая на клавиатуре дробное двузначное число вида Y, X — целая и дробная его части выражают координаты по вертикали и горизонтали соответственно. «Охотник» вооружен приемником, имеющим направленную антенну, так что сигналы от «лис» принимаются лишь по восьми направлениям, как показано на рис. 44. На индикатор выводится число «лис», расположенных по этим направлениям. Если на каком-то направлении оказались две «лисы», то считаются обе.

Если координаты какой-либо «лисы» совпадают с положением «охотника», то на индикаторе появляется сообщение ЕГГОГ. Обнаруженная «лиса» исчезает с игрового поля и сигналы уже не посылает.

Меняя ход за ходом положение «охотника» и анализируя информацию, выдаваемую калькулятором, играющий путем логических умозаключений или графических построений должен обнаружить всех «лис» за наименьшее число ходов. Играть в эту игру можно и в одиночку, но интереснее проводить соревнования.

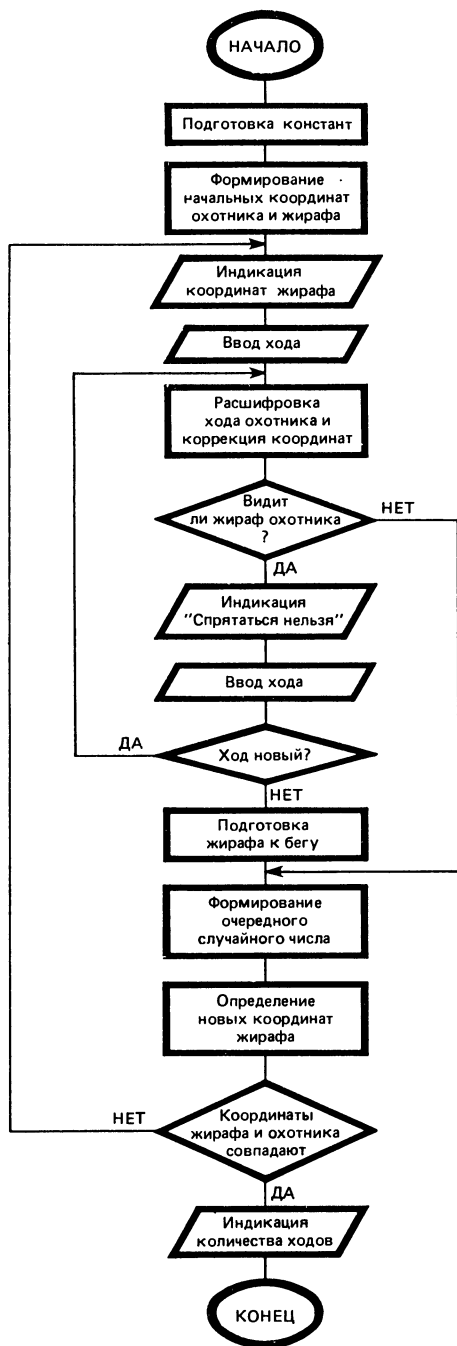


Рис. 43

Инструкция по работе с программой (рис. 45):

1. Нажмите В/О, наберите число 0,ЧЧММ, например 0,1234. Переключатель Р-Г установите в положение Р.

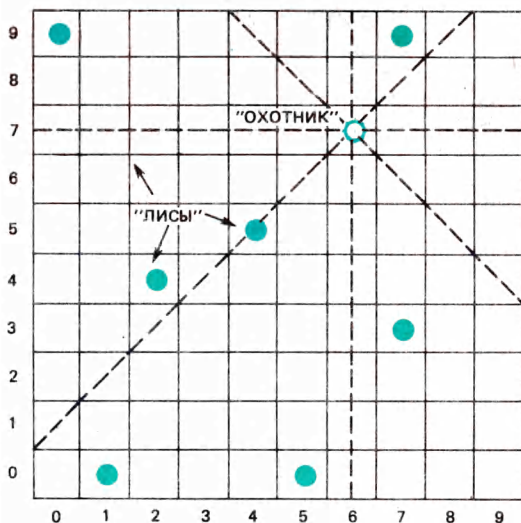


Рис. 44

Нажмите С/П. Калькулятор примется распределять «лис» по игровому полю. Через минуту на индикаторе появится сигнал готовности «0».

2. Введите координаты «охотника» Y , X С/П, где Y и X — целые числа от 0 до 9, разделенные запятой. Например, 7,6 — это позиция охотника на рисунке. На индикаторе появляется число «лис», видимых из данного квадрата, или ЕГГОГ, если «лисы» найдены. В нашем случае получаем 1. Затем вводим новые координаты «охотника» Y , X .

3. Для повторения игры с иной расстановкой «лис» переходите к п. 1.

Кому незнакомо желание заглянуть в ответы, собранные на последних страницах задачника, прежде чем решать трудную задачку? Любители подобных подсказок на сей раз будут разочарованы: программа построена таким образом, чтобы нельзя было узнать координаты «лис» с помощью клавиши ИП.

