

Чтение книги не требует специальных знаний ни по математике, ни по вычислительной технике. Но и просто развлекательной ее назвать нельзя. Книга дает возможность читателю заглянуть в мир ЭВМ, мир, в котором всем нам предстоит жить в недалеком будущем.



ЧЕЛОВЕК И КОМПЬЮТЕР

И.Д.Данилов  
Г.В.Славин

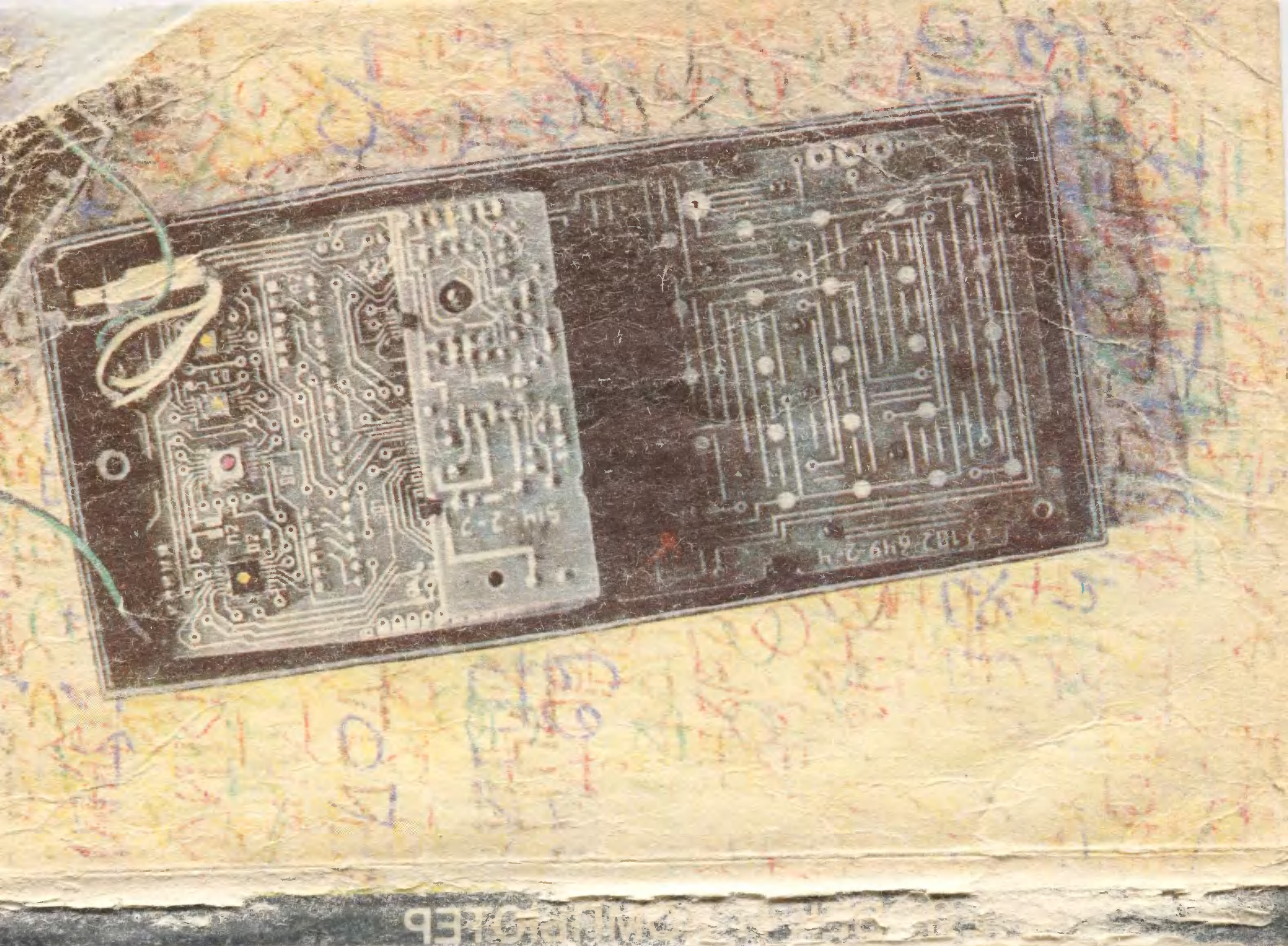
5

ВЕЧЕРОВ  
С МИКРО-  
КАЛЬКУЛЯТОРОМ

Калькулятор в семье. Как научиться писать программы? Калькулятор работает таймером. Бюро погоды на дому. Нужно ли оптимизировать программы? С микрокалькулятором в космос. Что такое структурное программирование? Об этом рассказывает книга.

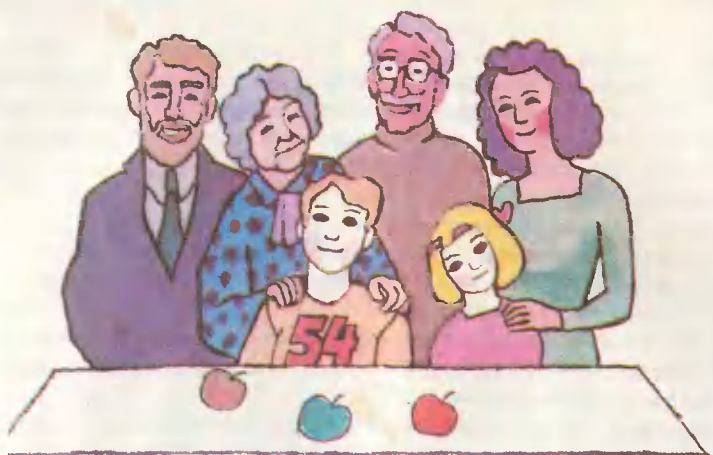






		2 <sup>й</sup> СИМВОЛ														
		0	1	2	3	4	5	6	7	8	9	-	L	Г	Е	
1 <sup>й</sup> СИМВОЛ	0	0	1	2	3	4	5	6	7	8	9	,	/	БП	Сх	1 FBx
	1	+	-	x	÷	$\sqrt{x}$	$F10^x$	$Fe^x$	Flg	Fln	Faresin	Farcos	Farctg	Fsin	Fcos	Ftg
	2	$F\pi$	$Fv$	$Fx^2$	$F\frac{1}{x}$	$Fx^y$	$F\bigcirc$									
	3															
	4	П0	П1	П2	П3	П4	П5	П6	П7	П8	П9	ПА	ПВ	ПО	ПД	П*
	5	С/П	БП	В/О	ПП	КНОП			$Fx \neq 0$	FL2	$Fx \geq 0$	FL3	FL1	$Fx < 0$	FLO	$Fx = 0$
	6	ИП0	ИП1	ИП2	ИП3	ИП4	ИП5	ИП6	ИП7	ИП8	ИП9	ИПА	ИПВ	ИПО	ИПД	ИП*
	7	$Kx=00$	$Kx=01$	$Kx=02$	$Kx=03$	$Kx=04$	$Kx=05$	$Kx=06$	$Kx=07$	$Kx=08$	$Kx=09$	$Kx=0A$	$Kx=0B$	$Kx=0C$	$Kx=0D$	$Kx=0F$
	8	КБП0	КБП1	КБП2	КБП3	КБП4	КБП5	КБП6	КБП7	КБП8	КБП9	КБПА	КБПВ	КБПО	КБПД	КБП*
	9	$Kx \geq 00$	$Kx \geq 01$	$Kx \geq 02$	$Kx \geq 03$	$Kx \geq 04$	$Kx \geq 05$	$Kx \geq 06$	$Kx \geq 07$	$Kx \geq 08$	$Kx \geq 09$	$Kx \geq 0A$	$Kx \geq 0B$	$Kx \geq 0C$	$Kx \geq 0D$	$Kx \geq 0F$
	-	КПП0	КПП1	КПП2	КПП3	КПП4	КПП5	КПП6	КПП7	КПП8	КПП9	КППА	КППВ	КППО	КППД	КПП*
	L	КЛ0	КЛ1	КЛ2	КЛ3	КЛ4	КЛ5	КЛ6	КЛ7	КЛ8	КЛ9	КЛА	КЛВ	КЛО	КЛД	КЛ*
	Г	$Kx < 00$	$Kx < 01$	$Kx < 02$	$Kx < 03$	$Kx < 04$	$Kx < 05$	$Kx < 06$	$Kx < 07$	$Kx < 08$	$Kx < 09$	$Kx < 0A$	$Kx < 0B$	$Kx < 0C$	$Kx < 0D$	$Kx < 0F$
	Г	КИП0	КИП1	КИП2	КИП3	КИП4	КИП5	КИП6	КИП7	КИП8	КИП9	КИПА	КИПВ	КИПО	КИПД	КИП*
	Е	$Kx = 00$	$Kx = 01$	$Kx = 02$	$Kx = 03$	$Kx = 04$	$Kx = 05$	$Kx = 06$	$Kx = 07$	$Kx = 08$	$Kx = 09$	$Kx = 0A$	$Kx = 0B$	$Kx = 0C$	$Kx = 0D$	$Kx = 0F$





## Вместо предисловия

Семейные чтения. Как давно это было! А теперь? Разве что семейные просмотры телепередач да редкие обмены репликами...

Конечно, авторы меньше всего надеются, что предлагаемая книга вернет давнюю традицию. Ведь книга о программируемых калькуляторах, самых первых представителях домашних компьютеров, которые, по мнению многих, как раз и посягают на духовную жизнь человека и человеческое общение.

И все-таки. Почему бы не предположить, что совместный интерес к «бездушной» машине поможет «одухотворить» семейный быт?

Во всяком случае авторам очень хотелось бы, чтобы эту книгу читали вечерами всей семьей. Она и разбита поэтому на «вечера», герои книги — члены одной семьи: папа, мама, сын и дочь. Позвольте Вас, читатель, познакомить с ними.

Глава семьи, Игорь Петрович, 40 лет, специалист по электронике. Постоянно работает с ЭВМ. Знает программирование. Калькулятор освоил давно и начал применять его в своей работе. Программы, с которыми он знакомит домочадцев, а вместе с ними и читателей, были опубликованы в журнале «Наука и жизнь».

Елена Николаевна, его жена. По специальности историк, в доме — умелая хозяйка. Прекрасно готовит, одинаково хорошо вяжет на спицах и на машине. О вычислительной технике знает лишь из телепередач и статей в популярных журналах.

Их дети:

Андрей, 14 лет, учится в 8 классе, охотно читает научно-популярные журналы, особенно любит «Науку и жизнь» и «Технику — молодежи». По мнению авторов, калькулятор из всех членов семьи нужнее всего ему.

Маша, 5 лет. Немного умеет читать, начала осваивать цифры, а вообще больше всего любит играть в куклы и смотреть «Спокойной ночи, малыши».

Представим вам и родителей Елены Николаевны. Они тоже действующие лица этой книги. Оба — пенсионеры.

Николай Иванович, дедушка, 70 лет, принимал участие в разработке первых советских ЭВМ типа БЭСМ-2 и «Стрела» и любовь к ЭВМ сохранил до сих пор.

Мария Петровна, бабушка, никогда не только с вычислительной, но и ни с какой другой техникой дела не имела.

Нетрудно догадаться, что все действие книги разворачивается вокруг микрокалькулятора, пожалуй, главного персонажа книги. И хотя будет назван конкретный тип программируемого калькулятора «Электроника МК-54», но он может быть заменен на «Электронику БЗ-34», «Электронику МК-56», а также на более новые модели «Электронику МК-52» и «Электронику МК-61».

На примере обычной, как говорят «средней», семьи авторы хотели показать, что калькулятор не какой-то сверхсложный прибор, общение с которым дело избранных. Совсем наоборот, это довольно простая и удобная вещь, которая может оказаться полезной человеку любого возраста и профессии. Потому что программируемый калькулятор — это не электронные счеты или автоматизированная логарифмическая линейка, назначение которых очень специальное. Калькулятор может быть и учителем арифметики для самых маленьких, и помощником при машинной вязке, и домашним бюро погоды, и портативной игротеккой и многим-многим другим. Причем все это размещается в маленьком футляре и сравнительно недорого стоит. По крайней мере, значительно дешевле телевизора и магнитофона — двух, пожалуй, самых распространенных бытовых электронных устройств.

Конечно, освоить калькулятор труднее, чем телевизор, но не намного. И опять же, смотря для чего. Чтобы использовать его для работ по готовым программам, достаточно, наверное, часа. Чтобы программировать самому, времени нужно больше, но зато насколько это интереснее!

Авторы не ставят своей целью научить читателя программированию — этому посвящен уже целый ряд книг; не пытаются и снабдить читателя универсальным сборником программ — такие книги тоже существуют (список литературы дан в конце книги). Основная цель — показать, как можно использовать калькулятор в быту.

В книге приведено более двух десятков программ: обучающих, игровых, бытовых, вычислительных. Если вас интересуют только сами программы, можно ограничиться чтением их текстов и инструкций. Если вас интересует, как работает стек калькулятора, что такое транслятор, как развивались алгоритмические языки или что такое структурное программирование, советуем прочесть «лекции» Игоря Петровича. Если же вы не боитесь посвятить книге пять вечеров, то прочитайте ее целиком.

Чтение это не требует никаких специальных знаний ни по математике, ни по вычислительной технике. Но и просто развлекательным его назвать нельзя. Всякое общение с вычислительной техникой неминуемо оказывает влияние на человека. Оно приучает к аккуратности, внимательности, четкости, а эти качества всегда очень нужны человеку.

Если же в результате знакомства с книгой читатель увлечется программированием, то цель, которую авторы ставили перед собой, будет достигнута. Читатель делает шаг к познанию мира компьютеров, мира, в котором всем нам предстоит жить в недалеком будущем.

Авторы выражают благодарность доценту Ю. В. Пухначеву, оказавшему при написании книги столь большую помощь, что его по праву можно считать одним из ее создателей.



## Вечер первый

### Рассказывает Андрей

Мы купили калькулятор. Что написано в инструкции. Маша учит цифры. Какие бывают учителя. Калькулятор учит считать в уме. Папа рассказывает о случайных числах.

Мы купили микрокалькулятор.

Вообще-то купил его папа. Но по нашему поручению. Потому что приобрести его решили мы все вместе — мама, папа, моя сестра Маша и я.

Мы всегда обсуждаем общие покупки. У нас даже очередь для них установлена. Калькулятор стоял за цветным телевизором, вязальной машиной для мамы и новым велосипедом для меня. Честно сказать, мне очень хочется новый велосипед. Надоело постоянно подкручивать гайки на старом, да и краска кое-где уже слезла. Но я добровольно отказался от велосипеда и проголосовал за калькулятор. Во-первых, калькулятор, как говорит папа, пригодится нам всем, а во-вторых, позавидовал я недавно Мишке Лихолетову. Он учится со мной в одном классе, 8 «Б», и у него уже есть калькулятор. Да к тому же не простой, а программируемый. Принес он его как-то в класс, поколдовал над ним минут пять, а потом предложил нам сыграть с калькулятором в «камешки». Есть такая игра. Мишка сказал, что по-настоящему она называется «Игра Ним». Нужно брать из кучи камешки, и кто возьмет последний — проиграл. Условие простое — брать каждый раз не меньше одного камешка и не больше определенного количества. Какого — это сам игрок решает перед первым ходом.

И вот стали мы играть — и каждый раз проигрывали. Сколько мы ни задавали камней в куче, и как ни ограничивали предельное число, которое можно брать за один раз, — все равно Мишкин калькулятор выигрывал. Потом он нас в крестики-нолики обыграл.

Когда нам проигрывать надоело, Мишка нам другую игру показал — «Гонки». В этой игре калькулятор — будто пульт управления гоночным автомобилем. Каждый ход состоит в том, чтобы задать скорость автомобиля и направление его движения. А калькулятор посчитает и покажет, насколько автомобиль вперед продвинулся и насколько его вбок ветром снесло: сбоку дует ветер, того и гляди к обочине прибьет. Тогда с этого места опять разгоняться нужно. Потому и направлять автомобиль приходится не прямо, а под углом к ветру, причем неясно, под



каким — ветер порывистый, то сильнее, то слабее. Гонки получаются азартные — кто быстрее к финишу придет\*

Загордился Мишка. У меня, говорит, еще таких игр сколько хотите есть, могу показать. А могу и новые придумать. Для этого надо уметь программировать. И показал нам программы — листки, исписанные какими-то иероглифами.

И вот теперь калькулятор у меня самого есть и тоже программируемый.

Вначале, правда, против его покупки возражала мама. Я помню некоторое время тому назад между родителями на кухне произошел разговор. Начала его мама.

— Не понимаю, зачем нам нужен калькулятор? — раздраженно говорила она папе. — То есть тебе-то он может быть и нужен для всяких там расчетов. Хотя тоже не возьму в толк, разве у тебя мало машин на работе?

Дело в том, что папа работает инженером в НИИ и постоянно проводит разные расчеты. На его письменном столе кучами лежат напечатанные на ЭВМ широкие бумажные ленты с дырками по краям, которые он называет то распечатками, то листингами. Часто вечерами папа рассматривает их, что-то там черкает, что-то записывает. . .

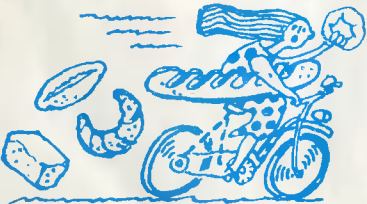
— Ну, положим, ЭВМ у нас на работе есть. Но она одна на весь отдел. Время работы на ней распределяют заранее, на неделю вперед. И не всегда это время удобно для меня. Это во-первых. Иногда же очень хочется провести небольшой расчет оперативно, а времени на ЭВМ нет. Калькулятор же будет всегда со мной. Это во-вторых. И наконец, в-третьих, некоторые задачи на маленьком калькуляторе можно решить быстрее, чем на большой ЭВМ. Вот, скажем, есть у тебя автомобиль и велосипед. На чем бы ты поехала на дачу?

— Конечно, на машине.

— А в булочную на углу?

— Я туда пешком иду.

— Но на велосипеде-то быстрее. — А на машине еще быстрее. — Не скажи. Пока откроешь гараж, выведешь машину, заведешь, потом ставь обратно, закрывай гараж. А на велосипеде — сел и поехал.



Похожая ситуация бывает и у меня. Решаю я, к примеру, какую-нибудь задачу и получаю ответ в виде громоздкой формулы. Нужно проверить, как ведет себя формула при определенных

сочетаниях параметров, график ее построить. Так чем писать программу, отлаживать ее — и все для одного расчета, лучше уж взять маленький калькулятор и через час получить нужный результат.

— Ну ладно, допустим. Я хоть и не очень разбираюсь в том, что ты говоришь, но мне понятно одно — калькулятор нужен тебе. Так и скажи!

— А разве тебе он не нужен?

— Мне? — Мама даже привстала из-за стола от изумления. — Ты, наверное, забыл, что моя профессия — историк. Вычислять мне нужно разве что, сколько лет правил Юлий Цезарь или, скажем, во сколько раз увеличилась империя греков при Александре Македонском. Прости, но это я довольно быстро делаю и в уме.

— А если что-нибудь посложней? Я уже не раз читал, как с помощью ЭВМ можно уточнять толкования исторических документов, оценивать степень причинно-следственных зависимостей между историческими событиями и даже моделировать их возможное течение.

— Я тоже об этом читала. Но ведь, наверное, такие расчеты твоему калькулятору не по зубам.

— Кое-что по зубам. Да и почему мы все время говорим о работе? Знаешь ли ты, что есть калькуляторные программы, по которым можно считать количество снятых или набираемых петель при машинном вязании? Что с помощью калькулятора можно придумывать узоры для свитера?

Мама заметно оживилась. Папа нащупал слабую струнку. Последнее время мама постоянно жаловалась, что ей надоело работать арифмометром при вязальной машине. Папа тем временем нанес следующий удар.

— А твои диеты? Расчеты лишних калорий? Это тоже можно поручить калькулятору.

Да, умеет отец находить аргументы.

— А посуду твой калькулятор мыть не умеет? — пробовала защищаться мама. — Может быть, его еще можно использовать в качестве полотера?

Но по ее тону было видно, что она сдается.

— Переубедить меня тебе, может быть, и удастся, но, кроме нас с тобой, есть еще дети.

Тут уже вмешался я и рассказал про Мишкин калькулятор.

— Вот, вот, — согласился папа, — с калькулятором можно просто играть.

— Что касается Андрея, я, пожалуй, спорить не буду. Калькулятор ему, действительно, может пригодиться. И не только для игр. — Голос мамы стал строже. — Теперь ведь информатику в школе проходят. Пусть привыкает к машинам. Но в чем уж ты меня не переубедишь, и здесь я буду непреклонна. Машу,

\* Авторам удалось выяснить, где опубликованы эти программы: «Ним» — в сборнике «Калькуляторы в играх и задачах» (М.: Наука, 1986), а «Гонки» — в книге И. Данилова «Секреты программируемого калькулятора» (М.: Наука, 1986).

пожалуйста, к калькулятору не приучай. Девочка пока не только считать не умеет, но и цифр-то толком не знает. И если она привыкнет к машинке, то потом дважды два в уме сосчитать не сможет. А я не собираюсь превращать свою дочь в тупой придаток к машине и тебе этого не позволю.

К моему удивлению, папа сразу согласился.

— Ты права. Конечно, девочке сначала надо научиться считать в уме.

Но почему в этом ей не может помочь калькулятор? — продолжил папа.

— Как это?

— Да очень просто. Я уже приготовил программы, с помощью которых ребенок Машиных лет и к цифрам привыкает и считать в уме учится.

Чем окончился разговор, я не знаю, потому что в это время родители вспомнили обо мне.

— А ты, Андрей, что — спать не собираешься? — строго спросила мама. Я успел только сказать, что готов подождать с велосипедом, и отправился в свою комнату.

Калькулятор выбирал, конечно, папа.

Им оказался программируемый калькулятор «Электроника МК-54». С одной стороны, этот микрокалькулятор — самый дешевый из всех программируемых, с другой — устройство проверенное и надежное, а главное — для него без всяких изменений годятся программы, опубликованные в разных популярных журналах, да и книжек про него (или про похожую на него модель «Электроника БЗ-34») выпущено уже немало.

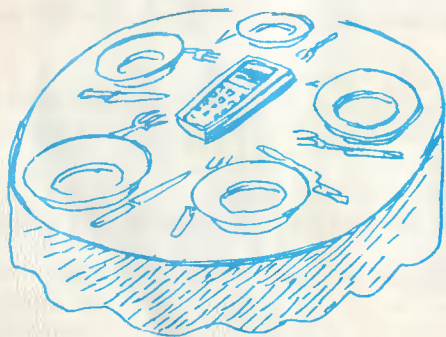
И вот вечером мы собрались вокруг стола, место праздничного пирога на котором в этот раз занял калькулятор.

Начали мы с изучения инструкции к машине. Называется она «Руководство по эксплуатации». И краткое описание калькулятора там есть, и полное. Жаль только оглавления нет, поэтому найти то, что нужно, бывает трудновато.

Первой инструкцию взяла мама и, раскрыв ее, строго спросила папу: «А ты при покупке калькулятора попросил проверить его

работоспособность по контрольному тесту? А сколько времени у тебя заняло вычисление теста 85?»

Папа вздохнул. И было от чего. В табл. 1 приведен ни много ни мало... 101 тест!



— Видишь ли, — ответил папа, — по-моему, эти тесты никто не проверяет. Это все равно, что при покупке радиоприемника проверять, как он ловит все станции.

— А зачем тогда эта таблица?

— Не знаю. Скорее всего для проформы. А как тебе нравится вот это. — Папа взял инструкцию и зачитал: «Пункт 4.2.12. Для расширения возможностей в области программирования, а также удобства контроля и отладки программ в микрокалькуляторе предусмотрены: а) специальная память объемом в 98 шагов программы. . .»

— Но ведь это правильно, — возразил я.

— Правильно? Даже, если оставить на совести авторов стиль изложения, то зачем это? Не написано же в инструкции к твоему велосипеду, что для катания у него предусмотрены колеса. Без «специальной области памяти» на калькуляторе просто нельзя программировать!

Разбор инструкции был окончен неожиданно.

— По-моему, ясно, — сказала мама. — Инструкция не то произведение, которое стоит читать вслух. Хотя, конечно, жаль. Поэтому пусть каждый попытается прочесть ее сам, если сможет, а ты Расскажи нам своими словами, что и как мы будем делать с этой «игрушкой».

— Играть, — вставила Маша. До этого она только переводила взгляд с одного говорящего на другого.

— Конечно, играть мы тоже будем, — поддержал ее папа. Но сначала давайте посмотрим, как работает калькулятор по программе. А чтобы и Маше было интересно, начнем с программы специально для нее. Пусть девочка освоит цифры, а потом и считать научится. Хочешь, Маша?

— Я играть хочу.