Структура программы:

00—09: подготовка «лис» к размещению. Вместо A и B стоящих в программе по адресам 01 и 02, нужно записать числа, определяющие число «лис» по формуле $10 \cdot A + B = 2 \cdot Л$. Число «лис» должно быть в диапазоне от единицы до семи, например, для игры с пятью «лисами» нужно задать 01.1 02.0.

10—30: команды, выполняемые в цикле, генерируют случайное число и форми-

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	П8	48	31	Сх	0Г	62	52	52
01	А	0А	32	С/П	50	63	КИП2	Г2
02	В	0В	33	КНОП	54	64	КИП2	Г2
03	П2	42	34	1	01	65	ВП	0С
04	ПО	40	35	+	10	66	,	0—
05	1	01	36	↑	0Е	67	БП	51
06	0	00	37	П6	46	68	33	33
07	ПА	4—	38	ВП	0С	69	ИП8	68
08	Фх ²	22	39	П8	48	70	КИП↑	ГЕ
09	ПС	4С	40	0	00	71	ПС	4С
10	ИП8	68	41	П4	44	72	ВП	0С
11	ВП	0С	42	ИП2	62	73	—	11
12	9	09	43	ПО	40	74	Фх≠0	57
13	Fc _{os}	1Г	44	КИПО	Г0	75	88	88
14	F _{arccos}	1—	45	ИП6	60	76	ПС	4С
15	Fπ	20	46	—	11	77	+	10
16	:	13	47	Фх = 0	5Е	78	Фх≠0	57
17	П8	48	48	69	69	79	88	88
18	ИПС	6С	49	ИПО	60	80	Фх ²	22
19	Х	12	50	БП	51	81	ИПС	6С
20	1	01	51	56	56	82	ИПА	6—
21	1	01	52	КИПС	ГС	83	Х	12
22	+	10	53	КП↑	LE	84	Фх ²	22
23	П1	41	54	ИПС	6С	85	—	11
24	КИП1	Г1	55	ПО	40	86	Фх = 0	5Е
25	ИП1	61	56	2	02	87	89	89
26	ИПА	6—	57	+	10	88	КИП4	Г4
27	:	13	58	ПС	4С	89	FL0	5Г
28	КПО	LO	59	ИП2	62	90	44	44
29	FL0	5Г	60	—	11	91	ИП4	64
30	10	10	61	Фх ≥ 0	59	92	БП	51
						93	32	32

Рис. 45

руют из него сразу обе координаты очередной «лисы».

31—32: формируется сигнал «установка лис завершена».

33—43: подготовка к поиску.

44—48: проверка «найдена ли очередная «лиса»?».

49—68: в случае если «лиса» найдена, происходит выключение ее передатчика. На индикатор выводится сообщение ЕГГОГ.

69—88: проверка «видна ли очередная «лиса» охотнику?». Эти же команды подсчитывают число видимых «лис».

69—93: команды, завершающие поиск.

Программа по своей структуре близка к линейной, поэтому на этот раз блок-схема не нужна.

Те читатели, которые станут переводить программу для компьютера, смогут увеличить размеры поля и число «лис». Можно даже перейти от двумерной игры к трехмерной.

БЫСТРОТА И ТОЧНОСТЬ*

До сих пор все игры, предназначенные для микрокалькулятора, мы вели в условном времени. Можно было сколь угодно долго раздумывать над каждым ходом — на результатах это не сказывалось.

Игры, собранные в этой главе, позволят не только испытать, но и тренировать свою реакцию. Игра «Кто первый» оттачивает ее в обстановке соревнования между двумя участниками.

Игра «Восемь цифр в уме» поможет развить оперативную, или, как ее еще называют, «короткую» память на цифры, столь необходимую нам в повседневной жизни.

Принцип динамического управления, который заложен в программу игры «Учебный полет», можно использовать и для серьезных расчетов.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	Сх	ОГ	17	↔	14	34	—	11
01	ПА	4—	18	ИПД	6Г	35	$F x \geq 0$	59
02	1	01	19	—	11	36	38	38
03	1	01	20	ПД	4Г	37	1	01
04	↑	ОЕ	21	ИПС	6С	38	ИПА	6—
05	ПС	4С	22	Х	12	39	+	10
06	F1/X	23	23	Fx^2	22	40	ПА	4—
07	ПД	4Г	24	1	01	41	2	02
08	Х	12	25	+	10	42	$F \cos$	1Г
09	ПВ	4Л	26	ПО	40	43	$F x < 0$	5С
10	ИПД	6Г	27	Сх	ОГ	44	10	10
11	ИПС	6С	28	ИПВ	6Л	45	↔	14
12	Х	12	29	FLO	5Г	46	С/П	50
13	Fπ	20	30	29	29	47	Б/П	51
14	+	10	31	С/П	50	48	10	10
15	ПД	4Г	32	+	10			
16	КИПД	ГГ	33	1	01			

Рис. 46

«Кто первый»*

Играют двое — каждый выбирает себе одну клавишу из двух: 0 или Сх. Цель игры — как можно быстрее нажать на свою клавишу, лишь только на индикаторе высветится число 9,9999999—01. За преждевременное, до окончания работы программы, нажатие клавиши засчитывается поражение. В этом случае следует возвратиться к началу игры — В/О, С/П.