— А это и будет игра, игра в цифры. Кстати, Андрей, думаю и тебе это будет интересно.

— Мне? — Я удивленно хмыкнул. — Что я цифр не знаю?

— Знаешь, конечно. Я имел в виду другое. Интересно и полезно тебе будет научиться работать с готовой программой.

— Вот и сама программа, — и папа положил на стол листок бумаги, исписанный иероглифами, как на Мишкиных программах.

#### ПОВТОРИ ЦИФРУ

	0	1	2	3	4	5	6	7	8	9
0	+	÷	ПД	1	0	↑	9	÷	ПВ	ИПД
1	1	1	х	Пπ	+	ПД	КИПД	ХУ	ИПД	—
2	ПД	9	х	1	+	ПС	КИПС	ИПС	С/П	—
3	Пх≠0	36	ВП	,	БП	27	ИПС	ИПВ	х	С/П
4	БП	09								



## Инструкция\*

1. число ↑ час ↑ мин В/О.

2. С/П «цифра». Нажать клавишу с той же цифрой С/П.

3. При правильном ответе: «цифра цифра цифра... цифра». Перейти к п. 2 (новое задание). При ошибке: «ЕГГОГ». Перейти к п. 2 (повторение задания).

— Прежде всего, — сказал папа, — программу нужно ввести в память нашей ЭВМ, нажимая на клавиши, названия которых записаны в тексте программы. Обрати внимание, что некоторые символы написаны прямо на клавишах. Для ввода команд, которые они означают, достаточно нажать саму клавишу. Если же символ команды написан над клавишей, то для ввода такой команды сначала нажимают клавишу «F» — вот она, в левом верхнем углу клавиатуры, а потом уже нужную клавишу.

— В общем, как верхний регистр на пишущей машинке, который нажимаешь, когда надо печатать заглавные буквы? — спросила мама.

— Совершенно верно. Только тут есть существенное отличие. Клавишу верхнего регистра можно поставить на фиксатор, и тогда каждая печатаемая далее буква будет заглавной. Клавиша «F» на микрокалькуляторе работает иначе. Она оказывает действие только на одну клавишу, нажатую после нее. Для следующей «надклавишной» команды надо снова нажимать клавишу «F».

И еще одно замечание. Некоторые обозначения в моих текстах отличаются от символов на клавишах. По установившейся традиции (так пишут программы в журналах «Наука и жизнь», «Химия и жизнь», «Техника — молодежи») я буду употреблять названия команд, которые использовались на модели «БЗ-34». Разночтений это внесет немного. Вот они: П вместо X → П, ИП вместо П → X, ↑ вместо В ↑. Мне они кажутся более простыми и наглядными, так же, как и привычные названия обратных тригонометрических функций:  $\arcsin$ ,  $\arccos$ , и  $\arctg$  вместо двусмысленных  $\sin^{-1}$ ,  $\cos^{-1}$  и  $tg^{-1}$ .

Итак, начали.

Папа отложил листок с программой и взял в руки калькулятор.

— Сейчас я его включу вот этим переключателем (он прикоснулся к левой кнопке под прозрачным окошечком на панели калькулятора) и на индикаторе (он показал на это окошечко) загорится нуль. Его появление означает, что наша карманная ЭВМ готова к работе.

Папа включил микрокалькулятор и в левом углу индикатора тотчас ярко загорелся нуль. Затем медленно, чтобы мы видели и запоминали, папа нажал клавиши «F» и «ПРГ». Нуль в левом

углу индикатора исчез и вместо него в правом углу вспыхнули два нуля.

— Теперь все готово для ввода программы. Будь внимателен, Андрей. Когда станешь нажимать клавиши, обрати внимание: в левой части индикатора будет появляться пара символов. Это код введенной команды, имя, под которым она известна калькулятору. В правом же углу вместо 00 станут появляться пары цифр 01, 02 и так далее. Это адреса ячеек памяти, куда будут последовательно записываться коды команд. К примеру, два нуля, которые сейчас светятся, — это адрес той ячейки, куда попадет первая из введенных команд. Кстати, обращай внимание на эти адреса. Это довольно простой, хотя и несколько поверхностный, способ проверки — не пропустил ли ты какую-нибудь команду или не внес ли лишнюю. Для более тщательной проверки желательно пользоваться специальной таблицей, где приведены коды команд\*.

Ну что, попробуешь? — предложил папа.

Я начал вводить программу. Сначала дело пошло довольно бойко, но вдруг я заметил, что после набора команды Fл с кодом 20 в правом углу индикатора появилась цифра 13, т. е. команда попала по адресу 12, вместо положенного 13. Я вопросительно посмотрел на папу.

— Наконец-то заметил, — сказал он. — Ищи теперь, какая команда пропущена. Подскажу: ошибка допущена недавно и проще всего ее найти, двигаясь по программе назад. Для этого надо нажимать клавишу ШГ со стрелкой влево (шаг назад) и следить за кодами в левой части индикатора. Действуй!

Нажав несколько раз эту клавишу, я, наконец, нашел, что по адресу 11 у меня записан код операции умножения 12, должна же быть единица.

— Нашел? — спросил отец. — Теперь вноси пропущенную команду и продолжай набор. Только впредь будь внимательнее, а то мы до глубокой ночи не закончим.

Набирал я программу минут двадцать. Несколько раз приходилось возвращаться и вставлять пропущенные команды, а один раз пришлось вычеркнуть лишнюю.

Наконец, программа введена и я вздохнул с облегчением.

— Вообще-то, сказать с уверенностью, что программа введена правильно, пока еще нельзя, — умерил мою радость папа. — Даже опытному программисту случается нажать не на ту клавишу. Поэтому после ввода новой программы полагается проверять ее правильность. Для этого решают специальный контрольный пример с заданным набором исходных данных, для которых заранее известен результат. На первый раз обойдемся без этого. Я внимательно следил за тобой и могу ут-

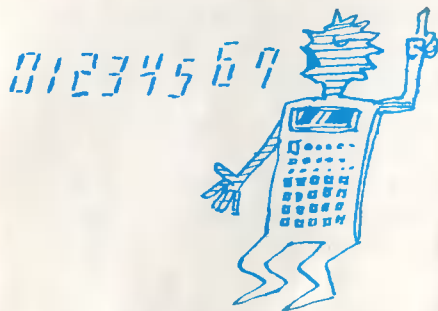
\* Все программы снабжены краткими инструкциями, в которых описаны действия по вводу исходных данных и порядок получения выходных результатов. Вводимые данные выделены полужирным шрифтом, а данные, выводимые на индикатор, заключены в кавычки.

\* Таблица кодов команд приведена на 3-й странице обложки.



верждать, что ошибок в программе больше нет. Можешь начинать работу.

Прежде всего переведем калькулятор в режим вычислений. Нажми клавиши F и АВТ.



В левом углу индикатора опять появился одинокий ноль. — Ну что, — спросил я, — отдать теперь калькулятор Маше?

— Нет, рано. Дело в том, что, хоть мы и говорим о нашем калькуляторе, как об ЭВМ, это скорее «полуЭВМ». Настоящая ЭВМ после вызова определенной программы обычно сама подсказывает, как с ней работать, что и когда вводить, на

какие клавиши нажимать... А наш калькулятор умеет говорить только цифрами. Сам понимаешь, что это язык довольно бедный. Ты спросишь, как же восполнить этот недостаток микрокалькулятора? Как получить полноценный вычислительный комплекс? Только рассматривая пользователя как его составную часть. Отсюда и необходимость параллельно с программой для калькулятора писать программу и для «думающей» части нашей ЭВМ, то есть человека. Такая программа называется инструкцией. Выглядеть она может, например, так.

Папа протянул мне листочек бумаги.

#### Инструкция\*

1. Ввести программу.
2. Перейти в режим вычислений.

3. Ввести три числа: а ↑ о ↑ с В/О С/П

(а — число месяца, в, с — текущее время: часы и минуты).

4. После появления какой-либо цифры на индикаторе нажать клавишу с той же цифрой и затем С/П.

5. Если ответ правильный — на индикаторе восьмизначное число, состоящее из тех же цифр, если сделана ошибка — сообщение ЕГГОГ (это слово переводится как ошибка) и повторение задания.

6. Продолжить работу после правильного ответа: нажать С/П и перейти к п. 4.

— Понятно? — спросил папа.

— Понятно, — ответил я. — Так как два первых пункта уже выполнены, перехожу сразу к третьему.

Я посмотрел на часы и набрал: 11 ↑ 19 ↑ 47 В/О С/П.

\* Если вы сравните эту инструкцию с той, что помещена непосредственно под текстом программы, то обратите внимание, что она более подробная. Краткие инструкции адресованы тем, кто хочет воспользоваться лишь текстом программ. Для тех кто собирается прочесть всю книгу, в некоторых случаях приводятся расширенные инструкции.

— Вот теперь отдай калькулятор Маше. Ну-ка, Маша, давай посрамим недоверчивых. Начиная привыкать к цифрам и к клавишам. Когда увидишь цифру в этом окошечке, ищи, на какой клавише она нарисована, и нажимай эту клавишу.

На индикаторе показалась девятка. Маша довольно быстро нашла клавишу 9 и нажала ее. Ничего не произошло. Маша недоуменно посмотрела на папу.

— Подскажи сестренке, что делать дальше, — обратился отец ко мне.

Я посмотрел в инструкцию и сказал: «Нажимай теперь клавишу С/П». Не дождавшись, пока она найдет эту клавишу, я нажал ее сам. Индикатор мгновенно погас, а потом на какие-то едва уловимые доли секунды на нем стали вспыхивать числа — то длинные, то короткие. Прошло несколько секунд, и индикатор заполнился девятками.

— Здорово, — пропела Маша, — но я сама хочу.

— Запомнила, где та клавиша, которой запускается машинка?

— Вот она, — сказала Маша.

— Правильно. Нажимай ее снова.

На индикаторе появилась восьмерка, и Маша нашла ее почти сразу. Игра ей начала определенно нравиться. Честно говоря, мне тоже. К тому же я был горд тем, что сам набрал программу.

Следующие цифры Маша тоже нашла быстро. Застопорилось дело с единицей, ее сестра упорно путала с семеркой. Наконец, единица была найдена, и все пошло дальше гладко.

Конечно, ничего трудного для меня в этой игре не было. Но тем не менее было в ней что-то такое, что заставляло задуматься. Во-первых, зачем понадобилось вводить время, если оно нигде не использовалось. Во-вторых, почему калькулятор так долго придумывал каждое новое число? В-третьих, как он ухитрялся придумывать задания таким образом, что я не в состоянии был угадать, какое число будет следующим. Начал он с девятки, потом показал восьмерку. Я решил было, что теперь будет семерка, но на экране появилась шестерка. Двойка появилась потом два раза подряд, а семерку пришлось ждать довольно долго.

Пока Маша возилась с цифрами, я решил задать эти вопросы папе.

— Молодец, что заметил. Вот видишь, — обратился он к маме, — калькулятор не только дочь считать учит, но и старшего заставляет думать. — А затем продолжил разговор со мной. — Чтобы ответить на твои вопросы, нужно поговорить о случайных числах. Тема это очень интересная, но Маше пока непонятная. Оставим ее для другого раза, а пока давай дадим Маше новое задание. С цифрами она немного разобралась, завтра с твоей помощью поиграет еще, а пока — новая программа. Она поможет Маше освоить многозначные числа.



— А что это такое? — спросила Маша.

— Сейчас увидишь. Как и предыдущая, эта программа тоже предназначена для обучения. Но она гораздо интереснее, так как не только предлагает задания, но и, как опытный педагог, варьирует темп обучения. Правильно отвечает «ученик», задания усложняются, стал он ошибаться, усложнения не происходит, а иногда, наоборот, задания даже упрощаются. Так что сейчас мы познакомимся с электронным учителем.

— Интересно, — вступила в разговор мама. — Мне всегда казалось, что для обучения прежде всего нужен такт, дружелюбие, индивидуальный подход, терпение. Да мало ли что еще! Кстати, я как-то прочитала, что двум категориям людей компьютеры не нужны: директорам и детям. Директорам потому, что у них и живых помощников хватает, а детям — по причинам, о которых я упомянула. Хотя мы и говорили недавно на эту тему, возвращусь к ней опять. Ты меня почти убедил, что в придаток машины наша дочь не превратится. Но как подействует на нее замена живого общения игрой с этим бездушным автоматом?

— Ну зачем так сгущать краски. Никто этого общения ее лишать не собирается. Ты сама употребила слово «игра». А если во время игры она кое-какие знания приобретет, так что же в этом плохого? Что же касается обучения с помощью компьютера, могу привести пример из одного английского журнала. Там я прочел интересную статью, которая так и называлась: «Может ли компьютер учить?» Автор, сам преподаватель английского, рассматривает два типа учителей. Тип первый — «профессор». Это волевой сильный человек, сам выбирающий метод и темп обучения, требующий от учеников беспрекословного подчинения и не гнушающийся линейкой, которой он в воспитательных целях лупит по рукам нерадивых учеников.



Второй тип — «педагог». Прообраз его — греческий раб времен покорения Древним Римом Греции. Эти рабы были намного образованнее своих хозяев, поэтому часто использовались для

обучения юношей из патрицианских семей. Раб-учитель постоянно сопровождал своего молодого господина и был готов ответить на любой его вопрос. Учил он только тому, что интересовало ученика. Педагог мог давать задания, правда, ученик не обязан был их выполнять.

Конечно, два этих типа представляют собой полюсы, между которыми в действительности находятся методы преподавания реальных учителей. Кто-то ближе находится к одному полюсу, кто-то — к другому. Когда компьютерное обучение стало реальностью, первые обучающие машины были подобны «профессорам» — учащийся должен был безоговорочно следовать жестко запрограммированной системе заданий, выдаваемых машиной. Кроме того, создатели первых обучающих программ исходили из порочного принципа: знания, дескать, можно раздробить на отдельные независимые порции и помаленьку кормить ими учеников. Большинство программ, составленных по подобным схемам, представляли собой набор текстов, появляющихся на экране. После текста следовали вопросы. Если ученик отвечал на них правильно, то ему показывали следующую картинку. При ошибке повторяли одну из предыдущих. Такие программы были остроумно названы «перелистывателями страниц».

Сегодня обучение с помощью компьютера немыслимо без участия учителя. Компьютер для него служит подспорьем, оживляя занятия и освобождая от скучных упражнений, таких, например, как придумывание однотипных примеров, вычерчивание сложных схем и тому подобное. В качестве такого подспорья можно использовать и программу, которую я написал.

Папа обернулся ко мне:

— Андрей, занимай свое место за «пультом» и вводи текст. Он дал мне новую программу и инструкцию к ней.

#### ПОВТОРИ ЧИСЛО

	0	1	2	3	4	5	6	7	8	9
0	F√	Fg	ПД	Cx	П5	Cx	П4	ИПД	F10 <sup>x</sup>	ПД
1	↑	КИПД	FO	ИПД	—	ПД	FO	ИП5	F10 <sup>x</sup>	x
2	ПС	КИПС	ИПС	С/П	—	Fx=0	36	ИП4	Fx=0	33
3	КИП4	БП	07	КИП5	БП	05	ИП4	Fx=0	46	ИП5
4	Fx≠0	05	1	—	БП	04	Cx	П4	БП	22

#### Инструкция

1. Нецелое число от 11 до 99 В/О С/П.
2. «число». Набрать цифры числа на клавиатуре С/П. При правильном ответе сложность задания — количество цифр выводимого числа — возрастает, при ошибке автоматически задание облегчается. В обоих случаях продолжить по п. 2.



Не могу сказать, что ввод новой программы прошел намного быстрее предыдущей. Но вопросов я задавал меньше, ошибки находил быстрее, хотя новая программа была значительно длиннее предыдущей. К тому же через несколько минут после начала ввода папу позвали к телефону и вся ответственность за правильность ввода легла на меня. Следили за мной только мама с Машей.

Закончив ввод, я передал калькулятор Маше и приготовился объяснять, как работать с программой. Папа в это время закончил свой разговор и вернулся.

— А ты уверен, что программа введена правильно? — спросил он.

— Вообще-то я следил за индикатором, как ты учил.

— И тем не менее давай привыкать к культурной работе с калькулятором. Раз полагается пропустить контрольный пример перед началом работы, давай так и сделаем. Вот текст контрольного примера: 73, 4638 В/О С/П «8» 8 С/П «3» 3 С/П «53» 55 С/П «2» 2 С/П «1» 1 С/П «11».

— Понятно, что нужно делать?

— Пока не очень. Зачем некоторые числа написаны в кавычках, а другие без них?

— Без кавычек написано то, что ты должен набирать сам. В кавычках — то, что появится на индикаторе.

— А почему этих чисел так много? Разве одного недостаточно?

— В нашей программе — нет. Нужно ведь убедиться, что программа работает правильно как при верном твоим ответе, так и при неверном. Когда ты получишь познакомиться с программированием, то узнаешь, что есть программы, состоящие из нескольких ветвей. Это как развилки на железной дороге. Путей, по которым может пойти вычислительный процесс, получается много. Естественно, необходимо убедиться, что программа ни на одном из этих путей не зайдет в тупик. Для этого и используют несколько заданий, не меньше, во всяком случае, чем ветвей в программе. Кроме того, в этой программе, как и в предыдущей, впрочем, работает датчик псевдослучайных чисел. Что это такое, мы поговорим после. Нужно проверить и его работу. Вот я и написал контрольный пример, состоящий из нескольких заданий.

Папины опасения оказались не напрасны. После того как я запустил калькулятор, он сразу же выдал не то.

— Не расстраивайся, — успокоил меня отец, — ошибки при вводе дело довольно распространенное. Для того и используют контрольный пример. Придется проверить всю программу. Переводи калькулятор в режим программирования. Не забудь только сначала нажать клавишу В/О, чтобы выйти на начало программы, и просматривай потом ее текст, раз за разом нажимая клавишу ШГ со стрелкой вправо.

Работа оказалась достаточно нудной. Может, правда, мне так показалось с непривычки. Наконец, я обнаружил ошибку: по адресу 26 у меня было записано 33 вместо 36. Я отступил на шаг назад, нажав ШГ со стрелкой влево, и приготовился исправлять.

— Подожди, подожди, — остановил меня папа. — Сейчас новых ошибок наделаешь.

— Но ты же сам учил исправлять ошибки именно таким способом.

— Да, если речь не идет об адресах перехода. Они выражаются двузначным числом. А две цифровые клавиши, нажатые подряд, воспринимаются как один код только после команд перехода. Если ты нажмешь их сейчас, то внесешь в две смежные ячейки коды цифр 03 и 06.

— Что же делать? Перебирать все с начала? — огорчился я.

— Все не надо. Достаточно перебрать одну команду перехода. Тогда и адрес калькулятор воспримет правильно. Так что отступай еще на шаг назад и повтори команду  $Fx = 0$ , а затем уже новый адрес.

Я исправил ошибку и приготовился вновь перевести калькулятор в режим вычислений, но папа опять остановил меня.

— Не торопись. Учти, что одна из заповедей программиста гласит: «Не поддавайтесь искушению повторить счет после первой же найденной ошибки. Проверяйте программу до конца и старайтесь выловить все ошибки».

И опять он оказался прав. Через несколько шагов я обнаружил еще одну ошибку, потом еще одну. Наконец, вся программа была проверена. Я переключил режим, нажав клавиши F, АВТ, и снова проверил работу программы по контрольному тесту. На этот раз все было так, как написано в инструкции.

— Вот теперь можешь отдавать калькулятор.

Я передал калькулятор сестре и объяснил ей, какие клавиши нажимать и когда. Маша углубилась в работу. Игра ей явно нравилась. Через пару минут она оторвалась от работы и спросила: «А как называется число, у которого цифры один, ноль, пять?».

— Сто пять, — сказал я.

— А это очень много? — вновь спросила она.

— Да как тебе сказать...

— Ну хватит, — вмешалась мама. — Совсем девочке голову разматывают. Она еще цифр-то как следует не знает, а вы ей трехзначные числа.

— Ты, пожалуй, права, — ответил папа. Эта программа, наверное, лучше подошла бы для школьника. Я как-то не подумал, что большие числа для ребенка не очень понятны.

— Ну вот, я так и знала. Андрею все, а мне ничего, — вздохнула Маша.



## СОСЧИТАЙ В УМЕ

	0	1	2	3	4	5	6	7	В	9
0	$\sqrt{\quad}$	Flg	ПД	Сх	п5	п6	пп	65	ип0	1
1	—	x	1	+	пА	киПА	пп	65	ип0	ипА
2	—	x	1	+	пВ	кипВ	ипА	ипВ	—	Fx<0
3	36	FBx	ипА	пВ	хУ	пА	ипА	ипВ	кпп9	п1
4	ипА	ипВ	ип0	÷	+	п2	С/п	ип1	—	Fx=0
5	59	кип5	5	Вп	5	5	С/п	Бп	06	кип6
6	Вп	—	ип2	Бп	46	ипД	F10 <sup>x</sup>	пД	кипД	хУ
7	ипД	—	пД	В/0	+	В/0	—	В/0	х	В/0

## Инструкция

1. m П9 п П0 (m — код арифметического действия: 74 — сложение, 76 — вычитание, 78 — умножение; п — максимальная величина числа в задании).

2. Нецелое число от 11 до 99 В/О.

3. С/П «А,В» (А и В — числа, над которыми нужно провести арифметическую операцию).

4. Ответ С/П. При правильном ответе: «5, 55», перейти к п. 3. При ошибке: «ЕГГОГ», перейти к п. 3 (повторение задания).

Примечания: 1. После появления сообщения «5, 55» можно изменить m и п, выполнив п. 1. Затем перейти к п. 3.

2. Количество правильных ответов можно вывести на индикатор командой ИП5. По команде ИП6 выводится число допущенных ошибок. Новый сеанс работы с обнулением счетчиков следует начать с выполнения п. 2.

3. Заключительные нули во втором выводимом числе, В, не появляются. Например, при  $p=1000$ , появление на месте числа В цифры 3 означает число 300, а цифр 14 — число 140. Числа 3 и 14 появятся в виде 003 и 014.

— Не расстраивайся, — успокоил ее отец. — Андрей тебе завтра новую программу покажет. Она будет учить тебя арифметике. А на часы-то ты посмотрела?

Вот это да! Было уже девять часов вечера. Маша даже забыла про «Спокойной ночи, малыши»!

Мы остались с папой вдвоем.

— Ну как тебе машина? — спросил он.

— Здорово. Но почти ничего непонятно. Вопросов у меня накопилась тьма.

— Это хорошо. На вопросы твои я потом отвечу обязательно. Многие я, пожалуй, представляю, другие задашь сам. Но сначала, так как время позднее, давай я тебе покажу еще одну программу. Завтра ты будешь развлекать ею Машу. Это программа из всех показанных сегодня, пожалуй, самая «умная». Она не только задумывает числа, но и предлагает выполнить над ними какое-нибудь арифметическое действие: сложение, вычитание или умножение, и сообщить ей ответ. Затем она проверяет правильность ответа и дает новое задание.

— Если я правильно понял, калькулятор учит считать в уме, то есть, без калькулятора?

— Совершенно верно. Это как раз был мой козырь в споре с мамой, когда я убеждал ее в полезности калькулятора для Маши.

Инструкция у меня получилась несколько громоздкой, поэтому я решил изложить ее тебе неформально. Итак, слушай и записывай основные моменты, чтобы ничего не забыть. Ведь тебе нужно научить работать с программой Машу. Излагать инструкцию я буду на живом примере.

Сначала нужно задать предел изменения чисел. Давай зададим 100. Это будет самое большое число, которое калькулятор покажет при вычитании, или предел суммы при сложении, или предел произведения при умножении. Вводим это число в регистр 0.

И папа набрал 100 П0.

— Затем нужно закодировать действие, которому калькулятор будет обучать. Каждая арифметическая операция имеет в программе свой код: сложение — 74, вычитание — 76 и умножение — 78.

С чего начнем?

— Давай со сложения.

— Хорошо. Вводим 74 П9, то есть, записываем код в регистр 9. Теперь осталось ввести произвольное число, с которого начнется игра. Это может быть любое нецелое число от 11 до 99. После ввода его останется нажать В/О С/П и калькулятор будет готов к работе. Начали?

Он смотрел на меня выжидающе, а я не знал, что делать. Какое число вводить? Когда папа просил ввести число месяца и время, мне было понятно. Посмотрел на часы и вводи. А тут просто любое. Я задумался.

— Чего же ты ждешь? — спросил папа.

Я объяснил и тут же спросил у него: — Если мне трудно придумать одно любое число, то каково калькулятору каждый раз их выдумывать?

Папа улыбнулся. — За калькулятор не беспокойся. Не так уж ему трудно. Тем более, что всю работу я за него выполнил.

— Ты что, записал все числа, которые он будет предлагать?

— Ну уж нет. Это непосильная работа. Да и ненужная. В таких случаях записывают не сами числа, а алгоритм, с помощью которого получаются случайные числа.

— А разве есть такие алгоритмы? Что-то я не понимаю. Ведь любой алгоритм будет давать определенные числа, а случай на то и случай, что его нельзя запрограммировать. Правильно я понимаю?

— Правильно, никакой алгоритм не может дать случайные числа. Но для нас подойдут и числа, которые, как бы попроще сказать, почти случайные, что-ли.



— Ничего, не понимаю.

— Ладно, раз это тебя заинтересовало и я к тому же обещаю тебе рассказать о случайных числах, придется прочесть тебе специальную лекцию. Пока мама нас не отправила спать, давай начнем.

## О случайных числах

Прежде всего, что такое случайное число? Вот, скажем, числа 2, 3 и 4. Можно ли сказать, что они случайны?

И да и нет.

Да, если это очки, выпавшие при трех бросаниях игрального кубика. Нет, если это, скажем, номера страниц книги, которую ты читаешь.

В чем же дело? Во втором случае мы знаем, что следующее число, номер следующей страницы должен быть 5. В первом же случае мы не знаем, сколько очков покажет кубик следующий раз.

Отсюда вывод — случайны не числа сами по себе, взятые в отдельности, а их последовательность. Поэтому-то обычно говорят именно о последовательности случайных чисел.

Из первого примера видно, кстати, как их можно получить. Бросание кубика обеспечит нам, как говорят математики, генератор случайных чисел, нормированный от 1 до 6. Появление любого числа в этом диапазоне будет равновероятно. Можно представить себе «кубик», нормированный и в другие пределы. Треугольная пирамида, например, обеспечит нам числа от 1 до 4. Октаэдр — от 1 до 8. Самый простой «кубик» — двусторонняя монетка. Она дает ряд, где встречаются лишь два числа: 0 — орел и 1 — решка.

Итак, последовательность случайных чисел — это набор, характерный тем, что никогда нельзя угадать, какое число будет в нем следующим.

Другой вопрос, зачем нужны такие наборы?

В быту они чаще всего используются в играх, внося элемент случайности в выбор очередного хода. В нардах, например, бросаются два кубика, и в зависимости от выпавшего на них числа очков игрок делает тот или иной ход. Да и ты часто играешь с Машей в разные игры, где число клеток на которые вы продвигаете свои фишки, зависит от выпавшего на кубике числа очков.

Но использование этих чисел не исчерпывается играми.

С их помощью решаются довольно сложные математические задачи, изучающие поведение сложных систем, зависящих от многих, часто случайных, факторов.

В этих ситуациях очень важно, что случайные числа, вырабатываемые генератором, равномерно распределяются в заданном диапазоне. То есть появление любого из них равновероятно.

Вот пример их использования.

Представь себе, что нужно посчитать площадь, ограниченную довольно сложной кривой. Один из способов как раз и предполагает использование случайных чисел.

Анализируемая фигура заключена в прямоугольник. Если теперь с помощью генератора случайных чисел расставить точки, то они равномерно покроют всю площадь. (Вот для чего нужна равномерность распределения.) Достаточно потом сосчитать долю точек, попавших внутрь фигуры, и взяв такую же долю от площади прямоугольника. Это и будет площадь искомой фигуры.

Случайные числа необходимы и при решении целого ряда задач физики, химии, при оценках достоверности экспериментальных данных, полученных в результате различных процессов...

Поэтому вопрос получения случайных чисел очень важный. Но если эти числа — результат какого-либо процесса, то как их получать на ЭВМ?

Раньше, правда, пытались использовать случайные процессы, проходящие в недрах ЭВМ, но это было давно, когда ЭВМ строились не на интегральных схемах, как теперь, а на вакуумных лампах. Но и тогда такой способ широкого распространения не получил по целому ряду причин. Одна из них была на первый взгляд парадоксальной. Оказалось, что программистам работать с «настоящими» случайными числами неудобно. Представь себе, нужно отлаживать программу, изучать ее отклик на заданный сигнал, определенную исходную информацию, а сигнал этот меняется от одного запуска программы к другому. Лучше, когда сигнал, точнее сказать, последовательность сигналов заранее известна.

Но тогда о каких случайных числах может быть речь? Это во-первых. А во-вторых, получать случайные числа, как мы выяснили, тоже задача непосильная.

Эти причины привели к идее создания генераторов «псевдослучайных» чисел. Слово «псевдо» греческого происхождения и означает мнимый, ложный. Математики, правда, трактуют его несколько иначе, как «почти».

Чем же отличаются псевдослучайные числа от случайных?

Ряд случайных чисел не имеет периода. Это означает, что сколько бы мы ни получили случайных чисел, угадать следующие невозможно. Ряд псевдослучайных чисел всегда имеет период. Следовательно, в принципе можно, выписав все члены последовательности в рамках периода, точно указать будущие члены ряда. Кроме того, если ряд случайных чисел никогда нельзя повторить, то ряд псевдослучайных чисел возможно повторять сколько угодно раз.

Это свойство, как ты понимаешь, очень удобно для отладки программ. Что же касается периодичности ряда, то, если длина неповторяющейся последовательности очень велика — тысячи и десятки тысяч чисел, для большинства практических задач этот ряд можно рассматривать как бесконечный.

Теперь давай поговорим о том, как получать псевдослучайные числа. Алгоритмы, которые для этого используются, называются генераторами псевдослучайных чисел. Результатом работы этих алгоритмов является набор чисел, равномерно распределенных в заданном интервале. Почему обязательно равномерно? Да чтобы появление любого числа было равновероятно.

К таким алгоритмам относится, например, расчет по формуле

$$r_{i+1} = \{r_i \cdot 11 + \pi\}.$$

Фигурные скобки здесь означают дробную часть числа,  $r_i$  и  $r_{i+1}$  — последовательные члены ряда.

Кстати, формулы, подобные написанной, называются рекуррентными. Они дают зависимость последующего числа от предыдущего. Именно этот генератор использован в моих программах.

Простая формула, лежащая в его основе, дает неповторяющийся ряд чисел, расположенных между нулем и единицей, длиной в несколько тысяч элементов. Произведя с числами несложное преобразование, можно получить новый набор, каждый элемент которого будет расположен в любом другом диапазоне. В общем случае, если мы хотим получить числа в пределах от  $a$  до  $b$ , то алгоритм дополняется формулой

$$R_i = a + (b - a)r_i.$$

Подобные преобразования также имеются в программах, рассмотренных нами сегодня. Думаю, что теперь тебе понятно, зачем мы вводим перед началом работы программы дату или любое число. Таким образом мы задаем первый член ряда, или, как говорят, запускаем генератор.

Прослушав «лекцию», я в душе позавидовал машине, которой не надо думать над случайными числами, и продолжил знакомство с программой.

Набрав на клавиатуре 65.4321 (самое случайное число, ко-



торое я смог придумать), я нажал В/О С/П и секунд через двадцать на индикаторе появилось число 21.09.

— Дробное число, которое появляется на экране, нужно рассматривать здесь как пару целых чисел: 21 и 9, — объяснил папа. — Задание состоит в том, чтобы сложить их и набрать на клавиатуре их сумму.

Я быстро набрал 30.

— Теперь, как обычно, С/П, — подсказал папа.

Нажав эту клавишу, я через несколько секунд с радостью увидел на индикаторе оценку своего труда: 5. 55. Я расценил это как три пятерки сразу. Потом я нажал клавишу С/П и прочитал новое задание: 79,28. Естественно, захотелось убедиться, правильно ли считает калькулятор. Я ввел заведомо неправильный ответ: 15 С/П. На мою ошибку калькулятор отреагировал сообщением ERROR. Я снова нажал С/П, и задание повторилось.

— Так будет до тех пор, пока ты не наберешь правильный ответ, — пояснил папа. — Только после этого загорится новое задание.

В любой момент, — продолжил он, — можно изменить условия игры, задать новый предел изменения чисел или новую операцию. Для этого надо изменить содержимое соответствующих регистров. Думаю, что ты запомнил: в регистре 0 записан предел заказываемых машиной чисел, а в регистре 9 — код операции. Потренируйся немного с программой, ведь завтра ты будешь учить арифметике Машу, поэтому постарайся сам не попасть впросак. Да, чуть не забыл. Очень важный момент. Если второе число оканчивается нулем, то нуль этот не показывается. Например, 18,4 означает 18 и 40, а числа 18 и 4 показываются как 18.04. Думаю, что ты и сам это бы понял, но проще предупредить заранее. Вообще, когда будешь писать инструкции к программам, помни, что рассчитывать на догадливость того, кто пользуется программой, считается у программистов плохим тоном. Инструкция должна давать полную информацию о том, что делать во всех возможных случаях и как трактовать любые получаемые результаты.

Хотя инструкции в такой форме используются только для калькуляторных программ, навык, полученный при их написании, пригодится и при работе на больших ЭВМ.

Как я уже говорил, инструкции для работы на больших ЭВМ включают в текст самой программы. Но суть от этого не меняется. Более того, все сообщения должны быть очень точными и трактоваться однозначно. Предусматривается проверка вводимых данных на допустимость и вывод специальных сообщений. Их называют диагностическими. Подобно врачу, устанавливающему диагноз болезни, они устанавливают диагноз ошибки. Например, если бы ты в качестве случайного числа ввел отрица-

тельную величину, то компьютер ответил бы на твой ввод что-то подобно этому: «Недопустимое число. Повторите ввод».

Впрочем, мы увлеклись. Наш калькулятор столь подробных сообщений выдавать еще не умеет. Поэтому нужно потренироваться, чтобы читать сообщения, состоящие из одних цифр.

Потренироваться в этот вечер мне, к сожалению, не удалось. В комнату вошла мама и строго сказала: «А который теперь час? Не пора ли закрывать заседание клуба?»





## ВЕЧЕР ВТОРОЙ

Рассказывает Елена Николаевна

*Калькулятор вносит новое в жизнь семьи. Еще две программы обучения счету. Лекция о стеке. Что такое транслятор? Программируем сами. Сколько километров от Вологды до Керчи? Калькулятор рисует узоры.*

Никогда не подумала бы, что калькулятор может внести столько нового в жизнь семьи. Прошел почти месяц после его покупки, а разговоры дома до сих пор часто вертятся вокруг него. Да что дома! Стоило мне рассказать о покупке на работе, как все сотрудники разом стали давать советы. Оказалось, что у большинства моих сослуживцев калькуляторы имеются давно и не лежат без дела. Мария Сергеевна теперь не садится за вязальную машину без калькулятора. Антонина Петровна стала худеть, используя «калькуляторную» диету. Даже наш начальник, Николай Николаевич, и тот смущенно признался, что каждый вечер играет с калькулятором в разные игры, программы которых составляет его сын, студент.

У нас же Маша с Андреем постоянно спорят из-за калькулятора, из-за «машинного времени», как говорит Игорь. Он сам предложил распределять это время заранее, как на настоящей ЭВМ, но с учетом не потребностей, а поведения.

Но главное не в этом. Пришлось мне признать правоту Игоря — Маша меньше, чем за месяц, научилась сносно считать. Во всяком случае испытание, которое мы устроили ей и сыну наших друзей, Машиному ровеснику, показало, что таблицу умножения Маша знает гораздо лучше. А ведь специально она эту таблицу не учила.

Правда, ее приобщение к калькулятору не обошлось без «синяков» и «шишек». Игорь развил слишком бурные темпы — начал обучать дочку арифметике, когда она еще цифр себе толком не представляла.

Программа «Повтори число» показалась мне слишком сложной для дочери. К тому же при работе с ней сразу нельзя понять, правильно ли ты ответил или ошибся. Я сказала об этом мужу. Он со мной согласился, но переделывать программу не стал. «Андрей научится программировать как следует и исправит эту программу. Дело это не слишком трудное, а ему будет и интересно, и полезно», — сказал Игорь. Зато две программы, которые он предложил потом, мне очень понравились, да и Маше тоже. Работать с ними просто, а пользу ей они принесли большую.



## ПАЛОЧКИ И ЦИФРЫ

	0	1	2	3	4	5	6	7	8	9
0	Cx	7	ПО	1	П1	ПС	F10 <sup>x</sup>	ПА	ИП1	ИПС
1	1	+	ПС	ХУ	ИПА	x	1	+	КПС	FL0
2	09	Cx	ПО	С/П	ПД	9	—	Fx<0	32	КИПД
3	БП	23	ИПД	Flg	1	+	ПД	КИПД	ИПД	БП
4	23									

## Инструкция

1. В/О С/П «0».

2. Если «нарисовать» на индикаторе палочки, используя клавишу «1», то результат после нажатия С/П — число палочек. Например, III С/П «3». Если ввести любую цифру от 1 до 8, то результат — выведенные на индикатор палочки. Например 2 С/П «11».

3. Перейти к п. 2.

Первую из этих программ Игорь назвал «Палочки и цифры». Заданий никаких она не ставит, а, наоборот, дает ответы на вопросы, поставленные ребенком. Ребенок может ввести с клавиатуры произвольное число палочек — единиц, а машина сосчитает их и покажет цифрой их количество. Правда, так как индикатор рассчитан только на 8 позиций, то и число палочек ограничено этой же величиной. Для начала это, по-моему, неплохо. Во всяком случае понятнее ребенку, да и полезнее, чем сразу задавать ему примеры на сложение и вычитание. К тому же программа очень умная. Она может не только подсчитать число палочек, но и наоборот, то есть по набранной цифре рисовать их на индикаторе. Причем, для перехода от режима к режиму вообще ничего делать не надо. Машина сама понимает, что если задали палочки, она подсчитывает их число, а если задали цифру, рисует палочки.

Время получения 0 при подготовке к работе около 30 секунд, время ответа — 2 — 4 секунды.

Должна признаться — эта программа доставила большое удовольствие не только Маше, но и мне.

Понравилась нам и другая программа. В отличие от первой она показывает палочки и просит ребенка сосчитать их число или показывает число и просит «нарисовать» палочки.

В ней Игорь вновь использовал генератор псевдослучайных чисел, о котором он рассказал Андрею. А для переключения режима работы — считать палочки или рисовать их — использовал переключатель Р — Г. Вообще этот переключатель задает режим вычисления тригонометрических функций: Р — радианы, Г — градусы. Для Маши же, чтобы ей было легче запомнить Игорь сказал, что в этой программе Р означает «рисуй», а Г — «гадай», то есть «считай».

## СОСЧИТАЙ ПАЛОЧКИ

	0	1	2	3	4	5	6	7	8	9
0	Flg	ПД	7	ПО	1	П1	ПС	F10 <sup>x</sup>	ПА	ИП1
1	ИПС	1	+	ПС	ХУ	ИПА	x	1	+	КПС
2	FL0	10	Cx	С/П	ИПД	ИП2	x	Flg	+	ПД
3	КИПД	ХУ	ИПД	—	ПД	8	x	1	+	ПС
4	Flg	Fcos	Fx>0	52	КИПС	С/П	ИПС	—	Fx=0	44
5	БП	59	КИПС	ИПС	С/П	КИПС	—	Fx=0	53	КИПС
6	ИПС	/-/	x	БП	23					

## Инструкция

1. Нецелое число от 11 до 99 В/О С/П «0».

2. С/П «задание» (палочки или цифры).

3. Ответ С/П. При правильном ответе сообщение вида: — пп ... п (п — число палочек). Перейти к п.2. При ошибке повторение задания.

Примечание. Все задания определяются положением переключателя Р — Г. В положении Г — задание в виде палочек, которые нужно пересчитать, в положении Р — в виде числа, обозначающего количество палочек, которое нужно вывести на индикатор.

Эти программы вместе с показанными Игорем раньше, закрыли, как выразился муж, проблему матобеспечения для дочери.

Я попробовала программировать и сама. Оказалось, что это интересно. А вначале я побаивалась. Все-таки математику подзабыла изрядно. Выяснилось, что особых знаний математики для работы на калькуляторе и не нужно, во всяком случае для решения простеньких задач. Достаточно обыкновенного здравого смысла.

Правда, многие вещи мне еще не понятны. Все эти стеки, подпрограммы, косвенные команды для меня, как китайская грамота. Несколько успокоило меня то, что и Андрей относился к этим премудростям так же. Попытались мы с ним сами разобраться, да ничего не вышло. Тогда сын предложил, пусть отец устроит нам «ликбез», прочтет лекции по этим вопросам, как он рассказал ему как-то о псевдослучайных числах. Я, кстати, пожалела, что на этой лекции не присутствовала.

Несколько вечеров Игорь колдовал над калькулятором и вчера предупредил нас, что у него все готово и он может рассказать нам о стеке.

И вот сегодня вечером мы собрались на лекцию. За лекторской кафедрой, или, иначе, по одну сторону стола — Игорь, по другую — публика: мы с Андреем. Машу после некоторого сопротивления с ее стороны от лекции отстранили.

— Во-первых, тебе это неинтересно, а во-вторых, сегодня по телевизору новый мультфильм, — объяснил ей муж.

Последний довод оказался решающим.



## Лекция о стеке

Стек, — начал Игорь Петрович профессорским тоном, — слово английское. Оно означает оно охапку или поленицу и используется в вычислительной технике для характеристики способа организации памяти.

Представьте себе узкую трубку, поставленную вертикально, в которую забрасывают шарики. Первый шарик падает на дно трубки, на него ложится второй, потом третий... Если перевернуть трубку, то первым выскочит тот шарик, который зашел последним, затем предпоследний, в самом конце выскочит шарик, попавший в трубку первым. Подобным же образом устроен стек. Его характеризует принцип: «последним пришел, первым вышел».



По принципу стека устроена в калькуляторе область памяти, где хранятся числа, непосредственно участвующие в арифметических операциях. Каждое число занимает при этом одну ячейку. Она называется регистром стека. Таких регистров всего четыре. У каждого есть свое имя, обозначаемое латинской буквой X, Y, Z и T. Есть и еще один дополнительный регистр, своего рода склад для забывчивых или по-научному «регистр предыдущего результата». О нем я скажу позже. Самый главный из регистров стека — регистр X. Его содержимое мы обычно видим на экране калькулятора. Если опять вернуться к аналогиям и сравнить стек с трубкой, поделенной на этажи, то RX будет находиться на первом этаже. Следующий, второй, этаж занимает регистр Y. Эти два регистра называют операционными потому, что из них берутся числа для проведения арифметических операций: сложения, вычитания, умножения и деления. Причем при вычитании в верхнем регистре PY находится уменьшаемое, а при делении — делимое. Думаю, вам понятно, что при сложении и умножении взаимное расположение участников операции — они называются операндами — не имеет значения.

Расположенные на третьем и четвертом этажах регистры Z и T — своеобразные камеры хранения, где тоже можно держать числа и извлекать их по мере надобности для проведения вычислений. Непосредственно использовать их мы не можем. Нельзя, скажем, вычест содержимое регистра T из содержимого регистра X или получить произведения чисел из регистров Z и Y. Каждый раз при выполнении арифметических действий — сложения, вычитания, умножения, деления — нам доступно лишь то, что лежит в RX и PY. Именно оттуда берутся операнды этих двуместных операций. Математики говорят также и об одноместных операциях, таких, которые требуют для своего выполнения лишь одного числа. Это, например, извлечение квадратного корня или вычисление синуса. Такие операции можно проводить только с содержимым регистра X.

К счастью, содержимое регистров можно перетасовывать. Иногда это происходит автоматически. Так, при проведении двуместной операции результат ее попадает в RX; содержимое PZ опускается в PY и таким образом

становится доступным; то, что находилось в PT, переходит в PZ, а содержимое PY пропадает. Что же касается числа из RX, то оно не пропадает, а опускается в «подвал» — регистр X1. Понятно теперь, почему он называется регистром предыдущего результата, или «регистром для забывчивых»?

— А какой прок от этого регистра, если мы не можем достать его содержимое? Ведь, как я понял, числа движутся в стек сверху вниз? — спросил Андрей.

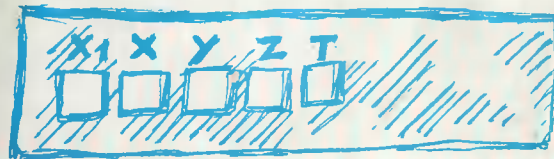
— Не торопись, правильно понял ты только, что автоматически движутся числа сверху вниз при проведении арифметических операций. Я еще не успел сказать, что в нашей власти изменять направление этого движения и заставить числа в стеке перемещаться, как шарики в пирамиде Рубика. Для этой цели служат специальные команды. Обычно их работу описывают с помощью специальных диаграмм, но я подумал, что интереснее промоделировать эти движения на самом калькуляторе.

Дело осложняется тем, что на индикаторе мы видим лишь содержимое RX, чтобы увидеть числа из других регистров, как раз и нужно применять команды, которые мы собираемся изучать.

— Замкнутый круг какой-то, — сказала я.

— Не зря же я корпел несколько вечеров. Программа, которую я составил, позволяет наглядно видеть перемещения чисел по стеку. Можно сказать, что я использовал калькулятор для моделирования калькулятора. Это, в общем, для вычислительной техники не такая уж редкость. Очень часто поведение одной машины моделируют с помощью специальной программы на другой.

А теперь представьте себе, что индикатор нашего калькулятора разбит на части — разряды длиной в один символ-цифру, каждый из которых показывает содержимое определенного регистра стека. Первый разряд — регистр X1, второй — X, третий — Y, четвертый — Z, и пятый — T. Для удобства я сделал даже специальный трафарет. На нем над соответствующими разрядами написаны имена регистров стека.



Как и всякая другая, моя модель имеет ограничения. В отличие от реального стека в моей модели в каждом «регистре» может храниться лишь одна цифра (в реальном — число, имеющее до 8 значащих цифр). Во всем остальном модель адекватно отображает свойства стека, поэтому изучать их по ней будет легче.

Свою программу-модель я назвал «Числа по кругу».



## ЧИСЛА ПО КРУГУ

	0	1	2	3	4	5	6	7	8	9
0	кппс	+	кбпд	кппс	—	кбпд	кппс	x	кбпд	кппс
1	$Fx^2$	кбпд	кппс	$F\odot$	кбпд	кппс	ху	кбпд	ип1	бп
2	22	ху	п6	9	—	$Fx \geq 0$	31	пп	62	$\uparrow$
3	кбпд	ип4	ип3	ип2	ип6	п2	$F\odot$	п3	$F\odot$	п4
4	ху	п5	5	п0	0	п6	ипв	x	кип6	+
5	$FLO$	46	с/п	$\uparrow$	1	—	3	x	па	кбпа
6	ип2	п1	ип5	ип4	ип3	ип2	в/о	сх	1	0
7	пв	3	5	пд	6	0	пс	1	п1	$\uparrow$
8	$\uparrow$	$\uparrow$	кбпд							

## Инструкция

1. БП 67 С/П «1111».
2. Код стековой операции С/П. На индикаторе новое положение чисел в стеке.
3. Перейти к п. 2.

Примечание. Коды стековых операций в модели: + (1), — (2),  $\times$  (3),  $Fx^2$  (4),  $F\odot$  (5), ху (6), FBx (7), ввод нового числа (цифры) с клавиатуры: цифра  $\uparrow$  8 С/П.

Программа, как видите, довольно сложная. О ней мы сейчас говорить не будем. Если хотите, разберитесь в ней сами. Думаю, это вам будет полезно. А теперь давайте приступим к демонстрации.

Муж передал текст программы сыну и Андрей приступил к работе. Не перестая удивляться, как быстро он освоил клавиатуру. Мне кажется, что вводит программы он даже быстрее Игоря. Вот и теперь всего несколько минут понадобилось Андрею для ввода длинного перечня команд.

— Запускается программа несколько непривычно для вас, — сказал Игорь. — Не с нулевого адреса, а с 67-го. Собственно, ничего необычного в этом нет. Программу можно запускать с любого адреса. Для этого управление надо передать на этот адрес. Удобнее всего это сделать, когда нужный адрес — нулевой. Тогда передача заключается в нажатии всего одной клавиши В/О. Если же интересующий нас адрес ненулевой, то в таком случае для передачи управления сначала следует нажать клавишу БП, потом набрать этот адрес, а затем уже для запуска машинки нажать клавишу С/П. Итак, переводим калькулятор в режим вычислений и нажимаем БП 67 С/П.

Как и всегда, после нажатия этой клавиши началось таинственное мелькание цифр на индикаторе. Честно говоря, до сих пор не могу отделаться от мысли, что в это время маленькое устройство напряженно думает и на индикаторе мелькают именно отражения его мыслей.

Но вот движение прекратилось и наша трафаретка показала, что во всех регистрах стека записаны единицы.

- А что дальше? — нетерпеливо спросил сын.  
 — Вот тебе перечень команд и условных кодов в моей модели.

Команда	Код
+	1
—	2
x	3
$Fx^2$	4
$F\odot$	5
ху	6
FBx	7

Примечание. Ввод нового числа: цифра  $\uparrow$ . Все наборы кончаются нажатием клавиши С/П.