Итак, программа набрана, клавиши распределены. Нажимаем В/О, С/П. Через непредсказуемый промежуток времени (он может длиться от 10 до 70 с), на индикаторе появляются девятки. Теперь — кто быстрее! После того как нажата клавиша 0 или Сх, продолжаем игру, нажав на клавишу С/П. Кстати, опоздавший может и не нажимать свою клавишу.

Если переключатель Р-Г находится в положении Р, то на индикаторе появится текущий счет. Чтобы начать следующий раунд, необходимо еще раз нажать С/П. Положение Г означает, что партнерам не будут известны результаты каждого тура. В этом случае игра не прерывается.

Как уже говорилось, калькулятор ведет счет победам и поражениям. Чтобы узнать результат нескольких партий, каждый из партнеров может воспользоваться клавишами ИПА. Положительное число на индикаторе соответствует числу побед игрока, нажимающего кла-

вишу 0. Напротив, для того, кто управляет клавишей Сх, целью будет отрицательный счет игры. Разумеется, возможен и третий результат — ничья, т. е. ноль в регистре А.

Программа введена? Приступаем к игре. Впрочем, нелишне проверить точность набора программы (рис. 46) — установите переключатель Р-Г в положение Р, нажмите В/О, С/П. Когда индикатор выдаст команду «старт», не спеша нажмите клавиши 0, С/П. На индикаторе появится единица. Еще раз С/П... «старт» Сх С/П. В этот раз на индикаторе должен появиться ноль — сигнал, что программа в порядке и можно начинать игру.

Структура программы:

00—01: установка итогового табло на ноль.

02—05: подготовка и запись константы 11 в РС.

06—07: запись в РД исходного случайного числа.

08—09: запись в РВ сигнала «старт».

10—20: формирование случайного числа от нуля до единицы.

21—26: вычисление и запись в Р0 длительности случайного временного интервала.

27—28: подготовка стека.

29—30: формирование задержки от нажатия С/П до появления сигнала «старт».

31: остановка и индикация сигнала «старт»: «9,9999999—01».

32—40: проверка состояния стека и вычисление счета игры.


```

5 REM кто быстрее
10 S1=0: S2=0
20 PRINT "счет"; S1, S2
30 FOR I=1 TO 5000*RND(0): NEXT
40 FOR I=1 TO 10: GET A$: NEXT
50 PRINT: PRINT: PRINT: "старт"
60 GET A$
70 IF A$="" THEN 60
80 IF A$="1" THEN S1=S1+1
90 IF A$="0" THEN S2=S2+1
100 PRINT CHR$(147)
100 GOTO 20

```

Рис. 47

```

1.1. XEQUTE FX (-1, 177660, 64)
2.1. SET J=FX(1, 177660)
3.1. IF (J-192) 4.1, 5.1, 4.1
4.1 GOTO 2.1
5.1. SET K=FX(1, 177662)

```

Рис. 48

41—44: проверка «в каком положении переключатель Р-Г».

45—46: индикация текущего счета.

47—48: переход к новому туру игры.

На этот раз структура программы настолько проста, что блок-схема алгоритма не понадобится. Программу очень легко перевести, например, на Бейсик (рис. 47).

Для тех, кто пожелает воспользоваться ею, следует сделать небольшое примечание. Подобно тому как люди, живущие в разных местностях, могут говорить на различных диалектах одного и того же языка, компьютеры тоже работают с разными диалектами — различными версиями алгоритмических языков.

Употребленный в приводимой программе оператор GET A\$ в других диалектах Бейсика может выглядеть, например, как A\$=INKEY или A\$=KEYIN или TINRUT A\$.

Для тех, кто программирует на Фокале, смоделировать эту команду окажется не так просто. Один из вариантов замены приведен на рис. 48.

Строка 1.1 запрещает принимать от клавиатуры сигнал прерывания, т. е. теперь нажатие клавиш не будет приостанавливать выполнение программы.

Строка 2.1 опрашивает так называемый регистр состояния клавиатуры. Это специальная ячейка памяти, содержимое которой зависит от того, нажата ли хоть одна из клавиш клавиатуры. В случае

если клавиша нажата, / примет значение 192.

Строка 3.1 — это не что иное, как проверка «нажата ли клавиша?». Если клавиша не нажата, то управление передается на команду 4.1.

Строка 4.1 — безусловный переход. Поскольку клавиша не нажата, компьютер вновь возвращается к опросу клавиатуры.

Строка 5.1. Сюда управление передается только, когда клавиша нажата. Команда, записанная в строке, «читает» код последней нажатой клавиши и присваивает его переменной K.

«Восемь цифр в уме»*

Хорошая память на цифры необходима не только любителям вычислений «в уме», но и всем, кому приходится сталкиваться с необходимостью быстро запомнить ряд цифр. Примеров тому немало, вспомним хотя бы скороговорку профессионалов из Мосгорсправки, диктующих телефонный номер.

Кто-то справляется с подобными задачами без труда, кому-то приходится беспрестанно повторять необходимый номер, пока он не перейдет из оперативной памяти в долговременную.

Развить оперативную память на цифры, в общем-то, нетрудно. Поможет в этом занимательная игра.