— Теперь, чтобы промоделировать какую-нибудь операцию, нужно набрать ее код. Скажем, код ввода — 8. Начнем с этой операции. Введем, к примеру, число 5. Набираем 5  $\uparrow$  8 С/П. Через 20 секунд на индикаторе высветилось: 1 5 1 1 1.

— Теперь введем четверку: 4  $\uparrow$  8 С/П.

Таким же образом мы ввели 3 и 2. В результате этих манипуляций на экране появилось 1 2 3 4 5.

— Мы видим, что ввод нового числа в стек сдвигает все числа в стеке «вверх», выталкивая содержимое самого верхнего регистра Т наружу.

Теперь проиллюстрируем двуместную операцию. Все они в смысле движения чисел по стеку работают одинаково, поэтому ограничимся сложением.

Андрей набрал 1 С/П. Снова замелькали цифры и на индикаторе появилось 2 5 4 5 5.

— Как я и говорил, все «опустилось», — заметил Игорь. — Обратите внимание, что содержимое регистра Т не просто «перехало» в РZ, а оставило на прежнем месте свою копию. Иными словами, скопировалось в РZ.

Теперь посмотрим, как работают команды, ответственные за движение чисел по стеку.

Начнем с команды подъема FBx.

После набора 7 С/П числа в стеке действительно поднялись, причем содержимое самого нижнего регистра X1 не изменилось — оно скопировалось в РХ, а вот то, что было в РТ, пропало. Этот факт был немедленно отображен на индикаторе: 2 2 5 4 5.

Теперь поменяем местами числа в РХ и РY. К такой операции часто прибегают, чтобы правильно расположить операнды для деления или вычитания. Соответствующая команда в модели выглядит так: 6 С/П. На индикаторе засветились цифры: 2 5 2 4 5.

— Точно, — сказал Андрей, — как в инструкции: содержимое двух регистров поменялось местами, а то, что было в остальных, осталось на месте.



— Не совсем так, — возразил Игорь. — Попробуй переставить X и Y еще раз.

Андрей снова набрал. Индикатор бесстрастно зафиксировал: 5 2 5 4 5.

— А почему в X1 теперь пятерка? — спросил сын.

— Потому что одновременно с перестановкой чисел в регистрах X и Y содержимое первого из них копируется в X1, а предыдущее содержимое исчезает. Первый раз в X1 и X были одинаковые числа, вот ты этой перестановки и не заметил.

Еще одна стековая операция: циклический сдвиг содержимого всех регистров. На калькуляторе эта команда обозначается  $\odot$ , а в моей программе нужно набрать 5 C/P.

После этой команды на индикаторе появилось: 2 5 4 5 2.

— Обратите внимание, что и тут, кроме самой перестановки, происходит копирование содержимого регистра X в регистр X1.

Теперь, наверное, вы поняли, как обращаться с этой программой. Советую потратить время и освоить стековые операции. Знание их очень пригодится при написании калькуляторных программ. Недаром говорят, что программирование на калькуляторе, это прежде всего работа со стеком.

Тут мне пришла в голову мысль.

— Слушай, Игорь, а ведь программа твоя, кроме утилитарного значения — освоить работу стека может быть занятной игрой. Играть в нее сможет даже Маша. Это как кубик Рубика. Дано, скажем, расположение чисел в стеке 1 2 3 4, а нужно перестроить его и получить 4 3 2 1. По-моему, это интересно.

— Точно, — согласился Игорь. — Я, правда, не ставил целью создать игру, но она получилась сама собой. Можете играть с пользой для дела. Но игра игрой, а мне было интересно составлять эту программу, потому что сам я на этом примере убедился, с какими проблемами сталкиваются разработчики новых ЭВМ при необходимости моделировать их работу на старых. Такие же проблемы возникают при разработке программ, моделирующих работу «фиктивных» машин: Алгол-машин, Бейсик-машин или Паскаль-машин.

Иначе говоря, машин, понимающих различные алгоритмические языки, ибо и Алгол, и Бейсик, и Паскаль — названия алгоритмических языков, или, как их еще называют, языков высокого уровня.

— Ну, сел на своего конька. Ты даже не заметил, что мы перестали тебя понимать.

— Да, папа, — согласился сын. — Но с другой стороны, то, что ты говоришь, очень интересно.

— Тогда придется прочесть вам очередную небольшую лекцию.

## Об алгоритмических языках и трансляторах

Появление вычислительных машин сразу же поставило на повестку дня проблему общения с ними. Необходимо было придумать способы записи алгоритмов, простые и доступные человеку и одновременно понятные машине. Только в этом случае можно было надеяться на широкое использование вычислительной техники специалистами самых разнообразных профессий.

Так появились алгоритмические языки. Нестрого алгоритмический язык можно определить как набор определенных символов, слов и фраз, составленных из этих символов, и правил, показывающих, как понимать тексты, включающие эти слова и фразы. От языка требуется однозначность при интерпретации текста, понятность текстов как человеку (ему их писать), так и машине (ей их выполнять). Что касается однозначности, то «человеческие» языки сразу отпали. Вспомните старую шуточную загадку: «А и Б сидели на трубе. А упало, Б пропало. Что осталось на трубе?». Весь смысл здесь на двояком понимании И: либо как союза, либо как третьей буквы. Так вот для ЭВМ такие ситуации недопустимы. Нужно, чтобы текст всегда допускал только одно толкование.

Следующие два требования по сути своей противоречивы. Судите сами. Чем богаче язык, чем больше в нем слов, идиом, оборотов, тем легче человеку выразить на нем многообразие своих мыслей, в том числе и суждений об алгоритмах. Машина же в конечном счете понимает лишь «язык», состоящий из нулей и единиц. Поэтому, чтобы написанная на алгоритмическом языке программа была понятна для ЭВМ, приходится ставить между человеком и компьютером посредника — специальную программу, переводящую текст с алгоритмического языка на язык ЭВМ.

Такие программы получили название трансляторов. Сегодня трансляторов создано очень много. Во-первых, потому что много стало различных типов компьютеров, а во-вторых, и алгоритмических языков предостаточно. О причинах многообразия языков мы поговорим позже, а пока — о типах трансляторов.

Существуют два типа трансляторов: интерпретаторы и компиляторы. Интерпретаторы переводят программу дословно, причем в ритме синхронного перевода. Прочитал — перевел — выполнил. Для этого транслятор должен постоянно находиться в памяти. Ведь ему нужно все время просматривать текст программы на алгоритмическом языке. Это удобно при отладке. Достаточно переписать оператор по-другому, и интерпретатор тут же начнет выполнять исправленную программу.

Неудобство же такого способа перевода в том, что часть оперативной памяти ЭВМ постоянно занята самим транслятором. Поэтому для хранения самой задачи остается меньше места. Кроме того, приходится неоднократно переводить одно и то же. К примеру, операторы, записанные в теле цикла, должны переводиться столько раз, сколько этот цикл будет прокручиваться. Естественно, это замедляет выполнение программ.

От этого недостатка свободны трансляторы другого типа — компиляторы. Их работу можно сравнить с деятельностью переводчика литературного произведения. Сначала читается все произведение, а затем делается литературный перевод. Конечно, такой перевод происходит медленнее, чем дословный, но зато его надо сделать только один раз. Результат работы компилятора — новый текст программы, уже на языке компьютера. При выполнении такой программы нет необходимости держать в памяти транслятор. Да и работать скомпилированная программа будет быстрее.

Недостаток же компилятора в том, что программист не может исправить текст оттранслированной программы. Ведь он не знает, где хранятся команды, соответствующие тому или иному оператору: перевод-то сделан не дословно, а по смыслу. Поэтому при отладке приходится исправлять исходный текст, потом снова транслировать всю программу и только после этого выполнять расчеты по программе на ЭВМ.



В настоящее время используются оба типа трансляторов. Причем интерпретаторы более популярны на маленьких персональных компьютерах, а компиляторы — на больших и сверхбольших ЭВМ, где цена машинного времени неизмеримо выше. Существуют и трансляторы, использующие оба принципа.

Однако всем им присуща одна и та же проблема. Чем сложнее язык программирования, тем сложнее должна быть и программа-транслятор. Отсюда и пошло многообразие языков. Одни из них попроще, зато и трансляторы для них простые, и программы, получаемые в результате трансляции, более эффективны. Но, к сожалению, программировать на них трудно. Другие языки побогаче, в них больше возможностей для записи сложных алгоритмов, мических конструкций, но и трансляторы с этих языков пообъемнее и сложнее, да и оттранслированные программы не очень эффективны. В общем, постоянно приходится выбирать между удобствами и возможностями.

Есть и еще одна причина, которая способствовала языковому многообразию. Раз нельзя сделать язык, удобный всем пользователям (и ЭВМ, разумеется), то почему бы не создавать языки целевого назначения? Ведь, скажем, экономисту не очень нужны синусы и логарифмы, математик же вполне может обойтись без широкого набора средств для работы с текстами. При таком подходе можно добиваться больших удобств для пользователя при сравнительно простом трансляторе.

Так стали появляться языки, ориентированные на решение специализированных задач. Например, Фортран предназначался для проведения научных технических расчетов. Это вообще был первый алгоритмический язык. Он довольно прост, имеет широкий набор средств для записи вычислительных алгоритмов, зато удобен при обработке текстовой информации.

Чуть позже Фортрана был создан язык Кобол. Его назначение — решать экономические и управленческие задачи. В Коболе прекрасно развиты средства ввода и вывода. К достоинствам его следует отнести и похожесть конструкций на фразы обычного (английского) языка. Вместе с тем, на Коболе трудно программировать вычислительные операции.

Язык Снобол создан специально для обработки цепочек литер, то есть текстов.

Можно привести в качестве примера специализированного языка Лисп. Он используется при машинном анализе таких научных проблем, как искусственный интеллект, теория игр, а также для обработки текстов.

По мере увеличения быстродействия и оперативной памяти компьютеров предпочтение стали отдавать все же языкам универсальным. К наиболее популярным среди современных языков относятся Бейсик, Паскаль, Рапира.

Язык Рапира создан у нас в стране новосибирскими учеными. В нем используются слова русского языка. Предназначена Рапира в первую очередь для обучения программированию школьников, хотя может быть с успехом использован для написания серьезных программ.

Паскаль тоже создавался для обучения. Однако простота и четкость структур программ, написанных на этом языке, сделали его очень популярным среди программистов.

Наконец, Бейсик. Надо сказать, что все перечисленные языки используются на персональных компьютерах. Но по распространенности Бейсик держит, безусловно, пальму первенства.

Простота этого языка, легкость его изучения и использования, а также тот фактор, что Бейсик появился раньше и Рапиры и Паскаля, привели к тому, что сегодня трудно назвать ЭВМ, на которой не было бы транслятора с этого языка.

Кстати, им уже оснащаются и программируемые микрокалькуляторы. Есть он, например, на отечественном калькуляторе «Электроника МК-85». Но вернемся к нашему микрокалькулятору «Электроника МК-54».

— Мне захотелось слегка оживить нашу беседу:

— Скажи, Андрей, что означает слово «транслятор»?

— Сын задумался. — Вот уж не думал, что изучение калькулятора даст вам повод проверить мое знание английского языка, — сказал он.

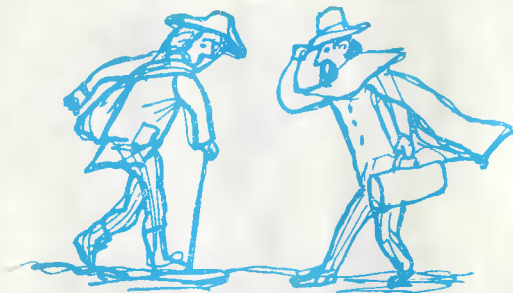
— А почему бы и нет? — вступил в разговор Игорь. — Смотрите, сколько проблем мы сможем затронуть благодаря нашему электронному другу. Это еще раз показывает взаимосвязанность различных областей Знания. Я произношу это слово с большой буквы. Сегодня, чтобы учиться программированию, нужно и в английском ориентироваться, и с математикой быть на ты, да и логику мышления дисциплинировать.

— Ну так как, Андрей? — вернула я сына к своему вопросу?

— Догадался. Слово транслятор происходит от английского глагола translate — переводить. Значит транслятор и есть переводчик.

— Очень точно! — воскликнул муж. Не зря учишь язык в школе.

А теперь давайте поговорим о составлении программ. Я вижу, что здесь вы сталкиваетесь с некоторыми трудностями.



Начнем с конкретной задачи. Помните, недавно мы смотрели по телевизору передачу о театре?

— Да, — ответила я. — Хорошая передача. Очень интересно сыграл артист Бурков встречу Счастливецва и Несчастливецва.

— Это из «Леса», что ли? — проявил свою эрудицию Андрей.

— Вот, вот, — подхватил Игорь. Два бродячих актера встречаются на дороге. И идут они — один из Вологды в Керчь, а другой из Керчи в Вологду... А как вы думаете, какое расстояние между этими городами?

— Большое, наверное. А какое, я и не задумывалась.

— Ну-ка Андрей, неси карту и линейку.

Через минуту то и другое лежало на столе.

— На карте обозначен масштаб, — продолжал Игорь. Можно приложить линейку, измерить расстояние в сантиметрах, затем умножить на масштаб. Измерил? — спросил Игорь у сына. — Вот



тебе калькулятор. Перемножить на нем пару чисел можно и без программы. Сколько получилось?

— 1560 километров.

— Ничего себе, — не удержалась я. И ведь пешком.

— Да, слава богу, тяжелые времена для актеров миновали.

Ну ладно, оставим их в покое. А ты, Андрей, вычисли-ка расстояние от Ленинграда до Монтевидео.

Андрей полистал маленький атлас и разочарованно сказал: «Не знаю, как это сделать — ведь они в разных полушариях и на разных картах».

— А моряки еще давным давно умели это делать. Причем довольно точно. Во всяком случае точнее, чем ты с помощью карты и линейки. Единственное, что для этого требуется, знать координаты искомых точек на земной поверхности: широту и долготу. Затем координаты подставляют в формулу и вычисляют искомое расстояние.

Вот эта формула:

$$S = 111,2 \cdot \arccos [\sin \varphi_1 \cdot \sin \varphi_2 + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \cos (\lambda_1 \lambda_2)].$$



Здесь  $S$  — расстояние в километрах,  $\varphi_1$  и  $\lambda_1$  — широта и долгота первой точки,  $\varphi_2$  и  $\lambda_2$  — широта и долгота второй точки. (Для северного и восточного направления они берутся со знаком «плюс», а для южного и западного — со знаком «минус».)

Формула эта возникла в сферической геометрии, то есть геометрии на шаре. Величина 111,2 — средняя длина дуги в один градус на земной поверхности, измеряемая в километрах. Нужно подставить в формулу соответствующие координаты и провести расчет. Так мореплаватели раньше и делали.

— А что у них таблицы синусов и косинусов были? — дело это поинтересовался Андрей.

— Таких таблиц, которыми вы пользуетесь в школе, у них, конечно, не было. Но были другие, похожие.

— Насколько я понимаю, формулу написал ты не для того, чтобы мы поупражнялись в вычислении синусов и косинусов на калькуляторе, — сказала я. — Давай посмотрим, как быстро он считает.

Я набрала на клавиатуре первое попавшееся число, 456, и нажала F и sin. Экран потух и через несколько секунд загорелся результат:  $-4.5210^{-1}$ .

— Здорово. Я за это время даже таблицы полистать бы не успела.

— В общем, не очень-то здорово, — задумчиво сказал Игорь. Две секунды на расчет тригонометрической функции — непустым делом для вычислительного устройства. Конечно, это недостаток нашего калькулятора. Но все познается в сравнении. Как ты правильно заметила, это значительно быстрее, чем искать значение функции в таблицах или на логарифмической линейке, к тому же удобнее и точнее. Но это к слову. Вернемся к нашей формуле.

Скажи-ка Андрей, за какое время ты можешь выполнить расчет по ней вручную — с помощью таблиц, разумеется?

Андрей уставился в потолок, что свидетельствовало о его глубокой задумчивости, и сказал: «Ну, минут за десять смогу».

— А если составить программу для калькулятора, то расчет по ней не займет и полминуты. Я хочу, чтобы вы с мамой этим сейчас и занялись. А я пока читаю газету.

Игорь раскрыл свой любимый «Советский спорт». Я решила, что пора перехватывать инициативу.

— Значит так. Сначала надо решить, как организовать ввод. — Чтение «Школы начинающего программиста» в «Науке и жизни» не прошло даром. — Будем вводить парами, — подхватил Андрей. Он тоже читал журнал, — сначала  $\varphi_1$  и  $\lambda_1$  через стрелку, потом также  $\varphi_2$  и  $\lambda_2$ . После ввода сразу запишем их в адресуемые регистры. Начало программы сложилось сразу: ввод  $\varphi_1 \uparrow \lambda_1$ , затем собственно программа

00.П1.01.XY 02.П2. 03. C/П

За фрагментом следует второй ввод  $\varphi_2 \lambda_2$  и продолжение программы.

04.П3 05.XY 06.П4

Значения координат записаны в регистрах с первого по четвертый. Можно приступать к вычислениям.

07.ИП2 08. Fsin 09.П5 (запоминаем промежуточный результат)10

ИП4 11. Fsin 12.П6 (запоминаем уже другой синус)

Стоп! Здесь надо умножить  $\sin \varphi_1$  на  $\sin \varphi_2$ . Значение второго синуса как раз в регистре X и больше нам не понадобится. Поэтому записывать его в адресуемый регистр не надо. Можно сразу вызывать содержимое P5 и проводить умножение. Я



зачеркнула команду 12 и вместо нее написала 12.ИП5 и дальше 13.X.14.П6.

Сын посмотрел на меня с уважением. Вместе мы продолжили

15.ИП2 16. Fcos 17.П7 18.ИП4 19. Fcos

Далее по проторенной дорожке

20.ИП7 21.X 22.П8

Теперь вычисляем косинус разности

23.ИП1 24.ИП3 25

Я твердо запомнила, что при вычитании уменьшаемое должно быть в регистре Y.

26.Fcos 27.ИП8 28.X 29.ИП6 30. + 31.Farccos

Теперь можно записать в программе числовой сомножитель

32.1 33.1 34.1 35., 36.2 37.X

Ставим последнюю команду

38.C/П

Андрей начал вводить программу.

Все это время Игорь сидел с отсутствующим видом, углубившись в чтение футбольных сообщений, и никак не вмешивался в наши действия. Наконец, ввод был закончен, и Андрей «снял» координаты: для Вологды —  $39,5^\circ$  восточной долготы и  $59^\circ$  северной широты, а для Керчи — соответственно  $35,5^\circ$  и  $45,5^\circ$ . Мы поставили переключатель Р — Г в положение Г — градусы. Андрей перевел калькулятор в режим вычислений, ввел исходные величины и запустил калькулятор на счет. Секунд через 20 на индикаторе появилось: 1525.0119.

Результат не так уж сильно отличался от полученного измерения по линейке. Следовательно, программа считала правильно. Но точность! Мы получили расстояние, измеренное до сантиметров!

Наконец, Игорь отложил газету.

— Так, сказал он. — Начнем обсуждение ваших результатов с конца. Сколько получилось? — он посмотрел на индикатор. Примерно это будет 1520 километров.

— Почему примерно? — возмущился Андрей. — У нас же вычислен точный результат.

— Не совсем так. Конечно, считает калькулятор точнее, чем вручную. Но точность результата зависит не только от расчетных погрешностей, хотя и они могут играть важную роль. Самые существенные для результата погрешности кроются обычно в исходных данных. Возьмем, к примеру, нашу задачу. С какой точностью измерены координаты городов? До одной десятой

доли градуса? А это уже даст итоговую погрешность результата порядка десяти километров.

— Так значит восемь цифр результата не нужны? — спросил сын.

— Нет, почему же. Они могут понадобиться. Но только в одном случае — когда расчет рассматривается как контрольный пример, используемый для проверки правильности ввода программы. Достоинство любой ЭВМ в том, что в отличие от человека вычисления по одной и той же программе с одними и теми же исходными данными она всегда будет повторять абсолютно одинаково. Поэтому критерием правильности введенной программы будет воспроизведение контрольного примера и получение результата с точностью до всех выводимых значащих цифр. Поэтому-то результат контрольного примера и приводят с излишней точностью. Ясно?

— Ясно, — не очень дружно ответили мы с Андреем.

— Надеюсь, что после некоторого времени самостоятельной работы голоса ваши станут звучать уверенней. А сейчас давайте поговорим о вашем совместном произведении.

Задачу свою вы выполнили. Создали программу, которая позволяет проводить требующиеся расчеты и получать их результаты. Работать с вашей программой удобно. Очень хорошо, что ввод вы сделали безадресным, то есть не заставляете пользователя запоминать, какие данные и в какой регистр нужно вводить, нажимать лишние клавиши при вводе. Мне нравится, что и числовую константу вы записали прямо в текст программы. Если место в памяти есть, то пожалуй, такая запись надежнее, чем ввод числа в адресуемый регистр. Так что с внешней точки зрения к программе претензий нет. Но вот хорошо ли она написана?

Заранее предвижу ваши возражения. Вы уже достаточно подкованы и знаете, что сравнивать программы трудно и вообще хорошая программа — это работающая программа. Все это так. И тем не менее ... Программу, решающую вашу задачу, можно сделать почти вдвое короче. При этом она останется столь же удобной, а время ее работы даже несколько уменьшится. Да и переделка тоже не отнимет много времени.

Поймите меня правильно. Я не призываю вас, сделав программу для решения конкретной задачи, днями и неделями пытаться ее совершенствовать. Я хочу, чтобы вы научились составлять короткие и быстрые программы сразу, не тратя время на их оптимизацию. Это — техника программирования. Смысл ее аналогичен технике спортсмена. Возьмите гребца. Если он не умеет правильно работать веслом, то сколько сил ему надо приложить, чтобы развить приличную скорость! Если же он знает, что весло не надо «топить», не надо «бросать», то



## Инструкция

1. Сх П2 П5 П6 П9 ПС1 П0 П1 П3 П8 83 ПА 88 ПВ В/О С/П «0»
2. 0 или 1 (заданное число) С/П «число побед калькулятора» ХУ «число побед игрока».
3. Перейти к п.2.

Примечание. Переключатель Р—Г перед началом работы поставить в положение «Р».

— Программа, как видите, довольно велика, — сказал Игорь Петрович. — Из 98 ячеек программной памяти она занимает 96. Поэтому некоторые операции перед ее запуском приходится делать вручную. Они, правда, несложны. Нужно очистить регистры 2, 5, 6, 7, 9 и С, то есть занести в них нуль, потом записать единицу в регистры 0, 1, 3 и 8 и ввести еще два числа: 83 в регистр А и 88 в регистр В.

— Многовато, — сокрушенно сказал Юрий Кузьмич.

— А чему удивляться! Программа-то ни много, ни мало мышление моделирует. Думаю, в памяти каждого из вас сидит побольше чисел. Ну вот, все готово. Нажимаем В/О С/П. На индикаторе появился 0. Калькулятор готов к соревнованию.

Условия игры просты. После каждого останова надо ввести 0 или 1 и нажать С/П. Если калькулятор угадает набранное число, он прибавит единицу к счетчику своих побед. В противном случае увеличит на единицу счетчик игрока. Эти счетчики расположены в регистрах стека. Победы калькулятора — в РХ, они видны на индикаторе, ваши победы — в РУ. Чтобы посмотреть его содержимое, надо нажать клавишу ХУ. Вот и все. Начинай, Юра.

Очень скоро счет стал расти в пользу калькулятора. Тогда Юрия Кузьмича сменил Сергей Владимирович. Ситуация не изменилась.

— Эх, вы, мыслители, — вмешался Леонид Антонович, — дайте мне. Он достал из кармана монетку, стал подбрасывать ее и вводить единицу при «орле» и нуль при «решетке». Успехи калькулятора пошли на убыль.

— Так-то, — сказал он, — мой «калькулятор» не хуже твоего.

— Ничего не поделаешь, — согласился Игорь Петрович, — твоя взяла. При такой стратегии в чет-нечет проиграть нельзя. Правда, и выиграть тоже. А калькулятор ведь не только не про-



игрывает, он может и выиграть. Моя программа в отличие от твоего идеального датчика случайных чисел пытается учесть индивидуальные особенности игрока, не применяющего случайную стратегию. У игрока всегда присутствует элемент «воли», склоняющий его к некоторой периодичности в загадывании чисел. Вот этот волевой фактор и пытается определить программа и использовать при своих ответах. Как видите, это ей удается. Но, впрочем, хватит, — он отложил калькулятор.

— Какие новости? — обратился Юрий Кузьмич к присутствующим. — Леня, ты на новую работу перешел?