В ее основу положен гибкий алгоритм. Программа реализует один из двух режимов в зависимости от положения переключателя Р-Г. В режиме Р к числу, которое нужно запомнить, каждая новая цифра будет добавляться с левого края. Когда игрок справится с задачей запоминания таких чисел, можно поставить переключатель в положение Г.

Теперь цифра будет добавляться не только слева, но и справа. Каждый раз конкретное положение новой цифры будет случайным. Правда, сыграв с машинкой несколько партий в таком режиме, вы заметите, что позиция, на которую записывается каждая новая цифра, зависит от ее величины. В то же время нетрудно добавить несколько команд, чтобы расположение новой цифры перестало быть определенным. Способов можно предложить много, но интересно

найти тот, который позволит обойтись меньшим числом команд.

Рассмотрим, как играть с микрокалькулятором в обоих режимах. Набрав программу, установите на ноль счетчик адресов командой В/О, наберите текущее время в виде 0, ЧЧММ и запустите счет командой С/П. Через 5—10 с на индикаторе появится случайная цифра от единицы до девяти. Предположим, что время было введено в виде дроби 0,1815. Пусть переключатель Р-Г стоял в положении Р. В этом случае на индикаторе появится «2». Так как решено тренировать оперативную память, то как можно быстрее постарайтесь нажать клавишу 2 и клавишу С/П. После недолгой паузы на индикаторе появится «62». Пока что все просто 6 2 С/П... Легко заметить, что каждая новая цифра добавляется к числу слева.

...«5316462». Вводим 5 3 1, дальше вроде бы 4 6 и, как всегда, 2 С/П... «1»! На индикаторе вновь появилось одноразрядное число. Это результат ошибки игрока. В том случае, если игроку удалось безошибочно ввести все числа, включая восьмиразрядное, на индикаторе появляется знак победы — число «11111111».

Для новой игры никаких исходных данных не требуется. Нажмите лишь клавишу С/П.

Структура программы:

00: запись стартового числа 0,ЧЧММ в регистр Д.

01—04: подготовка необходимых констант.

05—16: вычисление случайного числа.

17—22: формирование в регистре С целого случайного числа в диапазоне от единицы до девяти.

23—26: команды, определяющие положение переключателя Р-Г.

27—36: эта группа команд получает управление, если переключатель установлен в положение Р. Команды записывают новую цифру слева к числу, которое следует запомнить, помещают результат в РА и выводят число на индикатор.

37—40: проверка «число введено верно?». Если память не подвела, то управление перейдет на следующий фрагмент программы, в противном случае игра возобновится с самого начала.

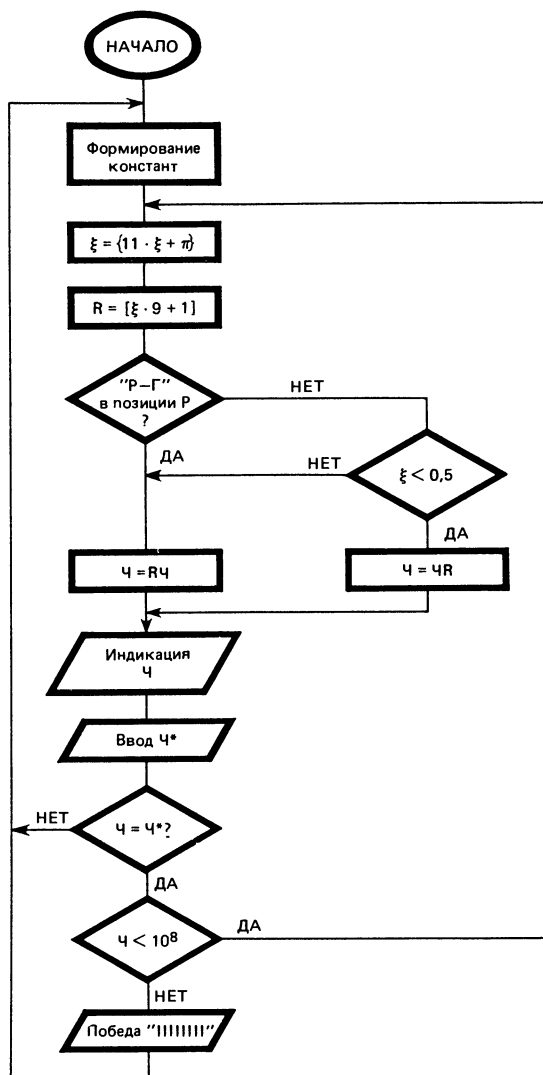


Рис. 49

41—42: цикл «число восьмиразрядное?». Если число на индикаторе меньше 1.10^8 , то управление переходит на блок 05—16. 43—47: формирование и индикация сигнала победы 11111111.

48: это команда позволяет продолжить игру.

49—55: команды, дописывающие к числу на индикаторе новую цифру справа.

56—63: команды, которые в режиме Г решают, где дописать очередную цифру — справа или слева.

Помещенная на рис. 49 блок-схема соответствует не только программе для

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ПД	4Г	22	КИПС	ГС	44	F10x	15
01	Сх	0Г	23	Fπ	20	45	9	09
02	ПА	4—	24	Fsin	1С	46	:	13
03	8	08	25	Fx = 0	5E	47	С/П	50
04	Π0	40	26	56	56	48	В/О	52
05	ИПД	6Г	27	8	08	49	ИПС	6С
06	1	01	28	ИΠ0	60	50	ИПА	6—
07	1	01	29	—	11	51	1	01
08	X	12	30	F10x	15	52	0	00
09	Fπ	20	31	ИПС	6С	53	X	12
10	+	10	32	X	12	54	БΠ	51
11	ПД	4Г	33	ИПА	6—	55	34	34
12	КИПД	ГГ	34	+	10	56	ИПД	6Г
13	≥	14	35	ПА	4—	57	2	02
14	ИПД	6Г	36	С/П	50	58	F1/x	23
15	—	11	37	ИПА	6—	59	—	11
16	ПД	4Г	38	—	11	60	Fx ≥ 0	59
17	9	09	39	Fx = 0	5E	61	49	49
18	X	12	40	01	01	62	БΠ	51
19	1	01	41	FLO	5Г	63	27	27
20	+	10	42	05	05			
21	ΠС	4С	43	8	08			

Рис. 50

```

5 REM восемь цифр
10 INPUT "Р или Г?"; FLAG$
20 K=0
30 FOR I=1 TO 8
40 R=INT (9 * RND (0) +1)
50 IF FLAG$ = "Г" THEN 80
60 K=K+R*INT (10↑ (I-1))
70 GOTO 100
80 IF R < 5 THEN 60
90 K=K*10+R
100 PRINT K
110 FOR T=1 TO 500:NEXT
120 PRINT CHR$(147)
130 INPUT KK
140 IF K-KK <> 0 THEN PRINT "ошибка": GOTO 10
150 NEXT I
160 PRINT "победа"
170 GOTO 10

```

Рис. 51

калькулятора (рис. 50). По ней написана и программа этой же игры на Бейсике для микро-ЭВМ (рис. 51). Теперь компьютер сам проследит за тем, чтобы пауза между появлением числа на индикаторе или экране и вводом числа с клавиатуры не превышала секунды.

«Отвечают» за паузу команды FOR I=1 TO 500: NEXT. Этот, как его принято называть, «пустой» цикл выполняется как раз около секунды, причем постоянно демонстрируется число, которое требуется запомнить. Лишь только выполнение цикла завершается, экран очищается командой PRINT CHR\$(147) и компьютер ожидает ввода числа.

В других версиях Бейсика есть спе-

циальная команда для формирования паузы, которая так и называется PAUSE. Можно встретить и аналогичную ей команду DELAY.

Точно так же обстоит дело и с командой очистки экрана. Чаще всего, однако, встречается команда CLS (от английского CLEAR SCREEN — очистить экран).

«Учебный полет»

С помощью микрокалькулятора можно испытать себя в роли пилота. Трасса полета специально предназначена для тренировки начинающих летчиков. Профиль трассы изображен на рис. 52.

Цель игры заключается в том, чтобы пролететь по маршруту, т. е. подняться над горой, поднырнуть под тучу, еще раз подняться над горой, после чего аккуратно зайти на посадку и приземлиться на другой аэродром.

Препятствия условны, однако если произойдет авария — самолет войдет в грозовую тучу или врежется в «гору», — то полет не будет засчитан.

Как только пилот освоит управление самолетом и изучит трассу полета, можно усложнить рельеф местности. Те, кому маршрут поначалу покажется чересчур сложным, смогут приподнять «тучу» выше или сделать немного ниже «горы».

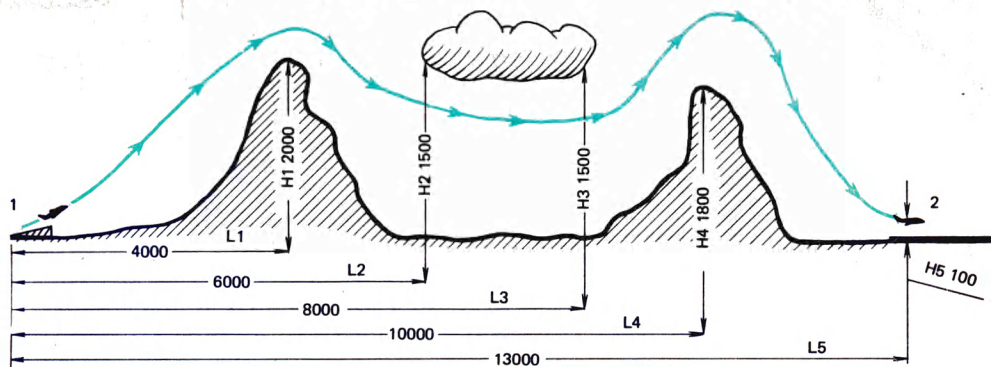


Рис. 52

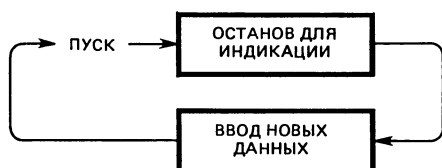


Рис. 53

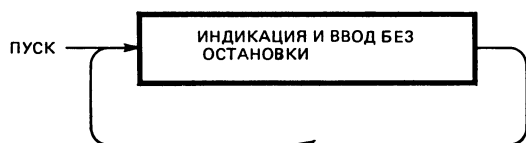


Рис. 54

Важное отличие этой игры от большинства игр, разработанных специально для программируемого микрокалькулятора, заключается в том, что на этот раз игроку придется иметь дело не с традиционной последовательностью действий, изображенной на рис. 53.