— Да, я теперь занимаюсь экономикой в геологии, точнее, работаю над созданием АСУ в этой области.

— А что, электроника тебе уже наскучила? — спросил Сергей Владимирович.

— Не то чтобы наскучила. Но я ведь давно уже стал заниматься не столько физикой, сколько программированием. Вот и захотелось попробовать себя в этой области всерьез. Интересно, что с самого начала я столкнулся не только с новой тематикой, но и с новой терминологией.

— Например? — поинтересовался Игорь Петрович.

— Ну, хотя бы «отрешать задачу».

— Действительно, — сказал Сергей, — не очень понятно. Решить задачу — это ясно, а «отрешать»? Ее что, каждый день решают?

— Да, есть задачи, которые оешают каждый день, есть еженедельные, ежемесячные.

— Ну, извини. Зачем же решать каждый день одну и ту же задачу? В том-то и смысл решен я, что прочел его один раз и получил то, что надо. Мне это ежедневное решение кажется нелепым. Это все равно, как если бы я каждый день решал задачу о распределении электронов в полупроводнике или, чтобы Андрею было понятней, о встрече двух путников, вышедших из пунктов А и Б. Стал бы ты, Андрей, решать такую задачу каждый день?

— Конечно нет, — ответил Андрей, — ее надо решить один раз, записать решение в виде формулы, а потом только подставлять в нее разные числа.

— Правильно, — похвалил его Сергей Владимирович, — а твои экономисты, Леня, как считают?

— Прежде всего, боюсь, что у тебя, да и у Андрея тоже, несколько однобокое представление о том, что такое задача. Ты и Андрей оказались в плену привычных представлений. Ну-ка, скажи, Андрей, что такое по-твоему «задача».

— Задача, это... это, когда есть условие, например камень бросают вертикально вверх с такой-то скоростью, и надо что-то определить, например на какую высоту поднимется камень. Или когда есть уравнение и надо найти неизвестное.

той же скорости он достигнет, затратив намного меньше сил, чем его коллега-новичок.

Вот я и хочу вам рассказать, как «правильно держать весло», или, по-другому, как писать короткие быстрые программы.

Для этого нужно хорошо знать свойства стека. Ибо программирование на микрокалькуляторе, повторю еще раз, — это прежде всего работа со стеком. Здесь вам должна пригодиться та программа-игрушка, которую я показал час назад: «Числа по кругу». Познакомьтесь с ней, и сами увидите, сколько возможностей дает знание стековых операций: Очень полезно при создании программы рядом с командами программы писать содержимое регистров стека. Впрочем, давайте рассмотрим это на примере составления программы, решающей ту же задачу, что и ваша, то есть вычисление расстояния между двумя точками на шаре. Писать программу будем в столбик. В первой колонке адрес команды, затем сама команда и после нее — содержимое регистров стека.



Начнем с ввода. Хотя я и похвалил вас за его организацию, тем не менее я сделал бы ввод не в два приема, как у вас, а в один. Четыре числа — это максимум того, что можно вместить в стек. А у нас данных именно столько. Далее не будем спешить с записью введенных величин в адресуемые регистры. Может быть, для части из них достаточно будет «временной жилплощади» стека.

Итак, ввод:

$\varphi_1 \uparrow \lambda_2 \uparrow \varphi_2 \uparrow \lambda_2$

После ввода числа расположатся по регистрам стека так:

$\lambda_2 \varphi_2 \lambda_1 \varphi_1$

В самом важном регистре  $R_X$  находится величина  $\lambda_2$ . Однако в нашей формуле она входит лишь в комбинации с  $\lambda_1$ , которая лежит пока в  $R_Z$ . Поэтому есть смысл, не проводя записи величины  $\lambda_2$  в адресуемый регистр, поменять местами содержимое  $R_X$  и  $R_Y$  и начать обрабатывать  $\varphi_2$ . Удобнее начать

обработку с вычисления косинуса, чтобы получить сомножители второго слагаемого и избавиться потом от массы промежуточных результатов. Перед вычислением косинуса значение  $\varphi_2$ , правда, следует сохранить: оно еще понадобится для получения синуса. Впрочем, давайте писать:

00 XY  $\varphi_2 \lambda_2 \lambda_1 \varphi_1$

01 PZ  $\varphi_2 \lambda_2 \lambda_1 \varphi_1$

02 Fcos cos  $\varphi_2 \lambda_2 \lambda_1 \varphi_1$

Величина  $\varphi_2$  запомнена в  $R_2$ , косинус от нее вычислен. Стоит ли его записывать в адресуемый регистр? Наверное, нет. Проще переписать его в стек, в  $R_T$ , а самим начать работу с  $\lambda_1$  и  $\lambda_2$ :

03 FO  $\lambda_2 \lambda_1 \varphi_1 \cos \varphi_2$

04 —  $\lambda_1$  —  $\lambda_2 \varphi_1 \cos \varphi_2 \cos \varphi_2$

05 Fcos cos  $(\lambda_1 - \lambda_2) \varphi_1 \cos \varphi_2 \cos \varphi_2$

Видите, значения  $\lambda_1$  и  $\lambda_2$  нам удалось обработать, не пересылая их в постоянные хранилища — адресуемые регистры. А теперь для получения последнего сомножителя второго слагаемого осталось вычислить косинус  $\varphi_1$ . Сделать это просто:

06 XY  $\varphi_1 \cos(\lambda_1 - \lambda_2) \cos \varphi_2 \cos \varphi_2$

07 P1  $\varphi_1 \cos(\lambda_1 - \lambda_2) \cos \varphi_2 \cos \varphi_2$

08 Fcos cos  $\varphi_1 \cos(\lambda_1 - \lambda_2) \cos \varphi_2 \cos \varphi_2$

Почему стоило запомнить  $\varphi_1$ , думаю понятно. Так же, как и  $\varphi_2$ , величина эта нам еще пригодится. Что же касается второго слагаемого, то для его вычисления все готово.

09 X cos  $\varphi_1 \cdot \cos(\lambda_1 - \lambda_2) \cos \varphi_2 \cos \varphi_2 \cos \varphi_2$

10 X cos  $\varphi_2 \cdot \cos \varphi_1 \cdot \cos(\lambda_1 - \lambda_2) \cos \varphi_2 \cos \varphi_2 \cos \varphi_2$

Теперь можно приступить к отработке первого слагаемого, оставив второе до поры до времени в стеке.

11 IP1  $\varphi_1 \cos \varphi_2 \cdot \cos \varphi_1 \cdot \cos(\lambda_1 - \lambda_2) \cos \varphi_2 \cos \varphi_2$

12 Fsin sin  $\varphi_1 \cdot \cos \varphi_2 \cdot \cos \varphi_1 \cos(\lambda_1 - \lambda_2) \cos \varphi_2 \cos \varphi_2$

13 IP2  $\varphi_2 \sin \varphi_1 \cos \varphi_2 \cdot \cos \varphi_1 \cdot \cos(\lambda_1 - \lambda_2) \cos \varphi_2$

14 Fsin sin  $\varphi_2 \sin \varphi_1 \cos \varphi_2 \cdot \cos \varphi_1 \cdot \cos(\lambda_1 - \lambda_2) \cos \varphi_2$

15 X sin  $\varphi_1 \cdot \sin \varphi_2 \cos \varphi_2 \cdot \cos \varphi_1 \cos(\lambda_1 - \lambda_2) \cos \varphi_2 \cos \varphi_2$

16 + sin  $\varphi_1 \cdot \sin \varphi_2 + \cos \varphi_2 \cdot \cos \varphi_1 \cdot \cos(\lambda_1 - \lambda_2) \cos \varphi_2 \cos \varphi_2 \cos \varphi_2$

17 Farccos arccos  $[\sin \varphi_1 \cdot \sin \varphi_2 + \cos \varphi_2 \cdot \cos \varphi_1 \cdot \cos(\lambda_1 - \lambda_2)] \cos \varphi_2 \cos \varphi_2 \cos \varphi_2$

18 1 1

19 1 11

20 1 111

21 , 111,

22 2 111, 2 arccos  $[\sin \varphi_1 \cdot \sin \varphi_2 + \cos \varphi_2 \cdot \cos \varphi_1 \cdot \cos(\lambda_1 - \lambda_2)]$

23 X 111, 2 arccos  $[\sin \varphi_1 \cdot \sin \varphi_2 + \cos \varphi_2 \cdot \cos \varphi_1 \cdot \cos(\lambda_1 - \lambda_2)]$  —

24 C/P



Программа закончена. Получилась она короче вашей более чем в полтора раза. Время ее создания практически не изменилось. То есть, сокращение достигнуто просто-напросто за счет технических навыков. Причем в этом примере и навыки-то касались одной стороны программирования — работы со стеком. Как видите, это не так уж мало. Не исключаю, что программу можно написать еще короче. Но не намного: две-три команды. Вообще тратить много времени на сокращение длины программы на пару команд — занятие непродуктивное. Однако знание принципов программирования, позволяющее с самого начала писать сравнительно короткие и быстродействующие программы, очень полезно.

Ну а пока вводи, Андрей, новую программу. Убедись в ее работоспособности. Тестом может быть все тот же путь из Керчи в Вологду. Потом вычисли расстояние от Ленинграда до Монтевидео.

Операции с калькулятором были проведены быстро. Мы узнали, что между нашей северной Пальмирой и аргентинской столицей лежит 13600 километров. Я посмотрела на часы.

— На сегодня хватит, Андрей.

Сын без особого энтузиазма пошел в свою комнату. Мы с мужем остались одни.

— Ну что, как игрушка? Смотрю и ты ужеходишь во вкус.

Пожалуй, Игорь был прав. Только на сегодня я уже устала. Я включила телевизор: программа «Время» заканчивалась. Погоду на завтра обещали холодную, и я сразу подумала: шерстяные вещи я вяжу всему семейству сама. Это мне нравится. Вот только постоянные творческие муки: какой узор делать?

— Ты о чем задумалась?

— Да так. Свитер новый вязать пора, а узор не знаю, какой выбрать. Журнальные мне не нравятся.

— Да, — вдруг вспомнила я, — ты говорил, что калькулятор может придумывать узоры для вязки, или ты шутил. На таком крохотном экранчике не только узор, но и линию не нарисуешь.

— Здесь не размеры экранчика, а твоя фантазия важна. Только вот думаю я, не рано ли тебе давать эту программу.

— А что она уже у тебя есть?

— Есть. Правда, я не придумал ее, а взял из журнала. Программа интересная и я хотел оставить ее на потом, чтобы объяснить, как она работает.

— Да не нужно мне объяснений ее работы. Оставь их для Андрея. Просто объясни, как ей пользоваться. В конце концов, как работает телевизор, я тоже не знаю, но смотрю же его. Расскажи, как работать с программой. Я получу узор, начну вязать, а потом, может, и разберусь в ней, если захочу.

— Наверное, ты права. Пользователю не обязательно-знать, как работает программа, лишь бы он умел работать с ней.

— И все-таки не могу поверить, что калькулятор способен рисовать. Ведь, кроме цифр, ничего он показать не может.

— Это верно. Но ты забыла, что любую информацию можно закодировать цифрами. И если уже на то пошло, всего двумя цифрами: 0 и 1. На этом основана вся вычислительная техника, в том числе и наш калькулятор. А закодировать цифрами цвета — нет ничего проще. Ведь цифр у нас десять. Это намного больше, чем число цветов шерсти, которые ты используешь одновременно при вязке изделия. Значит «цветную» строку на калькуляторе получить можно. Теперь представь себе, что ты рассматриваешь какую-нибудь картинку через узкое окошко. Пододвинула окошко к одному фрагменту — посмотрела, сдвинула окошко — посмотрела другой фрагмент. Конечно, смотреть таким образом «Джоконду» или «Явление Христа народу» не очень удобно, но для вязального узора этот способ вполне приемлем. Пойдем дальше. Допустим, закодированный узор записан в нескольких адресуемых регистрах нашего калькулятора. Можно вызвать содержимое одного регистра на индикатор, зарисовать его, затем поступить также со вторым, третьим... В результате можно получить такую, например, картинку:

```

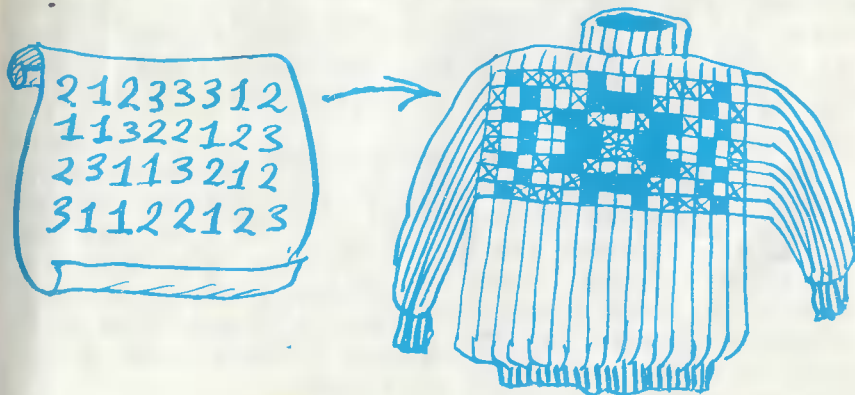
2 1 2 3 3 2 1 2
1 1 3 2 2 1 2 3
2 3 1 1 3 2 1 2
3 1 1 2 2 1 2 3

```

— Ну, как узор?

— Что-то я его не вижу.

— А я вижу. Чтобы сделать его нагляднее, достаточно заменить одинаковые цифры одинаковыми цветами. Для



— Вот и хорошо, — сказала Елена Николаевна. — Значит и я могу пойти решать свою задачу по приготовлению чая, а ты, Игорь, поставь чашки, достань сахар. Это будет твоя задача. — И Елена Николаевна вышла из комнаты.

Игорь Петрович начал накрывать на стол. — А не кажется ли вам, друзья, — прервал молчание Юрий Кузьмич, — что мы слишком увлеклись? В конце концов есть проблемы не менее важные. Например, как ставить и решать задачу, — и он посмотрел на Леонида Антоновича.

Но ответил ему Сергей Владимирович. — Действительно, постановка задачи сегодня становится узким местом. Программированием сейчас занимаются не только математики. И частенько медик, биолог или химик имеют доступ к машине, но не могут четко сформулировать то, что они от нее хотят.

Вот, к примеру. Вы знаете, что я гипертоник. Теперь меня стали лечить по-новому. Лекарств не дают, а дают физическую нагрузку — велоэргометр кручу. Мне при этом меряют давление и пульс до начала занятий, во время их и после небольшого отдыха. Метод пока недостаточно апробирован, поэтому врачи подробно изучают, как меняется самочувствие больного при тренировках. Цифры давления и пульса при изменении нагрузки, естественно, скачут и важно собрать статистику реакции различных людей на нагрузку. Кроме того, надо сравнить, как меняются эти параметры у здорового и больного человека.

— А что тут сложного, — вставил слово Андрей. — Дать одинаковую нагрузку больному и здоровому и посмотреть, как у них меняется пульс и давление.

— Правильно. Но что с чем сравнивать?

— Как что с чем? Конечно пульс с пульсом и давление с давлением.

— А с чего ты взял, что у всех здоровых людей одинаковое давление?

— Так ведь температура у всех примерно одинаковая, наверное, и пульс с давлением тоже.

— Это не совсем верно. Температура, кстати, у здоровых людей тоже разная. Медики считают для простоты, что температура менее  $37^{\circ}$  нормальная, а более — повышенная. Но у одного человека нормальная температура, скажем,  $36,2^{\circ}$ , для него  $36,8^{\circ}$  будет повышенной, а для другого нормальная —  $36,8^{\circ}$ ; для него, скажем,  $37,2^{\circ}$  будет еще в норме. Пульс и давление у разных здоровых людей тоже могут различаться.

— Но можно же взять среднее.

— Можно, — вмешался Юрий Кузьмич. — Но можно с этим средним и впросак попасть. Сейчас я вспомнил поучительную историю, которую рассказал мне мой друг Олег. Он математик и доктор экономических наук. Как-то раз к нему обратились за

помощью в таком деле. Существовали в одном городе два завода. Условия труда на них были примерно одинаковые, бытовые условия тоже. Одинаковой была и средняя зарплата. Однако почему-то с одного завода сотрудники постоянно уходили, на другом же стабильность работающих была образцовой. Обработал Олег данные по зарплате и получил интересные выводы. Средняя зарплата на обоих заводах действительно была одинаковой. Но распределение ее различалось. На неблагополучном заводе она колебалась от максимума до минимума очень незначительно. На хорошем разрыв был весьма большой. Поэтому на первом предприятии у сотрудников было весьма мало шансов «вырасти», вот они и покидали его. На втором же сотрудники были заинтересованы в повышении производительности своего труда и знали, что за хорошую работу они будут вознаграждены. Вот так с помощью статистики удалось разгадать секрет.

— И в правду поучительная история, — откликнулся Сергей Владимирович. — Обязательно расскажу ее своим медикам. А то ведь они, как и Андрей, склоняются к средним, не всегда подозревая, что нужно оценивать и другие параметры распределений. Но не только в статистических оценках дело. Вернемся к нашей медицинской проблеме. Измерения параметров пациента происходит в трех точках: до нагрузки, во время ее и после снятия нагрузки. При этом нормальной реакцией является повышение верхнего систолического давления и понижение нижнего диастолического. После снятия нагрузки эти цифры должны примерно подойти к норме, то есть к своим значениям до нагрузки, а может быть, и улучшиться. Но как оценить этот процесс количественно? Сравнение со средними цифрами, полученными при исследовании здоровых людей, как вы уже поняли, мало что дадут.

— А почему бы не использовать метод наименьших квадратов для оценки параметров кривой, проходящей через три точки для здоровых и больных людей? — спросил Игорь Петрович.

— Как раз к этому я и клоню, — ответил Сергей Владимирович. — Но чтобы использовать этот метод, надо как минимум знать о его существовании. Только в этом случае можно правильно сформулировать задачу и отдать данные для анализа программисту.

— Подожди, а зачем отдавать их программисту? — поинтересовался Игорь Петрович.

— Ну, хотя бы потому, что не в каждой медицинской лаборатории есть ЭВМ и не все медики умеют работать на ней.

— ЭВМ действительно есть далеко не везде. К тому же она обычно дороговата, да и требует специального ухода. Но вот микрокалькулятор, что я вам уже показывал, может приобрести не только каждая лаборатория, но и каждый исследователь, если он того захочет. А задачу, подобную описанной тобой,



## КАЛЕЙДОСКОП

	0	1	2	3	4	5	6	7	В	9
0	+	÷	п9	сх	п1	п2	п3	п4	пп	24
1	4	по	кип1	вп	4	кп1	FL0	12	пп	24
2	1	с/п	бп	03	4	по	9	п5	1	0
3	пв	ип9	1	1	х	Flп	+	п9	кип9	ху
4	ип9	—	п9	ип6	х	1	+	кп5	FL0	31
5	кипА	кипв	кипс	кипд	5	по	ипв	ипс	ипд	ипА
6	пп	78	ипс	ипд	ипА	ипв	пп	78	ипд	ипс
7	ип8	ипА	пп	78	ипс	ипв	ипА	ипд	п7	FO
8	ипв	х	+	ип8	х	+	ипв	х	ип7	+
9	кипо	+	кп1	В/О						

## Инструкция

1. п П6 (п — число цветов, используемых в узоре).
2. день час мин В/О.
3. С/П «1» ИП1 «1-я строка узора» ИП2 «2-я строка» ИП3 «3-я строка» ИП4 «4-я строка».
4. Перейти к п. 3.

Примечание. После получения очередного узора можно изменить число цветов, выполнив п.1. Затем перейти к п.3.

простоты, чтобы не идти за фломастерами, будем на месте единиц рисовать белые квадратики, на месте двоек — серые, а тройки покажем черным цветом. Получится так:

— Это уже на что-то похоже, — согласилась я.

— А кто тебе мешает использовать другие цвета? Красный, скажем, синий, зеленый. Калькулятор будет выдавать тебе сочетания цветов, а вопрос их подбора — дело вкуса. Дальше же поступать следует, как в занятой детской игрушке — калейдоскопе. Там из нескольких цветных стеклышек составляется маленькая картинка, вроде той, что мы нарисовали, а затем она многократно отражается в зеркале. Получаются красивые узоры, глаз от них не оторвешь. Только вот чуть-чуть шевельнул трубку и узор распадается, а получить вторично такой же практически невозможно. Здесь же картинка будет перед глазами любое время. Можно не спеша зарисовать ее, а можно и повторить.

Роднит эту программу с калейдоскопом огромное число узоров, которые можно получить. По желанию можно менять и число цветов, из которых будет строиться узор. Причем чем больше «цветов», тем больше вариантов. Для двух цветов можно получить всего 256 вариантов, для четырех — больше 65 тысяч, а для шести — вообще астрономическое количество — свыше полутора миллионов. Правда, вряд ли ты захочешь вязать шестцветный свитер. Поэтому думаю двух — четырех цветов будет вполне достаточно.

— Да, ничего себе, — только смогла выговорить я.

— Что, великовата? Но посуди сама. Нужно было записать алгоритм получения псевдослучайных чисел, запрограммировать процесс построения строк узора, а их четыре штуки, сформировать строки так, чтобы их было удобно рассматривать. Так что дел у программы достаточно.

— Но мы договорились, что о самой программе говорить не будем. Расскажи лучше, как ей пользоваться.

— Это просто. После ввода программы и перехода в режим вычислений необходимо ввести начальное число, которое «запустит» генератор псевдослучайных чисел. Чтобы пользователь не ломал голову над случайным выбором (опыт говорит, кстати, что пользователь вводит чаще всего какое-нибудь одно, ну два, поллюбившиеся ему числа), в программе предусмотрен ввод трех чисел. Но не совсем случайных: число месяца, часы и минуты. Такой ввод обеспечивает довольно малую вероятность повторения последовательности узоров.

Но еще раньше в регистр 6 нужно заслать желаемое количество цветов. Давай для начала ограничимся тремя.

Вводим 3 П6.

Теперь посмотрим на часы: сегодня 15-е число, 22 часа 26 минут. Вводим 15 ↑ 22 ↑ 26.

Остается нажать В/О С/П и калькулятор приступит к формированию узора. После остановки программы на индикаторе — первая строка узора, а три следующие в регистрах 2, 3 и 4. Извлекать их будем командами ИП2, ИП3 и ИП4.

Узор готов. Чтобы получить другую «картинку», достаточно нажать одну клавишу: С/П. Если нужно изменить число цветов, то перед нажатием этой клавиши его надо занести в регистр 6.

Работа увлекла. Каждые полминуты я получала новый узор. Я поймала себя на том, что и срисовываю их не всегда. Сам процесс вычислений, мигание индикатора, появление совершенно неожиданных последовательностей цифр — все это завораживало.

Вернул меня к действительности голос Игоря.

— Учти, что единственное недопустимое сочетание входных данных для программы — это 0 часов 0 минут. Запускать программу в полночь нельзя. А ты, кажется, приближаешься к этому часу. Так что не вздумай вводить новую дату.

Я посмотрела на часы. Ничего себе! А ведь я хотела еще заняться обедом на завтра. Так закончился вечер второй.



## Вечер третий

### Рассказывают авторы

Пришли гости — друзья Игоря Петровича. Калькулятор играет в чет-нечет. Что такое задача и как ее ставить? В помощь страдающим гипертонией. Правильно ли мы питаемся? Калькулятор вычисляет интеграл. Сергей Владимирович сажает ракету на Луну.

Сегодня вечером Игорь Петрович принимал гостей. Собралась старая компания — бывшие однокашники хозяина дома. Собирались они редко — раза два в году — и очень ценили эти встречи. Общение было, как теперь говорят, неформальное, то есть говорили все и обо всем. У каждого был свой конек, но понимали они друг друга с полуслова и всем было интересно.

В последнее время на этих встречах присутствовал Андрей. Сын вырос, и Игорь Петрович не без оснований полагал, что ему полезно побыть в кругу взрослых людей.

В этот вечер в центре внимания оказался калькулятор.

— А хочешь он обыграет тебя в какую-нибудь игру? — спросил вдруг Игорь Петрович у Леонида Антоновича.

— Надеюсь не в шахматы, — это вступил Сергей Владимирович. Он имел мастерский балл по шахматам и считался в компании гроссмейстером, во всяком случае знатоком шахматных проблем.

— В шахматы, пожалуй, нет. Да это и большой машине трудновато сделать. А вот в игру попроще, скажем в чет-нечет, пожалуй, и обыграет.

— Давай, — вызвался Юрий Кузьмич, — пусть она попробует меня обыграть.

Игорь Петрович взял тетрадку, нашел нужную программу и начал вводить ее в память калькулятора.

— Вот программа, — Игорь Петрович показал всем листок бумаги. — Желающие могут ознакомиться с ее текстом.