Такая схема не подходит для этой игры. В самом деле, между остановом для индикации и повторным нажатием клавиши С/П может пройти сколь угодно много времени, тогда как полет на самолете — явление быстротечное.

Для моделирования подобных процессов используют иную последовательность — рис. 54. Многие игры для персональных компьютеров отличаются именно таким принципом построения.

Вернемся к самолету. Он уже занимает исходную позицию на взлетной по-

лосе аэродрома (слева на рис. 55). Кстати, рисунок стоит укрупнить, разметить на нем высоту и расстояние от аэродрома 1 до аэродрома 2.

Введите в память машинки программу (рис. 55) и рельеф местности в соответствии с табл. 7. Поставьте переключатель Р-Г в положение Г. Когда все будет готово, останется лишь нажать клавиши В/О, С/П.

Итак, если нажать клавишу С/П, самолет начнет разбег и поднимется в воздух. Теперь все будет зависеть только от пилота. Переключатель Р-Г — штурвал самолета. Штурвал на себя (в положение Г) — самолет послушно задержит нос, от себя (положение Р) — направится к земле.

Среднее положение переключателя не влияет на траекторию полета. Если пилот решил набирать высоту и отклонил штурвал в положение Г, то самолет устремляется вверх, причем за каждые 500 м полета траектория самолета отклоняется на 15° от прежнего положения. Соответственно при снижении траектория также будет меняться со скоростью — $15^\circ/500$ м.

Если пилоту удастся с первой же попытки довести самолет до аэродрома 2 и правильно посадить его, то счет по программе автоматически остановится, и на индикаторе появится угол, зафиксированный судьями, которые наблюдают за посадкой. Это угол между горизонтальной аэродрома и посадочной траекторией самолета. Величина угла в пределах от нуля до тридцати градусов означает, что посадка была успешной.

Угол, превышающий 30° , — признак

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИПВ	6L	33	X	12	66	КИПО	Г0
01	ИПВ	6L	34	ИПС	6C	67	↑	0E
02	1	01	35	+	10	68	↑	0E
03	5	05	36	ПП	53	69	$F_x 2$	22
04	+	10	37	85	85	70	$F\sqrt{}$	21
05	ИПВ	6L	38	ПС	4C	71	:	13
06	FBx	0	39	FO	25	72	ИПД	6Г
07	—	11	40	X	12	73	X	12
08	ПВ	4L	41	ИПД	6Г	74	—	11
09	5	05	42	+	10	75	$F_x \geq 0$	59
10	7	07	43	ПП	53	76	78	78
11	:	13	44	85	85	77	K—	27
12	9	09	45	ПД	4Г	78	ИПО	60
13	0	00	46	$F_x \geq 0$	59	79	1	01
14	$F \cos$	1Г	47	77	77	80	—	11
15	$F_x \neq 0$	57	48	ПП	53	81	$F_x = 0$	5E
16	20	20	49	91	91	82	00	00
17	$F_x \geq 0$	59	50	ИПВ	6L	83	ИПВ	6L
18	24	24	51	ПП	53	84	С/П	50
19	FO	25	52	91	91	85	7	07
20	FO	25	53	ИПС	6C	86	$F 10^x$	15
21	FO	25	54	ПП	53	87	+	10
22	ПВ	4L	55	91	91	88	FBx	0
23	↔	14	56	КИПО	Г0	89	—	11
24	↔	14	57	—	11	90	V/O	52
25	$F \sin$	1C	58	$F_x < 0$	5C	91	↑	0E
26	FBx	0	59	66	66	92	↑	0E
27	$F \cos$	1Г	60	ИПО	60	93	↑	0E
28	5	05	61	1	01	94	↑	0E
29	0	00	62	+	10	95	↑	0E
30	0	00	63	ПО	40	96	↑	0E
31	↔	14	64	БП	51	97	V/O	52
32	Bx	0	65	00	00			

Рис. 55

Таблица 7

ПД	ПС	ПВ	ПА	П9	П8	П7	П6	П5	П4	П3	П2	П1	П0
0	0	0	4000	2000	6000	—1500	8000	—1500	10 000	1200	13 000	—100	11
R	S	φ	L ₁	H ₁	L ₂	H ₂	L ₃	H ₃	L ₄	H ₄	L ₅	H ₅	

неудачи. Самолет почти наверняка получил повреждения. Ну а уж угол от 70° и более означает верную аварию.

Итак, мы остановились на том, что самолет поднялся в воздух. Параметры полета появятся на индикаторе примерно через 20 с. Сначала высота — h , затем угол — φ и пройденное расстояние S — эти три величины поочередно мерцают на индикаторе. Если числа будут видны плохо, советуем ослабить внешнее освещение, например, прикрепить на окошечко индикатора шторку так, чтобы тень от нее легла на индикатор.

После того как все три параметра по-

очередно промелькнули на индикаторе, в распоряжении пилота остается 5 с, чтобы отреагировать должным образом. Если и через 5 с штурвал будет оставаться неподвижным, самолет уже не станет слушаться запоздалой команды и продолжит полет по прежней траектории. Признаком того, что двигать штурвал уже поздно, является длительная, примерно в 3 с пауза в мерцаниях индикатора, связанная с вычислением функции $\cos(x)$ — см. адрес 14-й программы.

Что же делать, когда пилоту наскучит один и тот же рельеф местности? Достаточно разобраться с правилами програм-

мирования маршрута, и полет можно будет продолжить в новых условиях.

Весь рельеф в виде пар чисел L_i , H_i хранится в регистрах P1—PA. Первое число — расстояние от точки взлета до препятствия L_i — вводится в PA, P8, P6, P4, P2, т. е. в регистры с четными номерами. Второе — высота препятствия H_i — вводится в регистры с нечетными номерами: P9, P7, P5, P3, P1.

Поясним программирование рельефа конкретным примером. Предположим, речь идет о горе, удаленной от взлетной площадки на 4 км. Пусть высота горы равна 2 км, тогда ее координаты пилот найдет в регистре A — 4000 и в регистре 9—2000.

Следует учитывать, что высота препятствий типа «туча», т. е. препятствий, под которыми нужно пролететь, вводится со знаком минус. Для учебной трассы, что изображена на рисунке, координаты «тучи» будут выглядеть так: P8=6000, P7=—1500, P6=8000, P5=—1500.

Регистры PB, PC, PD отведены для хранения текущих данных: угол траектории φ , пройденное расстояние S по горизонтали, высота полета h соответственно.

Освоить управление самолетом не так сложно, как это может показаться с первого взгляда, однако поначалу самолет наверняка будет врезаться и в «гору», и в «тучу», и в «землю» или просто пролетать мимо аэродрома. Во всех этих случаях на индикаторе высветится сигнал ошибки ЕГГОГ.

Программа игры (см. рис. 56) позволяет изменять конструкцию самолета по желанию играющего. Так, можно заменить двигатель на более мощный, увеличив тем самым скорость полета, или, напротив, уменьшить скорость. Команды, определяющие величину скорости, размещены по адресам 28, 29, 30. Скорость до 999 км/ч можно вводить, последовательно записывая составляющие число цифры.

Маневренность самолета также легко изменить. В самом деле, величина 15° записана по адресам 02, 03, и пилот может заменить 15° на любое значение от 1° до 99° . Конкретная величина после нескольких попыток будет, по-видимому, определять квалификацию пилота.

Чтобы пользоваться программой было

удобно, объединим все исходные данные в табл. 5.

После ввода данных и программы нажмите В/О, установите переключатель Р-Г в положение Г и начинайте взлет — нажимайте клавишу С/П.

Для повторения игры достаточно записать 11 в P0 и нажать клавиши Сх ПВ ПС ПД В/О С/П.

Структура программы:

00—22: вычисление нового значения угла φ в зависимости от положения штурвала Р-Г.

12—18: определение положения переключателя Р-Г.

23—25: вычисление синуса угла φ .

26—27: вычисление косинуса угла φ .

28—35: вычисление расстояния, пройденного от аэродрома 1 к аэродрому 2, по формуле $S = S + 500 \cos \varphi$.

36—38: отделение целой части S для удобства пилота.

39—42: вычисление текущей высоты полета по формуле $h = h + 500 \sin \varphi$.

43—45: отделение целой части h для удобства пилота.

46—47: проверка «не врезался ли самолет в землю?».

48—49: индикация высоты полета в метрах.

50—52: индикация пройденного расстояния в метрах.

53—55: индикация текущего угла траектории.

56—59: проверка «не пролетел ли самолет мимо очередного препятствия?».

60—63: восстановление адреса регистра, в котором записана координата очередного объекта.

64—65: переход к новому циклу расчета.

66—67: проверка «не потерпел ли самолет аварию?».

77: индикация сообщения об аварии — ЕГГОГ.

78—82: проверка «совершена ли посадка на аэродром 2?».

83—84: индикация угла, под которым самолет зашел на посадку.

85—90: подпрограмма отделения целой части числа.

91—97: задержка для индикации полетных данных.

Программа игры для микро-ЭВМ, использующая графические возможности машины, получается достаточно сложной, однако, используя блок-схему, не-

трудно перевести программу на язык любой микро-ЭВМ.

ЗАКЛЮЧЕНИЕ

Надеемся, что, читая эту книгу с микрокалькулятором в руках, вы не только скоротали часы досуга, но хотя бы немного заинтересовались вычислительной техникой. Может быть, игра с микрокалькулятором послужит для вас своеобразным компасом, который поможет лучше ориентироваться в сложном мире вычислительной техники, станет первой ступенью лестницы, которая ведет в немного загадочный, но такой увлекательный мир ЭВМ.