#### ЧЕТ — НЕЧЕТ

	0	1	2	3	4	5	6	7	8	9
0	ИП8	П9	ИП7	П8	+	ИП3	П7	+	ИП2	П3
1	+	ИП1	П2	+	ИП0	П1	+	6	+	ПД
2	ИП6	ИП5	—	ИП5	С/П	П0	КИП6	ИП4	FX=0	71
3	ИП1	2	x	ИП2	+	2	x	ИП3	+	2
4	x	ИП7	+	Kx#0A	1	—	Kx#0A	2	—	Kx#0B
5	2	—	Kx#0B	1	—	Kx#0A	3	—	Kx#0B	1
6	—	Kx#0A	2	—	Kx#0A	2	—	Kx#0B	1	—
7	Kx#0B	КИП4	ИПС	ВП	9	Fcos	Farccos	Fr	+	ПС
8	ИПД	—	Kx<0B	ИП0	FX=0	00	БП	91	ИП0	FX#0
9	00	КИП5	Cx	П4	БП	00				



## Инструкция

1. Сх П2 П5 П6 П9 ПС1 П0 П1 ПЗ П8 83 ПА 88 ПВ В/О С/П «0»
2. 0 или 1 (заданное число) С/П «число побед калькулятора» ХУ «число побед игрока».
3. Перейти к п.2.

Примечание. Переключатель Р — Г перед началом работы поставить в положение «Р».

— Программа, как видите, довольно велика, — сказал Игорь Петрович. — Из 98 ячеек программной памяти она занимает 96. Поэтому некоторые операции перед ее запуском приходится делать вручную. Они, правда, несложны. Нужно очистить регистры 2, 5, 6, 7, 9 и С, то есть занести в них нуль, потом записать единицу в регистры 0, 1, 3 и 8 и ввести еще два числа: 83 в регистр А и 88 в регистр В.

— Многовато, — сокрушенно сказал Юрий Кузьмич.

— А чему удивляться! Программа-то ни много, ни мало мышление моделирует. Думаю, в памяти каждого из вас сидит побольше чисел. Ну вот, все готово. Нажимаем В/О С/П. На индикаторе появился 0. Калькулятор готов к соревнованию.

Условия игры просты. После каждого останова надо ввести 0 или 1 и нажать С/П. Если калькулятор угадает набранное число, он прибавит единицу к счетчику своих побед. В противном случае увеличит на единицу счетчик игрока. Эти счетчики расположены в регистрах стека. Победы калькулятора — в РХ, они видны на индикаторе, ваши победы — в РУ. Чтобы посмотреть его содержимое, надо нажать клавишу ХУ. Вот и все. Начинай, Юра.

Очень скоро счет стал расти в пользу калькулятора. Тогда Юрия Кузьмича сменил Сергей Владимирович. Ситуация не изменилась.

— Эх, вы, мыслители, — вмешался Леонид Антонович, — дайте мне. Он достал из кармана монетку, стал подбрасывать ее и вводить единицу при «орле» и нуль при «решетке». Успехи калькулятора пошли на убыль.

— Так-то, — сказал он, — мой «калькулятор» не хуже твоего.

— Ничего не поделаешь, — согласился Игорь Петрович, — твоя взяла. При такой стратегии в чет-нечет проиграть нельзя. Правда, и выиграть тоже. А калькулятор ведь не только не про-



игрывает, он может и выиграть. Моя программа в отличие от твоего идеального датчика случайных чисел пытается учесть индивидуальные особенности игрока, не применяющего случайную стратегию. У игрока всегда присутствует элемент «воли», склоняющий его к некоторой периодичности в загадывании чисел. Вот этот волевой фактор и пытается определить программа и использовать при своих ответах. Как видите, это ей удается. Но, впрочем, хватит, — он отложил калькулятор.

— Какие новости? — обратился Юрий Кузьмич к присутствующим. — Леня, ты на новую работу перешел?

— Да, я теперь занимаюсь экономикой в геологии, точнее, работаю над созданием АСУ в этой области.

— А что, электроника тебе уже наскучила? — спросил Сергей Владимирович.

— Не то чтобы наскучила. Но я ведь давно уже стал заниматься не столько физикой, сколько программированием. Вот и захотелось попробовать себя в этой области всерьез. Интересно, что с самого начала я столкнулся не только с новой тематикой, но и с новой терминологией.

— Например? — поинтересовался Игорь Петрович.

— Ну, хотя бы «отрешать задачу».

— Действительно, — сказал Сергей, — не очень понятно. Решить задачу — это ясно, а «отрешать»? Ее что, каждый день решают?

— Да, есть задачи, которые решают каждый день, есть еженедельные, ежемесячные.

— Ну, извини. Зачем же решать каждый день одну и ту же задачу? В том-то и смысл решен я, что прочел его один раз и получил то, что надо. Мне это ежедневное решение кажется нелепым. Это все равно, как если бы я каждый день решал задачу о распределении электронов в полупроводнике или, чтобы Андрею было понятней, о встрече двух путников, вышедших из пунктов А и Б. Стал бы ты, Андрей, решать такую задачу каждый день?

— Конечно нет, — ответил Андрей, — ее надо решить один раз, записать решение в виде формулы, а потом только подставлять в нее разные числа.

— Правильно, — похвалил его Сергей Владимирович, — а твои экономисты, Леня, как считают?

— Прежде всего, боюсь, что у тебя, да и у Андрея тоже, несколько однобокое представление о том, что такое задача. Ты и Андрей оказались в плену привычных представлений. Ну-ка, скажи, Андрей, что такое по-твоему «задача».

— Задача, это... это, когда есть условие, например камень бросают вертикально вверх с такой-то скоростью, и надо что-то определить, например на какую высоту поднимется камень. Или когда есть уравнение и надо найти неизвестное.

— Молодец Андрей, — поддержал своего союзника Сергей Владимирович. — Сформулирую твою мысль почетче. Задача — это некоторая проблема, ситуация, когда даны определенные начальные условия, законы, по которым ситуация может меняться, а определить требуется параметры некоторой конечной ситуации.

— В общем, вы оба правы, — вмешался Игорь Петрович, — только позволяю себе еще более уточнить понятие. Недавно я листал книгу Джорджа Пойа «Математическое открытие». Там мне понравилось определение понятия «задача». Я сейчас вам его зачитаю.

Он подошел к книжной полке, достал книгу, быстро нашел нужную страницу и стал читать.

«Слово «задача» мы будем употреблять дальше в весьма широком смысле; поэтому прежде всего уточним, что будет подразумеваться под этим словом.



При современном укладе жизни добывание пищи обычно не представляет собой задачи. Если я проголодаюсь дома, то тащу что-нибудь из холодильника, в городе же — иду в какое-нибудь кафе или закусочную. Однако совсем другое дело, когда холодильник пуст или когда я оказываюсь в городе без денег, в таких случаях желание поесть приводит к задаче, иногда достаточно трудной. Вообще говоря, желание может иногда приводить к задаче, а иногда — нет. Если одновременно с желанием в моем мозгу сразу же, без какого бы то ни было усилия, возникает очевидное средство, с помощью которого, наверное, можно осуществить это желание, то задача не возникает. Если же такого средства нет, то это — задача. Таким образом, задача предполагает необходимость сознательного поиска соответствующего средства для достижения ясно видимой, но непосредственно недоступной цели. Решение задачи означает нахождение этого средства»<sup>1</sup>.

<sup>1</sup> Пойа Дж. Математическое открытие. — М.: Наука, 1976. — С.143.

Все замолчали.

— Ну как, Андрей, — обратился к мальчику Леонид Антонович, — все понял?

— Сначала вроде да, а потом не очень.

В это время в комнату вошла Елена Николаевна.

— Ну, академики, чай пить будете?

— Стоп, — сказал Юрий Кузьмич, — приготовление чая — это задача или не задача? Тут, Лена, твой муж прочел нам интересное определение этого термина. Игорь, повтори-ка для Лены начало цитаты.

— Ага, — сказала Елена Николаевна, прослушав чтение, — кажется я поняла. Если я пойду, залью чайник водой из-под крана и поставлю его на плиту, то какая же это задача? А вот, если выключат воду и перекроют газ, тогда приготовить чай — задача. Так?

— Получается, что так, — согласился Игорь Петрович.

— А как же понять, — вмешался Андрей, — что в «Науке и жизни» помещен алгоритм приготовления чая, где используются вода из-под крана и газовая плита? Ведь, если это не задача, то для чего тогда этот алгоритм?

— Вот это да! — обрадовался Леонид Антонович. — Молодец Андрей! Дай-ка мне, пожалуйста, «Англо-русско-немецко-французский словарь по вычислительной технике», вон он на полке лежит. Название сразу и не выразишь, а определения там компактные. Спасибо. Читаем, «task — задача. Основной объект выполнения работ в ЭВМ, требующий выделения ресурсов: памяти...» и т. д. Как это согласовать с определением Пойа?

— Друзья, — примирительно сказал Юрий Кузьмич, — не будем ссориться. Что поделать, мы живем в такое время, когда с одной стороны все употребляемые термины требуют строгого определения, а с другой — эти определения непрерывно меняются.

А что касается Пойа, дадим ему право на оригинальность. Ведь Ричард Фейнман, к примеру, упорно не считает математику наукой. Не подходит оно под его определение и все. Ну и что?

Посему, чтобы понимать друг друга, давайте договоримся. Будем считать, что задача — это то, что требует разрешения. Не очень строго, конечно, зато избавит нас от схоластических толкований. Кстати, именно так определяется слово задача в словаре Ожегова.

С этой точки зрения, твои геологи, Лена, совершенно правы. Они ежедневно решают одни и те же задачи, связанные с получением результатов по изменяющимся начальным данным. Точнее, проводят расчеты по одной и той же программе.

— Верно. Однако под задачей понимается не только программа, но и совокупность входной и выходной информации, способ ее хранения и вид выходных результатов.



## АППРОКСИМАЦИЯ ПАРАБОЛЫ

	0	1	2	3	4	5	6	7	8	9
0	КНОП	С/П	П2	ХУ	П3	1	2	ПО	КИП4	4
1	П1	ИП3	ПП	14	↑	КИПО	+	КП1	ХУ	ИП3
2	х	FL1	14	3	П1	ИП2	В/О	В	6	ПД
3	ИПА	ИП9	ИП9	ИПВ	÷	П2	КППД	П1	ИП4	ИПВ
4	ИП2	ИПА	КППД	ПВ	$Fx^2$	ИП1	÷	—	ИПА	ИПА
5	ИПВ	÷	П3	КППД	П4	ИП6	ИП2	ИП5	КППД	П6
6	ИП7	ИП3	ИП5	КППД	ИП6	ИПВ	х	ИП1	÷	—
7	ИП4	÷	ПО	ИПВ	КППД	ИП1	÷	П1	ИП5	ИП8
8	÷	ИП2	ИП1	КППД	ИП3	ИПО	х	—	ПС	В/О
9	1	2	ПО	Сх	КП1	FL0	94	С/П		

## Инструкция

1. БП 90 С/П «0».
2. В/О С/П «0».
3.  $x_i$   $y_i$  С/П.
4. Если исходные данные исчерпаны, то перейти к п. 5, иначе — к п. 3.
5. БП 27 С/П «а» ИП1 «в» ИПО «с» (а, в, с — коэффициенты аппроксимирующей параболы:  $y = ax^2 + bx + c$ ).

решить ей по силам. Больше того, программы для решения этой задачи уже существуют. Могу предложить твоим медикам ту, что опубликована в сборнике «Прикладные программы для микроЭВМ» авторов А. Н. Цветкова и В. А. Епанечникова<sup>1</sup>.

— Передай, Сергей, эту программу твоим медикам. Калькулятор они, безусловно, смогут достать. А тебе, Андрей, советую разобрать ее на досуге. В ней много поучительного. Например, автоматическая очистка адресуемых регистров перед началом работы, использование косвенно-адресного обращения к подпро-

граммам для сокращения длины программы, передача управления на начало программы, точнее на адрес 01 (потому-то первой записана «пустая» команда КНОП), с помощью команды В/О и многое другое. Обрати внимание, что эти приемы использованы не для красоты, а по необходимости — без них эта сложная программа просто не уместилась бы в микроскопической памяти микрокалькулятора.



<sup>1</sup> См.: Цветков А. Н., Епанечников В. А. Прикладные программы для микроЭВМ «Электроника БЗ-34», «Электроника МК-56» и «Электроника МК-54» — М.: Финансы и статистика, 1984. — С. 97.

— Ладно. Все это очень интересно, но если так и дальше пойдет, то, глядишь, и медики нас начнут лечить с помощью программ. Представляете, приходишь к врачу. Осмотрел он тебя и выписывает на фирменном бланке... программу для компьютера, приговаривая: «Придете домой, введите эту программу перед сном, остальное — она скажет вам сама».

Пока друзья обсуждали в таком же духе перспективы компьютеризации процесса лечения, в комнату вошла Елена Николаевна с чайником в руке. — Опять научный диспут? Весь стол бумажками заложили, а куда мне чайник поставить и пирсг?

— Убираем, убираем, — засуетились гости и через минуту пространство для чая с пирогом было готово.

В то время, когда Игорь Петрович резал ароматный, горячий пирог, а Елена Николаевна разливала чай, Леонид Антонович продекламировал:

Разлитый Ольгиной рукою,  
По чашкам темною струею  
Уже душистый чай бежал  
И сливки мальчик подавал.

— А, кстати, — продолжил он, — полезно ли пить чай со сливками?

— Моя жена ни за что не стала бы это делать, — сказал Юрий Кузьмич, помешивая чай ложечкой. — Она убеждена, что отдельные виды еды и питья смешивать не следует. Где-то когда-то прочитала, что это вредно. Нужно каждый раз есть что-нибудь одно. Скажем, только мясо, потом только картошку или только хлеб. Пить следует тоже или молоко, или чай, но не вместе.

— Слышал я про такую теорию, — скептически сказал Сергей Владимирович. — Только почему-то у всех народов кулинарные книги пестрят всякими салатами, грибными подливами к мясным блюдам, не говоря уже о всяких там чебуреках, беляшах, да и наших родных кулебяках и расстегаях.

— Да, — вступил в разговор Игорь Петрович, — вообще проблема питания, по-моему, выходит на одно из первых мест среди тем для обсуждения. Уступает она, пожалуй, только прогнозу погоды.

Существуют сотни диет. Для похудения и для набора веса, для понижения давления и для его повышения. Обсуждаются специфические особенности рациона питания для похода на Северный полюс и экспедиции в джунгли...

— Выбор рациона — это тоже задача и ее надо правильно поставить, а то могут получиться забавные казусы, — снова перехватил нить разговора Леонид Антонович. — Попросил меня как-то один мой приятель, заядлый турист, рассчитать ему набор продуктов для турпохода. При этом он поставил такие условия: при заданной калорийности продуктов цена их должна быть

минимальной. Сделал я программу, ввел его данные и получил, чтобы затратить минимум денег, лучше всего брать одну манную крупу. Но зато, чтобы нести ее, нужна бригада носильщиков. А если стремиться к уменьшению веса, то вполне можно обойтись одним сахаром, правда, стоять это будет дороговато, да и кто будет есть его целыми днями?

Но самая главная ошибка в формулировке задачи была в том, что не одними калориями сыт человек. Нужны ему и белки, и жиры, и углеводы.

— Между прочим, — вмешалась в разговор Елена Николаевна, — сколько я тебя, Игорь, просила составить такой рацион для меня и вообще для семьи?

— А я не забыл. Только составил я не рацион, а программу для проверки рациона. Не стоит все-таки тебя лишать удовольствия проявлять творческий подход в выборе блюд. А проверить свой выбор ты сможешь с помощью

программы. Во всяком случае с ее помощью ты сможешь проследить, правильно ли мы питаемся, чего нам не хватает и что в избытке. Идет?

— Давай, давай, — подбодрил хозяина Леонид Антонович, — может, и мне пригодится. Да и сослуживцам удружу. Покажи программу-то. Так, так... вроде я ее где-то видел. Не в «Науке и жизни» ли?

— Ну и память у тебя. Как шахматист все партии запоминает, так и ты программы. Действительно, была в журнале подобная программа. Я немного изменил ее, сократив на пару команд, и чуть-чуть упростил работу с ней. Так что можем проверить.

Вновь калькулятор появился на столе. Игорь Петрович включил его и сам стал вводить программу.

#### ОПТИМАЛЬНЫЙ РАЦИОН

	0	1	2	3	4	5	6	7	8	9
0	†	7	+	5	x	XU	2	F10 <sup>x</sup>	—	П1
1	пп	44	ип2	ип1	1	0	x	+	пп	45
2	сх	па	пв	пс	п2	с/п	по	FO	ипс	+
3	пс	FO	ипв	+	пв	FO	ипа	+	па	ипо
4	ип2	+	бп	24	x	п2	0	,	1	5
5	x	пс	4	,	5	+	пв	ип2	3	3
6	,	3	÷	1	0	+	па	ип2	с/п	в/0

#### Инструкция

1. рост † код вида деятельности в/о с/п «К» (калории) мин» ИПА «Б (белки) мин» ИПВ «Ж (жиры) мин» ИПС «У (углеводы) мин» с/п «К макс» ИПА «Б макс» ИПВ «Ж макс» ИПС «У макс» с/п «0».

2. Б † Ж † У † К (ввод параметров продукта, см. табл. на с. 60—61) с/п «К сумм» ИПА «Б сумм» ИПВ «Ж сумм» ИПС «У сумм».

3. Перейти к п. 2.

Примечание. Для исключения продукта из рациона ввести его параметры со знаком «—».

Закончив ввод и прогнав контрольный пример, Игорь Петрович очень серьезно произнес: «Начнем, пожалуй! Учтите только, программа эта предназначена для индивидуального пользования и перед началом работы требует ввода некоторых «анкетных» данных. Кто первый?»

— Леди вне очереди, — Леонид Антонович посмотрел на Елену Николаевну.

— Так, — начала она, родилась я в мае, в семье служащего.

— Ну, ну, не так подробно, — прервал ее Игорь Петрович. — Для программы нужны только твой рост и род занятий.

— А ты, что — не знаешь?

— Знаю. Рост — 164 сантиметра. Род занятий — умственный. Сейчас поясню, зачем это нужно, — продолжил Игорь Петрович. Для определения рациона надо знать род деятельности человека. Понятно, что шахтер должен потреблять больше калорий, чем, скажем, учитель. Поэтому вводится коэффициент тяжести труда.

— И как же зависит коэффициент от рода занятий?

— Я не диетолог, поэтому следовал указанию авторитета, а именно взял коэффициенты, как их рекомендует В. И. Воробьев в книге «Калькулятор «рацион». Для простоты я к тому же ограничил диапазон выбора такими величинами:

1 — служащие и вообще люди «нефизического» труда;

2 — механизированный труд: рабочие-станочники;

3 — частично механизированный труд: сталевары, строители;

4 — тяжелый физический труд: землекопы, грузчики и др.

Допускаются и промежуточные коэффициенты. Например, слесарь, работающий в научно-исследовательском институте, может иметь коэффициент 1.5.

— А что вычисляет твоя программа? — нетерпеливо спросила жена. — Может быть, она дает закодированное меню: скажем, 24 — яйца всмятку, 41 — курица жареная, 87 — рыба с соусом по-польски?

— Нет, до такого совершенства программа на нашем «семейном компьютере» еще не дошла. Она сначала вычисляет максимальное и минимальное суточное число килокалорий, а также белков, жиров и углеводов, потребление которых гарантирует пребывание в рамках допустимого веса. За идеальный же вес я принял вес в килограммах, равный разности роста в сантиметрах и числа 100.

— Только почему «вес»? — сказал Сергей Владимирович, во всем любивший точность. — Речь может идти только о массе. Она действительно измеряется в килограммах, а, Андрей?



— Да, — поддержал его школьник, — а вес измеряется в ньютонах.

— Ну, пожалуйста, можно массу. Только это, хоть и правильнее, но в разговорной речи непривычно. Поэтому я, с вашего позволения, буду говорить все-таки о весе человека.

Итак, вводим рост и род занятий, разделяя их стрелкой. Для тебя, Лена, это будет 164 ↑ 1. Затем, как обычно, нажимаем В/О, С/П.

Через несколько секунд мелькание цифр на индикаторе прекратилось и появилось число 2560.

— Это минимальное суточное потребление килокалорий. Можно извлечь минимальные цифры для белков: для этого нажимаем ИПА.

На индикаторе появилось 87.

— Теперь тоже самое для жиров. Число это хранится в РВ.

Игорь Петрович нажал ИПВ и получил В6. Далее он извлек из РС минимальное количество углеводов 384.

— Для получения максимальных значений этих величин достаточно нажать С/П и после остановки калькулятора выполнить те же манипуляции.

На индикаторе последовательно появились: 3200 (килокалорий), 106 (белки), 107 (жиры) и 480 (углеводы).

— Если поддерживать суточные потребления этих веществ в полученных пределах, то и вес будет держаться в норме. Не надо будет предпринимать усилий ни для того, чтобы похудеть, ни для того, чтобы поправиться. Ясно?

— Ничего не ясно, — ответила Лена. — Мне нужно конкретно знать, сколько мяса, рыбы, картофеля полагается съесть за день. А ты мне даешь какие-то абстрактные числа.

— Правильно. Я еще не успел сказать, что дальше нужно вводить параметры употребленных за день продуктов, а программа будет подсчитывать их суммы. Подбирая определенным образом продукты, надо стремиться к тому, чтобы суммы эти не выходили за рассчитанные ранее пределы.

Брать параметры продуктов можно из таблицы, опубликованной в «Науке и жизни».

Гороховый суп	7	6	35	23
Суп с фрикадельками	16	23	0	18
Рассольник	6	11	27	25
Солянка	23	30	15	40
Борщ	10	12	35	25
Щи	5	11	14	18
Тушеное мясо с картофелем	18	12	35	31
Котлеты	7	12	45	35
Яичница с колбасой	15	30	1	32
Гуляш	15	10	6	17
Шашлык	20	17	0	24
Пельмени	20	12	40	40
Голубцы	30	30	25	47

Рыбные блюда	17	11	4	18
Овощные блюда	17	11	40	20
Блинчики с мясом	23	15	40	85
Овсяная, манная каша	15	25	60	43
Рисовая, гречневая каша	6	14	45	30
Компот	0	0	35	18
Кофе с молоком, какао (1 ст.)	5	0	30	20
Стакан молока	3	6	10	12
Стакан кефира	3	10	6	12
Бутерброд	5	7	15	14
Сосиски	11	25	0	25
Колбаса копченая	15	33	0	36
Колбаса докторская	14	23	0	25
Яйцо	11	2	0	3
Курица (100 г)	18	13	1	20
Язык (100 г)	13	10	0	15
Говядина (100 г)	20	10	0	16
Свинина (100 г)	13	40	0	40
Хлеб (100 г)	7	1	45	22
Печенье (100 г)	7	12	23	40
Шоколад (100 г)	8	35	100	55
Сметана (100 г)	3	5	25	25
Сыр (100 г)	25	27	0	35
Овощи, фрукты (100 г)	2	0	10	55

Ассортимент можно расширить. Шифр для каждого продукта определяется так: первое число — количество белков, второе — жиров, третье — углеводов (все в граммах), четвертое — калорийность в килокалориях.

Хочешь, для тренировки подсчитаем сегодняшний рацион?

— Я попробую это сделать сама, — ответила Елена Николаевна. Она взяла калькулятор и вышла из комнаты.

— Компьютеризация твоей семьи, — сказал Юрий Кузьмич, — идет полным ходом. Пожалуй, это мне нравится. А последняя программа — вообще чудо. Я чувствую, мне тоже стоит приобрести такой калькулятор.

— Приобретай, приобретай. Он тебе пригодится.

— Мне? Может быть, для работы? Правда, я не представляю, как в него втиснуть программу, для которой мне с трудом хватает ЕС ЭВМ. А может быть, твой калькулятор поможет обработать выборку из тысячи чисел, оценив попутно распределение? Предложи его лучше Леониду. Он занимается проблемами базы данных. Так, может быть, он создаст с помощью твоей «суперЭВМ» ее «карманный вариант»?

Все весело рассмеялись.

— А что других расчетов у тебя не бывает? — невозмутимо спросил Игорь Петрович и, не дожидаясь ответа, продолжил. — Недавно я готовился к лекции и просматривал старые журнальные статьи по теории транзисторов. В одной из них я нашел формулу для коэффициента устойчивости транзистора в схеме с общей базой.

Игорь порывлся в папке на письменном столе и выложил на стол лист бумаги с формулой

$$C = \frac{\sqrt{1 + 2\beta \cos \varphi + \beta^2} \sqrt{\beta^2 + 2\beta \cos \varphi}}{X(1 + 2\beta \cos \varphi + \beta^2) - \beta \sin \varphi}.$$

— Может ли кто-нибудь из вас сказать, как ведет себя величина  $C$  при изменении параметра  $\beta$ ? Я имею в виду не предельные случаи, когда  $\beta$  стремится к нулю или бесконечности, а некоторый средний диапазон.