Возможно, разобравшись в структуре программ, вы напишите собственный, лучший вариант той или иной игры. В таком случае не сомневайтесь — вы хорошо освоили программируемый микрокалькулятор и можете браться за решение разнообразных практических задач, которые встречаются в вашей работе или учебе.

Если вам понравились игры и вы уже досконально освоили программирование на микрокалькуляторе, смело переходите на микро-ЭВМ. Освоить один из языков программирования, например Бейсик, вам будет легче, чем тем, кто впервые столкнется с вычислительной техникой. Быть может, когда у вас появится личный компьютер, пусть даже такой несложный, как «Электроника МК-85», вы вернетесь к алгоритмам из этой книги — перевод программ не вызовет больших затруднений.

СПИСОК ЛИТЕРАТУРЫ

Учебная литература по программированию для калькуляторов.

1. Данилов И. Д. Секреты программируемого микрокалькулятора. — М.: Наука, 1985.

2. Славин Г. В. Программирование на программируемых микрокалькуляторах типа «Электроника БЗ-34». — Таллин: Валгус, 1984.

3. Пухначев Ю. В., Данилов И. Д. Микрокалькуляторы для всех. — М.: Знание, 1986.

4. Трохименко Я. К., Любич Ф. Д. Программирование микрокалькуляторов «Электроника МК-52» и «Электроника МК-61». — Киев: Техника, 1987.

Циклы статей в центральных журналах:

5. «Наука и жизнь», рубрика «Семинар по

информатике», цикл статей «Школа начинающего программиста», начало в № 6, 1985.

6. «Техника — молодежи», рубрика «Для всех профессий», начало в № 2, 1985.

7. «Химия и жизнь», рубрика «Микро-ЭВМ для химиков», начало в № 9, 1984.

Справочная литература с библиотеками программ для микрокалькуляторов, научно-популярная литература о микрокалькуляторах.

8. Белый Ю. А. Считающая микроэлектроника. — М.: Наука, 1984.

9. Дьяконов В. П. Справочник по расчетам на микрокалькуляторах. — М.: Наука, 1985.

10. Кройль Г. Что умеет мой микрокалькулятор? — М.: Мир, 1981.

11. Цветков А. Н., Епанечников В. А. Прикладные программы для микро-ЭВМ «Электроника БЗ-34», «Электроника МК-56», «Электроника МК-54». — М.: Финансы и статистика, 1984.

12. Чакань А. Что умеет карманная ЭВМ? — М.: Радио и связь, 1982.

Сборники игровых программ и занимательных задач.

13. Микрокалькуляторы в играх и задачах. — М.: Наука, 1986.

14. Романовский Т. Б. Микрокалькуляторы в рассказах и играх. — Рига: Зинатне, 1984.

15. Трохименко Я. К. Игры с микро-ЭВМ. — Киев: Техника, 1986.

16. Трохименко Я. К. Микрокалькулятор, ваш ход! — М.: Радио и связь, 1985.

17. «Наука и жизнь», рубрика «Человек с микрокалькулятором», начало в № 10, 1983.

18. «Техника — молодежи», рубрика «Клуб электронных игр», начало в № 6, 1985.

Научно-популярное издание

Алексей Борисович БОЙКО

ИГРЫ С МИКРОКАЛЬКУЛЯТОРОМ

Главный отраслевой редактор Л. А. Ерлыкин

Редактор Г. Г. Карвовский

Мл. редактор Е. Е. Куликова

Художник Л. П. Ромасенко

Худож. редактор М. А. Бабицева

Техн. редактор И. Е. Жаворонкова

Корректор Е. И. Альшевская

ИБ № 8984

Сдано в набор 21.09.87. Подписано к печати 26.10.87. Т-21829. Формат бумаги 70×100¹/₁₆. Бумага кн.-журнальная. Гарнитура «Литературная». Печать офсетная. Усл. печ. л. 3,90. Усл. кр.-отт. 8,12. Уч.-изд. л. 4,84. Тираж 36 218 экз. Заказ 2549. Цена 11 коп. Издательство «Знание». 101835, ГСП, Москва, Центр, проезд Серова, д. 4. Индекс заказа 874311.

Ордена Трудового Красного Знамени
Чеховский полиграфический комбинат ВО «Союзполиграфпром»
Государственного комитета СССР по делам издательств,
полиграфии и книжной торговли
142300, г. Чехов Московской области

Адрес подписчика:

*Издательство «Знание» — крупнейшее
в стране издательство по выпуску
научно-популярной литературы.*

**Издательство выпускает
37 серий подписных
научно-популярных брошюр**

Подписная
научно-
популярная
серия



*Издательство
«Знание»*

Дорогой читатель!

Брошюры этой серии в розничную продажу не поступают,
поэтому своевременно оформляйте подписку.

Подписка на брошюры издательства «Знание» ежеквартальная,
принимается в любом отделении «Союзпечати».

Напоминаем Вам, что сведения о подписке Вы можете
найти в «Каталоге советских газет и журналов»
в разделе «Центральные журналы»,
рубрика «Брошюры издательства «Знание»

Цена подписки на год 1 р. 32 к.

**Наш адрес:
СССР,
Москва,
Центр,
проезд Серова, 4**

**МАТЕМАТИКА
КИБЕРНЕТИКА**