— Ну, с ходу этого не скажешь, — задумчиво протянул Юрий.

— А с помощью калькулятора ответ на этот вопрос можно дать довольно быстро. Составление программы и ее отладка при некотором навыке не займут и получаса. Примерно столько же займут расчеты графика в требуемом диапазоне. Причем заметьте, сделать это можно дома или на работе, не отходя от рабочего стола и не выключивая машинное время на институтской ЭВМ. Кроме того, составить программу на Фортране ты сможешь минут за десять. Набрать программу, транслировать ее, построить задачу — еще минут пятнадцать, да двадцать минут на отладку. Решение задачи займет, правда, считанные минуты. Так что даже при идеальном стечении обстоятельств использование большой машины для решения маленькой задачи по сравнению с калькулятором дает микроскопический выигрыш во времени. Ну а при неидеальном: то машина доступна не сразу, то операционная система стоит другая, то диск чужой... В общем, чаще всего решая небольшую задачу на калькуляторе, суммарное время от постановки задачи до получения результата, как правило, меньше. А ведь таких задач предостаточно.

— Ну, убедил, — с шутливым уважением произнес Леонид Антонович. — Завтра все прямо с утра за калькуляторами, а то к вечеру разберут.

— Напрасно иронизируешь. Желающих их купить намного больше, чем годовое производство. Звоните в специализированный магазин-салон.

— Нет, Игорь, кроме шуток, лично мне микрокалькулятор действительно пригодится, — вступил в разговор Юрий Кузьмич. — Постоянно, когда я решаю диффуры и довожу решение до квадратур, проанализировать решение бывает очень трудно.

Здесь нам надо прервать разговор и сказать, как это принято в детективных романах, что дальше пошел разговор профессионалов, непонятному непонятный. Поясним только, что диффурами на их языке называются дифференциальные уравнения, а квадратурами — их решения, доведенные до интегралов.

— Я с этими проблемами тоже сталкиваюсь, — между тем отвечал другу Игорь Петрович. — У меня, кстати, есть целый пакет таких программ. Если хочешь, то, пока Сергей обсуждает перспективы будущего шахматного матча, я тебе покажу свои достижения в интегрировании.

Начал я с самого простого, то есть составил программу для

# ИНТЕГРИРОВАНИЕ (ФОРМУЛА ПРЯМОУГОЛЬНИКОВ)

	0	1	2	3	4	5	6	7	8	9
0	ПО	ИПВ	ИПА	П9	—	ХУ	÷	П1	Сх	П2
1	ИП9	ПП	25	ИП2	+	ИП9	÷	ИП1	+	П9
2	FL0	09	ИП1	x	С/П	.....	ПОДПРОГРАММА	Y(X)...	Y(X)...	В/О

## Инструкция

1. В режиме программирования записать подпрограмму, вычисляющую значения интегрируемой функции  $y(x)$ , начиная с адреса 25 (аргумент — в  $PX$ , результат помещать в  $PX$ . Не использовать регистры 0, 1, 2, 9, А, В).

2. а ПА б ПВ.

3. п В/О С/П «интеграл».

4. Для повторения расчета с новыми значениями: п — перейти к п. 3; а и б — перейти к п. 2.

5. Для вычисления интеграла от другой функции: БП 25 ФПРГ и перейти к п. 1.

вычисления определенных интегралов методом прямоугольников. Формула, которую надо запрограммировать в этом случае, такова:

$$\int_a^b y(x) dx \approx \frac{b-a}{n} [y_0 + y_1 + \dots + y_{n-1}].$$

Программу я составил быстро. Сейчас я вам ее покажу.

Вычисление подынтегральной функции вынесено в подпрограмму, поэтому для перехода от расчета одного интеграла к расчету другого нужно только изменить команды этой подпрограммы (начинается она с адреса 25). Последовательность действий при работе с этой программой такова:

ввести программу;

25. Значение  $X$  брать из регистра  $X$  стека. Туда же помещать результат. При записи алгоритма этих вычислений не использовать регистры 0, 1, 2, 9, А, В;

ввести границы промежутка интегрирования: аПА, бПВ;

ввести числа точек разбиения: пВ/О С/П;

вывод: результат на индикаторе.

Я не стал писать, какие действия следует предпринимать для продолжения работы. Мне кажется, это понятно. Чтобы пересчитать интеграл в тех же пределах для другого разбиения, следует перейти к п. 4, тот же интеграл в других пределах — к п. 3, и, наконец, интеграл от новой функции — к п. 2.

— Это неуважение к пользователю, Игорь, — вмешался Юрий Кузьмич. — Все действия оператора описывать нужно всегда. Тем более, что все равно тебе пришлось дополнять эту инструкцию устным разъяснением.

— Виноват. Совершенно с тобой согласен. Но давай продолжим. Выбрал я известный интеграл



$$\int_0^1 \frac{dx}{1+x^2} = \frac{\pi}{4} \approx 0.78539815$$

и стал смотреть, с какой точностью и за какое время будет считать его моя программа. После ввода подпрограммы 25. Fx<sup>2</sup> 26. 1 27. + 28. F1/X 29. B/O я начал заполнять таблицу.

Кстати, я думаю, понятно, почему в инструкции записано, какие регистры нельзя занимать при записи подпрограммы. Ведь алгоритм расчета подынтегральной функции может оказаться достаточно сложным и при его записи целесообразно использовать адресуемые регистры. Так вот, чтобы не испортить содержимое регистров, используемых в основной программе, их занимать в подпрограмме не следует.

Но вернусь к таблице. Получилась она такой:

Число разбиений	Интеграл	Время, с
2	0.9	14
4	0,84529412	25
6	0,8259074	35
10	0,8099815	56
20	0,797794	109

Конечно, интеграл для отладки выбран не из легких. Сам знаешь, эта функция нелинейная. Но все равно точность уж слишком мала. Время расчетов зависит от числа разбиений почти линейно, то есть прямо пропорционально числу точек разбиения; чтобы достичь более или менее приемлемой точности, пришлось бы брать слишком много точек.

Поэтому я решил попробовать формулу трапеций:

$$\int_a^b y(x)dx \approx \frac{b-a}{n} \left\{ \frac{y_0}{2} + y_1 + y_2 + \dots + y_{n-1} + \frac{y_n}{2} \right\}.$$

Изменений в программу пришлось вводить совсем немного и она приобрела такой вид:

ИНТЕГРИРОВАНИЕ (ФОРМУЛА ТРАПЕЦИЙ)										
	0	1	2	3	4	5	6	7	8	9
0	ПО	ИПВ	ИПА	П9	—	ХУ	÷	П1	КИПО	ИПА
1	ПП	34	П2	ИПВ	ПП	34	ИП2	+	2	÷
2	П2	ИП9	ИП1	+	П9	ПП	34	ИП2	+	FL0
3	20	ИП1	х	С/П	.... ПОДПРОГРАММА Y (X)				....	В/0

#### Инструкция

Инструкция та же, что и к программе «Интегрирование (формула прямоугольников)», только ввод подпрограммы следует начинать с адреса 34.

Инструкция осталась прежняя, разве что поменялся начальный адрес ввода подпрограммы. Он стал равен 34. Но таблица, подобная предыдущей, изменилась существенно.

Число разбиений	Интеграл	Время, с
2	0.775	14
4	0.78279412	25
6	0.78424073	35
10	0.7849815	52
20	0.7852794	103

Так что, как видишь, при том же времени расчета точность удалось существенно увеличить.

Но я решил пойти дальше и посмотреть, как поведет себя на калькуляторе формула Симпсона, или метод парабол:

$$\int_a^b y(x)dx = \frac{b-a}{3n} [y_0 + 4y_1 + 2y_2 + 4y_3 + \dots + y_n]$$

Программа опять-таки изменилась незначительно, да и инструкция осталась прежней. Разве что начальный адрес подпрограммы чуть увеличился и стал равен 39. Впрочем еще одно примечание: величина  $n$  должна обязательно быть четной.

Что же касается таблицы результатов, то она изменилась очень сильно.

Число разбиений	Интеграл	Время, с
2	0.78333333	23
4	0.78539216	37
6	0.7853979	60
10	0.78539816	77

На первый взгляд кажется, что время счета увеличилось. Но ведь если ориентироваться не по числу точек, а по достигнутой точности, это время существенно сократилось.

ИНТЕГРАЛ ПО СИМПСОНУ										
	0	1	2	3	4	5	6	7	8	9
0	ПО	ИПВ	ИПА	П9	—	ХУ	÷	П1	ИПА	ПП
1	39	П2	4	П3	ИП9	ИП1	+	П9	ПП	39
2	П4	ИП3	х	ИП2	+	П2	6	ИП3	—	FLO
3	13	ХУ	ИП4	—	ИП1	х	3	:	С/П	....
4	.... ПОДПРОГРАММА Y (X)				.... В/0					

#### Инструкция

Инструкция та же, что и к программе «Интегрирование (формула прямоугольников)», но подпрограмма начинается с адреса 39.

Суди сам, при 10 точках за время чуть больше минуты мне удалось получить результат с семью верными знаками! Это почти предел возможностей калькулятора независимо от алгоритма.

— То, что ты мне рассказал и показал, очень интересно. Я представил себе, что демонстрация таких программ студентам при рассказе о численных методах позволит гораздо лучше разобраться в них. Это как демонстрация эксперимента при изучении физических законов, своего рода наглядное пособие по математике. Во всяком случае, я надеюсь заинтересовать этим своего младшего брата. Он учится на втором курсе. Думаю, что вообще постоянное использование калькулятора при изучении численных методов довольно перспективно.

Ну а сам я, пожалуй, возьму твою программу интегрирования по Симпсону. Мне она понравилась и, думаю, поможет мне сэкономить массу времени.

В комнату вошла Елена Николаевна.

— Вот, полюбуйтесь, — воскликнул Леонид Антонович, — хозяйка ХХI века.

Вид у Елены Николаевны, и правда, был забавный. В одной руке она держала чайник, в другой — калькулятор.

— Ладно, ладно, — смутилась Елена Николаевна. — Не хотите чаю, я обратно унесу.

Хотим, хотим, — ответил за всех Игорь Петрович. — А что ты успела посчитать, пока чайник закипал?

— Я убедилась, что калькулятор считает правильно. Он полностью подтвердил целесообразность моего завтрашнего меню. Оказалось, что оно приемлемо для всех. Для тебя, Игорь, правда, ближе к минимуму. Из-за роста, как я поняла. Для нас с Андреем — к максимуму. Но все в пределах нормы.

— Насколько я понимаю, — сказал Сергей Владимирович, — калькулятор у вас и впрямь превратился в атрибут семейного быта. А, скажи-ка, Андрей, ты тоже достиг каких-нибудь успехов в программировании.

— А как же! Хотите я вам покажу программу, которую мы с ребятами из класса написали? Это игра «Посадка на Луну». Можете сыграть.

— Давай, Андрей, твою программу, — вызвался Сергей Владимирович.

Андрей вышел из комнаты и через пару минут возвратился с тетрадкой.

— Ого! — с уважением сказал Сергей Владимирович. — Это что, полная тетрадь твоих программ?



# ПОСАДКА НА ЛУНУ

0	1	2	3	4	5	6	7	8	9
0 Сх	П4	ВП	3	ПА	П8	2	÷	ПС	1
1 5	0	ПВ	1	.	6	2	П9	Сх	С/П
2 ИПВ	ИПА	С/П	ПО	КИП4	ХУ	ИПС	ХУ	—	Fx<0
3 36	Сх	ПО	ИПС	БП	37	ПС	ИПВ	÷	2
4 +	ИПО	ХУ	÷	ИП9	ХУ	—	ИПВ	+	ПВ
5 ИПА	ХУ	—	ПА	Fx<0	20	/-/	ИПВ	Fx²	F√
6 +	1	+	ПД	КИПД	ИПД	9	F1/X	+	С/П

## Инструкция

1. В/О С/П «0»
2.  $h_0$  (начальная высота корабля) ПА  $v_0$  (начальная скорость) ПВ.  $m_0$  (исходная масса топлива) ПС  $g$  (ускорение свободного падения для планеты) П9.
3. С/П «h» ХУ «v» ИПС «m» ИП4 «f» (текущие значения).
4. S (расход топлива в следующую секунду).
5. Перейти к п. 3 и продолжать до посадки корабля.

Примечания: 1. Выполнять п. 2 не обязательно. По умолчанию предполагается:  $h = 1000$  м,  $v_0 = 150$  м/с,  $m_0 = 500$  т,  $g = 1,62$  м/с² для Луны.  
2. Смысл сообщений о качестве посадки расшифрован на с. 68.

— Не только моих. Здесь и программы моих друзей, и те, что я выписываю из книг и журналов. Но тетрадка это только для игровых программ. Расчетные программы у меня в другой тетрадке.

Андрей перелистал тетрадку, нашел нужную программу и начал вводить ее в калькулятор.

После ввода программы Андрей, точно так же, как это делал его отец, прогнал контрольный пример и стал объяснять.

— Цель игры — посадить корабль на Луну.

Ситуация у нас с вами, Сергей Владимирович, такая. Ракета находится на расстоянии  $N$  метров от Луны, скорость ее  $V$  метров в секунду, запас топлива в баках —  $S$  тонн. Величины эти, конечно, известны командиру. Сопла ракеты повернуты к Луне и, добавляя топливо в двигатель, мы начинаем тормозить. Каждую секунду на световом табло в командирской рубке появляется информация о высоте корабля над поверхностью планеты, скорости его и запасе топлива. Так как командиром корабля будете вы, то знайте, что перед тем, как посмотреть на табло, нужно нажать такие клавиши: ХУ показывает скорость, ИПС дает остаток топлива в баке, ИП4 — время с начала торможения. Что касается высоты ракеты над планетой, то она будет загораться на вашем табло сразу при остановке калькулятора после проведения очередного маневра.

Задача понятна — посадить корабль так, чтобы в момент посадки скорость его была близка к нулю, топлива при этом желательно израсходовать поменьше, да и провести посадку побыстрее — незачем долго мучить членов экипажа, ведь пере-



грузки при посадке не маленькие. Да, что касается топлива, дополнительное условие — если его останется меньше 95 тонн, то обратный старт на Землю станет невозможным.

— А что от меня-то требуется, товарищ генеральный конструктор?

— Как что? Именно вы будете регулировать подачу топлива в двигатель и тем самым управлять посадкой.

— Ясно. Еще один вопрос перед стартом. Чем отличается эта «посадка» от аналогов, которые я раньше видел.

— Все программы, которые изучили мы с Мишкой — это одноклассник мой, показались нам слишком медленными, долго ждать ответа калькулятора. Пришли мы к папе. Он дал упрощенный алгоритм и несколько советов. Так что программа эта — коллективный труд папы, Мишки и мой, работает быстрее, чем те, что я видел. Ждать ответа приходится не более десяти секунд.

— Очень хорошо. Ну, что, поехали?

— Одну минуту. Сначала надо установить на приборах начальные показания: высоту, скорость и загрузить топливо в баки.

Пусть высота будет 1000 метров, скорость — 150 м/с, запас топлива — 500 тонн. Да, еще нужно ввести в бортовой компьютер ускорение свободного падения на Луне —  $1,62 \text{ м/с}^2$ . Вводим эти числа соответственно в регистры PA, PB, PC и P9.

Теперь можно начинать. Но прежде я покажу вам оценки качества посадки, которые выставит калькулятор после прилунения ракеты:

- 1,111111 — идеальная посадка;
- 2,111111 — нормальная посадка;
- 3,111111 — тяжелая посадка;
- 4,111111 — посадка с легко устранимыми повреждениями;
- 5,111111 — повреждения, полученные при посадке, неустранимы. Обратный старт к Земле вряд ли возможен;
- 6,111111 — очень сильные повреждения;
- 7,111111 — разгерметизация отдельных блоков в результате сильного удара. Жизнь членов экипажа под угрозой;
- 8,111111 — гибель экипажа;
- 9,111111 — взрыв корабля.

Андрей нажал В/О и С/П и на индикаторе появилась начальная высота. Сергей Владимирович, подумав секунду, ввел в двигатель первую порцию топлива — 20 С/П. Через 10 секунд на индикаторе появилась высота, на которой теперь находился корабль, 856,4.

— Еще столько же.

20 С/П, и высота стала 719,4.

— Еще одна порция, побольше. — приговаривал Сергей Владимирович. 25 С/П. Высота 591 метр.

Здесь командир решил проверить остальные параметры: скорость (клавиша XY) — 128,4 м/с, запас топлива (ИПС) — 435 тонн, время в пути (ИП4) — 3 с.

После ввода два раза подряд по 30 тонн горючего, высота корабля снизилась до 366,9 м.

— Можно и прибавить, топлива еще много, — решил Сергей Владимирович и набрал два раза по 35. Табло показало: высота — 194,1 м, скорость — 76,6, топливо — 305, время — 7. Далее полет протекал так:

Расход топлива	Высота
40	130,5
45	85,6
40	57,4
40	46,2
10	38,1
10	33,2
0	26,6
0	18,4
5	10,9
7	5,1
8	1,6
10	1,2
5	1,5
(здесь скорость оказалась отрицательной (—0,3) и корабль начал снова удаляться от поверхности)	
0	0,3
3	3,111111



Коснулись поверхности Луны. После посадки выяснилось (ИПС), что топлива осталось 82 тонны, а время, затраченное на этот процесс (ИП4), 21 секунда.

— Ну, что вам сказать, Сергей Владимирович. Сели вы тяжело, но корабль цел, с экипажем все в порядке. Вот только топливо вы перерасходовали, на обратный путь не хватит. Придется на Луне ждать корабль-спасатель и надеяться, что его капитан справится с посадкой лучше вас.

Но не огорчайтесь. Для первого раза вполне прилично. Сергей Владимирович сидел возбужденный, как будто и в самом деле он только что был за штурвалом.

— А знаете, — обратился он к друзьям, — действительно интересно.

— Еще раз попытаетесь, Сергей Владимирович?

— Охотно, но в другой раз. Время уже позднее, нам пора расходиться.



## Вечер четвертый

### И вновь рассказывает Андрей

У нас опять гости — бабушка с дедушкой. Калькулятор помогает маме вязать на машине. Сколько дней прожили вместе дедушка с бабушкой со дня свадьбы? Дедушка вспоминает. Калькулятор работает таймером. Дедушка ловит «лисы».

Бабушка с дедушкой приходят к нам несколько раз в месяц. В этот вечер первой их встретила у дверей Маша.

— А у нас калькулятор есть, — важно сказала она.

— Это что такое? — спросила бабушка.

— Тебе, наверное, неинтересно. Это по дедушкиной части, — ответил я. Дедушка до ухода на пенсию работал в конструкторском бюро. Папа говорит, что он принимал участие в разработке первых отечественных ЭВМ и даже имеет некоторый опыт программирования на них.

— Это уж точно, по дедушкиной, — согласилась бабушка. — А мне бы что-нибудь для шитья или для вязания.

— Так у нас же для вязания давно машина есть, — заметила мама, — ты и сама на ней вязала. Но, знаешь, я к ней калькулятор приспособила.

— Это еще зачем?

— А ты помнишь, что самое неприятное при вязке?

— Ну еще бы. Когда петли считать надо. Сколько снять, сколько набросить. Намучаешься с этим. То лишнюю снимешь, то нужную не провяжешь. Скучно, конечно, но ничего не поделаешь.

— Так вот я теперь именно ничего и «не подделываю». Кнопки на калькуляторе нажала и передо мной номер ряда и количество петель.

— Хочешь, покажу?

— Ну-ну, покажи, — заинтересовался дед, — и мне интересно будет.

Я принес калькулятор, ввел программу и отдал калькулятор маме. Программу я взял из журнала «Наука и жизнь», но мы с мамой разобрались в ней сами, без папы, и теперь мама постоянно берет калькулятор, когда садится за вязальную машину. Программа довольно большая, но зато удобная в работе.

Мама принесла образец выкройки и стала объяснять, как пользоваться программой.



## ДЛЯ МАШИНОЙ ВЯЗКИ

	0	1	2	3	4	5	6	7	8	9
0	0	С/П	ПП	61	ИП1	П9	П5	ИП0	П8	П4
1	ПП	69	0	С/П	ПП	61	ИП0	ИП4	—	Фх ≠ 0
2	04	ИП1	ИП5	—	Фх ≠ 0	04	÷	ПА	Ф1/Х	Фх < 0
3	32	/—/	П7	1	—	Фх < 0	41	1	П7	БП
4	45	ИПА	ИП7	х	ПА	ИП9	ИП7	+	П9	ИП1
5	—	Фх < 0	04	ИПА	ИП8	+	П8	ПП	69	БП
6	45	ИП3	÷	П1	ХУ	ИП2	÷	П0	В/0	ИП9
7	П6	КИП6	ИП6	4	Ф10 <sup>х</sup>	х	ИПВ	+	П6	КИП6
В.	ИП6	С/П	В/0							

## Инструкция

1. ширина петли П2, длина петли П3.

2. В/О С/П «О».

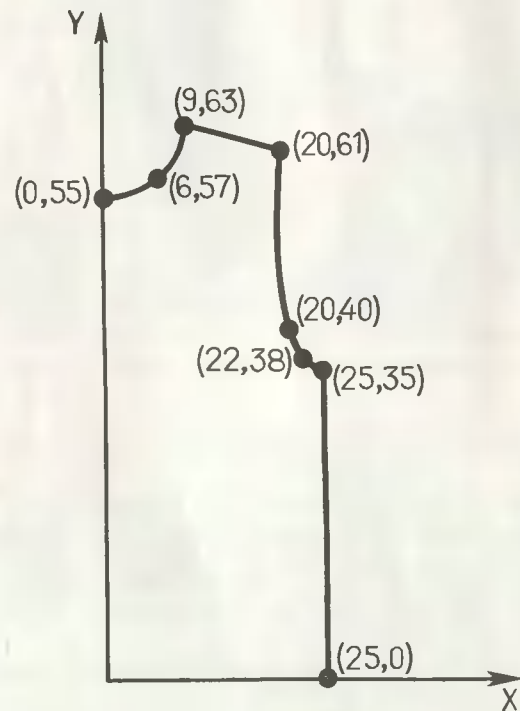
3. ХУ (координаты точки изгиба) С/П.

4. «а ... abbbb» (а ... а — номер ряда, bbbb (последние 4 цифры) — (число петель) С/П.

5. При появлении 0 на индикаторе перейти к п. 3, иначе к п. 4.

6. После окончания расчета одной стороны шаблона перейти к п. 3 и вводить координаты другой части шаблона.

Примечание. Вводить координаты узловых точек следует в порядке возрастания координаты Y.



— Сначала нужно на выкройке провести две линии — одну вдоль направления вязки, другую — поперек. Начинаться они должны в левом нижнем углу детали. Потом на чертеж наносят координаты точек излома, как на рисунке. Остается узнать размер петли. Это не сложно. Надо только провязать той же пряжей прямоугольный кусок, измерить его длину и ширину и разделить на число петель поперек и вдоль образца. Причем это можно сделать на том же калькуляторе.

Бабушка стала внимательно рассматривать чертеж выкройки, а мама тем временем продолжала.

Теперь все готово. Программа вводится в машинку, дедушка может тебе помочь в этом, а потом нужно нажимать кнопки и вводить в калькулятор координаты изгибов. В окошечке-индикаторе будет появляться число. Последние четыре его цифры — это количество петель, которые надо снять или добавить, а первые — номер ряда, на котором следует манипулировать с петлями.

Вот посмотри.

Мама ввела программу и загрузила в регистры размеры петли:  $13.5 \uparrow 40 \div П2$ ,  $9 \uparrow 40 \div П3$  (13.5 и 9 — это ширина и длина связанного эталона, а 40 — число петель. Оно получилось одинаковым и в длину и в ширину). После запуска программы на индикаторе появился ноль, и мама ввела координаты первой точки левой стороны.

— Учти, что вводить обязательно нужно от больших координат к меньшим, а то будут ошибки.

— Ох, сложно все это, — вздохнула бабушка.

— Это с первого раза. Когда привыкнешь, увидишь, что не так уж это сложно, зато очень удобно, — ободрила ее мама. — Но смотри дальше.

После ввода координат  $0 \uparrow 55$  С/П на индикаторе появилось 245000. Мама вновь нажала С/П. Загорелся ноль.

— Это значит, пора вводить следующие координаты, — пояснила она и набрала  $6 \uparrow 57$  С/П. После остановки калькулятора на индикаторе были цифры: 2460003.

— Ну и так далее, — продолжила мама. — Цифры эти означают, что с 245 ряда надо снять одну петлю, с 246 — 3 петли ... Так же рассчитывается правая сторона. После окончания всех расчетов передо мной готовая таблица. Я кладу ее рядом с машиной и горя не знаю. Ни о чем думать не надо, все перед глазами.

Ну как, тебе нравится?

— Действительно, удобно. Знаешь, Коля, — обратилась она к дедушке, — мне такая машинка тоже не повредила бы. А что она еще умеет делать?

— Меня учить умеет, — сказала сестренка.

— И чему же она тебя выучила? Покажи-ка?

— Андрюша, введи мою программу.

— Какую? У тебя их много.

— Про сложение и вычитание.

Я ввел программу, которую показал папа еще в первый вечер, и Маша стала демонстрировать, чему ее научил калькулятор.

— Гляди, и вправду молодчина! А не обманываешь ли ты меня? Может, ты просто запомнила, какие числа за какими идут, и сразу ответы набираешь?

— Что ты, бабушка, они же все время разные. Вчера одни,

сегодня другие. Как же их запомнишь? Это специальный генератор числа выдумывает. И никогда они не повторяются.

— Так уж и никогда? — улыбнулся дед. — Что, Игорь, никак генератор псевдослучайных чисел запрограммировал?

— Так точно, Николай Иванович. Элементарный, правда, но для наших целей вполне подходящий. Я его период не смотрел, но за несколько сотен ручаюсь.

— Да, для такой программы этого вполне достаточно, — согласился дедушка.

— Сколько раз слышала от тебя, — покачала головой бабушка, — об огромных вычислительных машинах. По твоим словам, для одной такой и комната наша может мала оказаться. А эту в шкаф положишь, потом и найдешь-то не сразу. В общем, скоро у нас с тобой юбилей — золотая свадьба. Вот, пусть, дети нам такую же подарят.

— Что ж, хороший будет подарок. Как вы, дети, на это смотрите?

— Положительно, — улыбнулся папа. — Кстати, хотите узнать, сколько дней вы прожили вместе?

— А чего тут узнавать? — ответила бабушка. — 49 лет да одиннадцать с половиной месяцев.

— А в днях сколько будет?

— Ну, умножить да сложить. . .

— Еще високосные годы надо учитывать, — прибавил дедушка.

— А в какой день недели вы зарегистрировались? Не помните? — снова спросил Игорь.

— Это ты слишком много хочешь, — сказала бабушка.

— Не так уж и много. У нас есть программа для микрокалькулятора, которая все это может. Ее, кстати, ваш внук написал. Хотите посмотреть?

— Это уж вы смотрите, а я пойду на кухню, Лене помогу.

Бабушка вышла из комнаты, а папа достал текст программы, дал его мне и придвинул калькулятор. Я ввел такую программу:

ВЕЧНЫЙ КАЛЕНДАРЬ

	0	1	2	3	4	5	6	7	8	9
0	пп	13	ип1	п2	ху	с/п	пп	13	ип1	ип2
1	—	ху	с/п	п0	фо	п4	ху	п3	ху	3
2	—	fx<0	2в	1	5	+	п4	кип0	кип4	ип4
3	ипа	х	п7	ип0	ипв	х	п8	кип7	кипв	ип7
4	ипв.	ип3	+	+	ипс	—	п1	7	÷	п8
5	кипв	ип1	ипв	7	х	—	в/0			

#### Инструкция

1. 30,6 ПА 365,25 ПВ 694066 ПС.

2. день ↑ месяц ↑ год (начальная дата) В/О С/П «код дня недели».

3. день ↑ месяц ↑ год (конечная дата) С/П «код дня недели» ХУ «количество дней между двумя датами».

Примечание. Код дня недели: 0 — воскресенье, 1 — понедельник, 2 — вторник и т. д.

Дедушка достал свой паспорт и прочел дату регистрации брака: 3 марта 1937 г. Я ввел эти данные в программу и быстро получил, что в этот день была среда. Потом я ввел дату сегодняшнюю: 15 февраля 1987 г., снова запустил программу и когда она закончила свою работу, вывел число дней между датами: 18246.

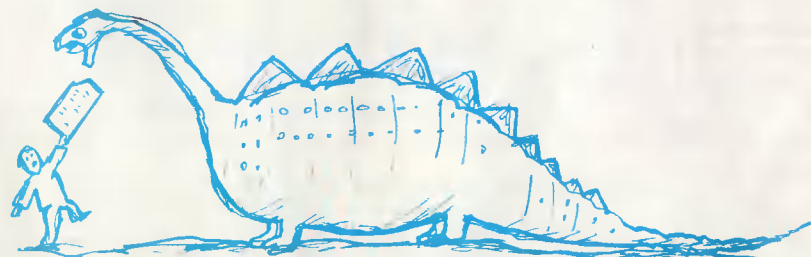
— Внушительный срок, — с уважением сказал папа.

Во время моих манипуляций с калькулятором дедушка внимательно следил за мной, потом он также внимательно рассматривал программу. Наконец, он сказал.

— Знаете, что я подумал? Если бы у меня была машина времени и я смог бы вернуться в пятидесятые годы и показал бы программисту тех лет, работавшему, скажем, на ЭВМ «Стрела» или БЭСМ-2, современные машины и программируемый микрокалькулятор, то он, наверное, выбрал бы калькулятор. Во всяком случае программирование на нем ему было бы намного понятнее.

— Почему? — удивленно спросил я, — ведь калькуляторы — самые современные ЭВМ, а ты говоришь о каких-то древностях.

— Да-да, — усмехнулся дедушка. — Современные микрокалькуляторы, использующие достижения самой современной технологии, по структуре и принципам программирования гораздо ближе к «первобытным чудовищам», чем к самым современным ЭВМ.



— Ну, Николай Иванович, придется вам об этом рассказать подробнее. У нас принято на наиболее интересные темы читать лекции в «домашнем университете». До сих пор единственным лектором был я, а теперь уступаю вам свое место за кафедрой.

— Ну, раз так, ничего не поделаешь, я согласен.



## От БЭСМ-2 к микрокалькулятору

Не буду подробно рассказывать об истории создания вычислительных машин. Кстати, слово «электронные» в привычном названии этих устройств я опустил неспроста. Первые вычислительные машины были не электронные, а электрические. В них роль переключающих элементов выполняли электрические реле. Принцип же их устройства напоминал именно калькулятор — области памяти, где хранились данные, то есть числа, и программа — совокупность команд, предписывающая последовательность их обработки, были разделены так же, как у калькулятора. Кроме того, первые ЭВМ опять же, как и современные программируемые калькуляторы, были приспособлены только для обработки числовой информации.

Конечно, размеры тех машин и калькуляторов несопоставимы. Например, первая автоматическая электрическая вычислительная машина «Марк-1», созданная в США в 1946 году, размещалась в помещении площадью более сотни квадратных метров, энергии потребляла, как средних размеров завод (ведь элементная база ее состояла из реле, требующих довольно большой мощности для переключения), а вычислительные параметры имела с сегодняшней точки зрения в высшей степени скромные: 60 регистров для записи констант, 72 накапливающих регистра. Это всего в несколько раз больше, чем возможности твоего калькулятора. Быстродействие же той «прамашинки» было даже в несколько раз меньше, чем у микрокалькулятора — 3 секунды длилась операция умножения.

Однако параллельно с развитием технической базы и переходом с электрических элементов на электронные, пересматривались и некоторые принципы архитектуры ЭВМ.

В частности, был пересмотрен и принцип разделения памяти. Здесь большую роль сыграли работы американского математика Дж. фон Неймана, предложившего рассматривать память вычислительной машины как единое целое. В каждой области памяти, ее называли ячейкой, можно было записать команду или число. Что есть что, определяла сама машина в соответствии с программой.

Программировать на первых машинах было довольно трудно. Я столкнулся с этим, работая на «Стреле». Понимали машины только свой собственный язык — язык команд. Каждая команда имела свой особый код — число, под которым она была известна машине. Программисту нужно было запоминать все эти коды. Кроме того, ему нужно было запоминать адреса ячеек, где находились команды. Ведь программа записывалась в последовательно расположенных ячейках, и при передаче управления очень важно было правильно указать адрес. Впрочем, современным калькуляторным программистам это должно быть понятно. Они тоже частенько сталкиваются с подобными проблемами.

Программистов в то время было очень мало, и работа их была окружена ореолом таинственности. Смотрели на них почти так же, как на физиков-ядерщиков или космонавтов.

Круг пользователей ЭВМ расширялся очень медленно. Во-первых, кому была охота запоминать многочисленные коды, писать абстрактные последовательности цифр? А ведь именно таковы и были программы, написанные в командах. Кому хотелось тратить время и силы на вылавливание и исправление многочисленных ошибок? Во-вторых, машин было не так уж много, ление многочисленных ошибок? Во-вторых, машин было не так уж много, стоили они очень дорого, да и машинное время было недешево. Поэтому к работе на ЭВМ старались допускать лишь «избранных». Машинное время давалось интервалами по 10—15 минут, расписание составлялось на неделю вперед. Поэтому основная работа программистов проходила не за пультом ЭВМ, а за рабочим столом. Попадая же за пульт, программист едва успевал ввести свою программу — громадную колоду перфокарт, судорожно выловить ошибки в программе и, если очень повезет, то и посчитать.

Стремление сократить длину программы и время ее счета приводило

к созданию запутанных, необыкновенно сложных программ, отлаживать которые становилось все труднее и труднее.

Короче говоря, необходимость облегчить труд программистов превращалась в первоочередную задачу. Первый шаг на этом пути был сделан, когда появились автокоды. Так стали называть языки, которые отличались от машинных тем, что вместо кодов команд можно было писать имена, а вместо абсолютных, числовых адресов команд разрешено было писать адреса символические. Писать программы на таких языках — их еще называли мнемокодами или автокодами — стало намного легче.

Судите сами. Вот как выглядела запись простой формулы

$$c = \frac{a + b}{a - b}$$

на языке команд:

05	10	0100	0110	0200
06	11	0100	0110	0210
07	13	0200	0210	0200

а вот как на автокоде:

+ A B C  
— A B D  
: C D C

Конечно, вторая запись намного понятнее первой, да и легче.

Но ничто не дается даром. Такое облегчение повлекло некоторые затраты. Ведь программа, где написаны символы, а не цифры, уже непонятна машине. Пришлось между человеком и машиной ставить посредника — программу-транслятор, которая переводила бы человеческий язык на язык машины.

— Николай Иванович, про трансляторы я им уже рассказывал, — вмешался папа.

— Очень хорошо, — продолжал дедушка, — скажу только, что первые трансляторы были довольно просты. Перевод шел, как говорили, «один к одному», то есть одна команда языка переводилась в одну команду машинную. Правда, и здесь уже приходилось заранее просматривать всю программу и ставить на место символических адресов абсолютные.

Но не только трудности программирования препятствовали массовому распространению вычислительной техники. Слишком мало было ЭВМ, слишком они были дороги и потому малодоступны. Размерами они отличались не просто большими, а, я бы сказал, громадными. Скажем, БЭСМ-2, выполнявшая несколько тысяч операций в секунду, очень немного по нынешним меркам, занимала зал в несколько сотен квадратных метров. Интересно, что сейчас микроЭВМ, уместяющаяся на письменном столе, имеет значительно большие возможности. Недаром говорят, что если бы авиация развивалась такими же темпами, как вычислительная техника, то сегодня полет из Москвы в Хабаровск занимал бы полчаса, билет на рейс стоил бы не дорожке пяти рублей.



Кстати, насчет энергетических расходов. Машина БЭСМ-2 состояла из нескольких тысяч электронных ламп. Ты, Андрей, уже проходил в школе, что это такое? А это были вакуумные приборы, похожие на обычные лампы накаливания. Снабжены они были катодами — устройствами, излучающими электроны, анодами — устройствами, эти электроны ловившими, и сетками, регулирующими движение электронов. Потребляли электронные лампы огромное количество энергии. Каждый катод нужно было нагреть, чтобы он излучал электроны, а саму лампу охлаждать, чтобы она не перегревалась. И на то и на другое требовалась энергия. Ситуация создавалась прямо-таки неразрешимая.

Чтобы сделать более мощную машину, приходилось использовать все больше и больше ламп. Лампы же — устройства не очень надежные. Состоят они из разнородных материалов: металлов, стекла, часто перегорают, могут и лопаться, как обычные электрические лампы. Поэтому ламповые цепи приходилось дублировать, чтобы выход из строя одной лампы не приводил к выходу из строя всей машины. Итак, чем больше рабочих ламп, тем больше ламп-дублеров. Чем больше энергии нужно на нагревание, тем больше энергии требовалось на охлаждение. Получалась настоящая энергетическая лавина. Поэтому не удивляйся, если тебе попадется на глаза какая-нибудь научно-фантастическая книга пятидесятих годов с описанием «думающих» машин. Авторы фантастических произведений чаще всего просто экстраполировали, то есть как бы количественно продлевали успехи сегодняшнего (для той поры) дня в будущее. Поэтому в рассказах и повестях часто присутствовала супермашина величиной с многоэтажный дом. Когда это чудовище начинало «думать», то мерк свет во всем городе или же государстве.

Изобретение транзисторов, а затем больших и сверхбольших интегральных схем положили конец этим не слишком оптимистичным прогнозам. И сейчас перед нами на столе не очень мощная, правда, но все же «думающая» машина.

Но, впрочем, я отвлекся. Так вот, микрокалькулятор по организации памяти находится на уровне самых первых вычислительных машин. По языковому же обеспечению он стоит чуть выше ЭВМ, понимавших только язык команд — с ними его роднит использование абсолютных адресов, и чуть ниже ЭВМ, оснащенных трансляторами с автокодов. Так же, как и они, калькулятор воспринимает символические обозначения команд. Облегчает работу и то, что названия эти написаны прямо на клавишах и для ввода команды достаточно нажать одну-две клавиши.

Вот так-то. Теперь понятно, почему мне так понравился калькулятор? Он напоминает мне о молодости, а это всегда приятно.

— Ну, ладно. Пока женщины заняты на кухне, расскажите мне о программе «Вечный календарь». Кто ее автор?

— Андрей, — сказал папа.

— Но ты же мне помогал! — возразил я. — Так что это наша совместная программа.

— Все равно, Андрей, ты молодец. А рассказать, как устроена программа, сможешь?

— Конечно, смогу.

— Ну давай по порядку. Начни с алгоритма.

— В книге Чаконя про микрокалькуляторы есть формула:

$$T = [365,25 \cdot \Gamma] + [30,6 \cdot M] + D - 694046,$$

где  $D$  — число месяца,  $M$  — номер месяца, причем в формулу надо подставлять  $M + 13$ , если  $M = 1$  или  $2$ , и  $M + 1$  во всех остальных случаях;  $\Gamma$  — год. Величину его нужно уменьшать на единицу, если  $M = 1$  или  $2$ . Квадратные скобки означают целую часть заключенного в них выражения.

Величина  $T$  показывает, сколько дней прошло от 28 февраля 1900 года до текущей даты.

Для определения дня недели используется еще одна формула:

$$D_n = \text{mod}(T, 7),$$

где  $\text{mod}$  — функция, называемая «деление по модулю». Результат ее — остаток от деления первого числа на второе.

$D_n$  — это код дня недели: 0 — воскресенье, 1 — понедельник и т. д.

— Подожди, подожди, — прервал меня дедушка. — Как ты будешь брать целую часть? Что-то я не вижу такой клавиши.

— А ее и нет, — ответил я. — И тем не менее брать целую часть можно! Но, может быть, я расскажу последовательно? — Давай.

— Сначала об общей структуре программы. Так как для определения промежутка между двумя датами нам придется дважды проводить вычисления по большой формуле, я вынес их в подпрограмму. Начинается она с адреса 13. Поэтому первая команда, которую выполняет калькулятор после ввода даты, — это переход на подпрограмму ПП 13. Результат работы подпрограммы — количество дней, рассчитанное по формуле, я записываю в адресуемый регистр  $P1$ , затем переписываю в  $P2$  и там храню до следующего расчета. Одновременно вычисляю код дня недели, который и выводится на индикатор. На этом первый этап вычислений прекращается. Далее можно ввести другую дату. На индикаторе вновь будет код дня недели. Чтобы получить число дней между датами, нужно нажать  $XU$  и эта величина появится на индикаторе.

Теперь я расскажу об основной вычислительной части программы. Оформлена она, как я уже говорил, в виде подпрограммы, начинающейся с адреса 13. Прежде всего она заносит дату в адресуемые регистры: день — в  $P3$ , месяц — в  $P4$  и год — в  $P0$ .

— А почему бы не заносить данные в регистры, расположенные по порядку? — спросил дедушка. — Это было бы куда легче запомнить.

— Э, нет. Здесь есть глубокий смысл. Я о нем скажу потом. Далее. Командами 19 — 21 я вычитаю из номера месяца тройку  $n$ , если остаток больше или равен нулю, то перехожу на команду по адресу 28, иначе прибавляю к номеру месяца число 15.

— Ничего не понимаю, — прервал меня дедушка, — у тебя в программе написано наоборот: «Если  $x < 0$ , то на 28». Во всяком случае я прочел бы эту команду именно так.

— Николай Иванович, Вы Андрея не путайте, — вмешался папа. — Дело в том, что создатели калькулятора посчитали более естественным такой порядок, когда условный переход происходит при невыполнении условия. То есть, когда написано  $Fx < 028$ , это означает, что при выполнении условия переходить на адрес 28 не надо, а нужно продолжить выполнение программы с команды, записанной после команды перехода.



— Очень оригинально. Особенно это поможет Андрею, когда он займется каким-нибудь языком высокого уровня, где все наоборот.

— Я с вами полностью согласен, но ничего не поделаешь.

— Ну, хорошо, — продолжил дедушка, — а почему ты прибавляешь 15, когда по формуле нужно прибавить лишь 13?

— А вот это наша с папой хитрость. Смотри. Сначала я из числа месяцев вычитаю тройку — это для проверки. Потом, если номер месяца меньше трех, то прибавляю 15, то есть  $15 - 3 = 12$ , записываю результат в P4 и командой 28 КИП4 добавляю еще единицу. Так я набираю нужное число 13. Перед этим я, кстати, уменьшаю номер года на единицу командой КИПО. Вот почему я засылаю год именно в нулевой регистр, а не, скажем, в пятый. Дело в том, что при косвенном обращении к регистрам с номерами от 0 до 3 содержимое их уменьшается на единицу, а при использовании регистров 4 — 6 их содержимое на единицу увеличивается. Вот я и достигаю, как говорит папа, минимумом команд максимума эффекта.

Теперь вернемся чуть-чуть назад. Если номер месяца больше или равен трем, то я пропускаю команды 23-27 и ограничиваюсь добавкой единицы к номеру месяца (команда 28).

— Молодец! — похвалил дедушка. Помню я, как программисты, составляя программу, старались сократить ее хоть на одну команду и были очень горды, когда им это удавалось. Правда, отлаживать такую программу практически было невозможно, — грустно прибавил он.

— Да, Николай Иванович, вы правы. У калькуляторных программистов желание оптимизировать программу часто превращает ее в совершенно невообразимое произведение. Вообще, к сожалению, работа с калькулятором — как заноза в структурном программировании. А без знания основных принципов структурирования программ сегодня никакой более или менее сложный программный комплекс не сделаешь.

— А что такое структурное программирование? — спросил я, — ты мне об этом ничего не говорил.

— Я расскажу об этом позже, а ты продолжи рассказ о программе.

— После проведения необходимых манипуляций с номером месяца и года я перехожу непосредственно к вычислениям. Команды 29 — 32 вычисляют первое произведение в формуле, а команды 33 — 35 — второе. Числа 30.6 и 365.25 предварительно записаны в регистры PA и PB. Результаты умножения записываются в регистры P7 и P8.

Теперь я могу ответить на твой вопрос, дедушка, о выделении целой части. Я извлекаю ее с помощью команд косвенной адресации: КИП7 и КИП8. Содержимое этих регистров при обращении к ним команды косвенной адресации не моди-

фицируется, за исключением тех случаев, когда там находилось нецелое число. В этих случаях в регистрах остается как раз целая часть прежнего их содержимого.

— Ну и ну! Прямо-таки всеильна эта косвенная адресация. В инструкции о ней, наверное, много написано.

— Что Вы. Там ничего об этом не сказано. В инструкции вообще написано много меньше того, что может калькулятор.

Я испугался, что папа с дедушкой увлекутся и я не смогу закончить рассказ о своей программе. Поэтому и решил вмешаться.

— Может быть я продолжу?

— Конечно, конечно.

— Впрочем, осталось совсем немного.

После выделения целых частей двух слагаемых (команды 37 — 38) я заканчиваю расчеты величины T и запоминаю ее в регистре 1 (команды 39 — 46). Теперь начинается расчет кода дня недели. Так как функции mod в языке калькулятора нет, то приходится вычислять ее по формуле:  $\text{mod}(T, 7) = T - 7 \cdot [T/7]$ . Это записано опять-таки с использованием команды косвенной адресации для выделения целой части командами 47 — 56. Последней командой подпрограммы является команда возврата В/О. Вот и все.

Ну как?

— Хорошо, — похвалил дед. — Теперь, хоть у меня и нет опыта работы с калькулятором, разобраться в программе я смогу. Трудно мне, честно говоря, судить, насколько твоя программа хороша, но если она выполняет свои функции — значит она уже хорошая. Тем более, что ты и ухищрения разные использовал для сокращения ее текста.

Однако одно замечание я хочу сделать. Речь идет об алгоритме. Если уж заниматься оптимизацией программы, то начать следует именно с алгоритма. Твоя формула не годится, например, для прошлого века. Она не учитывает, что по григорианскому календарю 1900 год не был високосным, да и 2100 год не будет. Это, конечно, нестрашно. Календарь, действующий в течение столетия, — не так уж плохо. Но раз она используется лишь для нашего столетия, не стоит ли ее несколько упростить? Например, ограничиться вводом двух цифр года вместо четырех. Программа при этом практически не изменится, время ее работы тоже, а вот для пользователя она станет удобнее. Вместо четырех цифр года, ему придется набирать лишь две.

— Как это? — не понял я сначала.

— А ты подумай.

— А, — почти сразу догадался я, — нужно самому умножить 365.25 на 1900 и это число включить в формулу.

— При этом нужно будет сразу изменить и шестизначное число в конце формулы, — добавил отец.

Я быстро перемножил пару чисел на калькуляторе, провел вычитание и написал новую формулу:

$$T = [365.25 \cdot G_1] + [30.6 \cdot M] + D - 81,$$

где  $G_1$  — две последние цифры года.

Действительно, программирование этой формулы ничего не изменит в программе, а доля ручного труда при вводе данных уменьшится. Если же константы записывать в саму программу, то модернизированная формула даже даст выигрыш в четыре ячейки: двузначное число вместо шестизначного.

— Преподал тебе дедушка наглядный урок оптимизации, — сказал отец. — Как видишь, чтобы оптимизировать программу, начать следует не с нее самой, а с алгоритма. Таким образом, как правило, проводятся самые радикальные оптимизации.

— Ну, что, умники, — вошла бабушка, — все выяснили? А у нас пирог готов.

Мы сели за стол ужинать.

— Да, Игорь, — проговорил дедушка. — Многие твой калькулятор может, но так, как наши жены пироги печь, — не умеет. Здесь одних расчетов мало. Это искусство.

— Николай Иванович, шутки шутками, но помочь калькулятор может, наверное, в любом домашнем деле.

— Скажи, пожалуйста, Лена, как ты следишь за созданием своего «шедевра»?

— Станный вопрос. По часам. Замечаю время, потом через час проверяю, готов ли пирог, и если готов, вынимаю его из духовки.

— А что будет, если ты пропустишь время?

— Пирог сгорит — только и всего.

— Значит, таймер тебе не помешал бы?

— Конечно.

— А давай заменим его калькулятором.

— Как это?

— Смотри. Принцип работы любых часов — это счет. В механических часах подсчитывается число колебаний маятника, которое затем преобразуется в движение стрелки; в электронных — число колебаний кварцевого или какого-нибудь другого генератора, которое преобразуется потом в цифровую индикацию. Наш калькулятор тоже умеет считать. Дело только за генератором. В качестве его мы сможем использовать время выполнения какой-либо команды. Для каждого экземпляра калькулятора — это величина почти постоянная. Конечно, нам не удастся сделать хронометр, работающий с точностью до десятых долей секунды. Для этой цели нужны специальные схемы, обладающие повышенной стабильностью и нечувствительные к изменению приложенного напряжения, температуры, влажности. Но нам такая точность и не нужна. Для измерений

же, скажем, с точностью до секунды калькулятор вполне подойдет.

— Я, кажется, понял, — вступил я в разговор и потянулся за калькулятором.

— Даже писать предварительно не будешь? — весело спросил папа.

— Не буду.

Я перевел калькулятор в режим программирования и набрал:

00.FLO 01.00 02.C/P

— А дальше что? — недоверчиво спросила мама.

— Дальше нужно узнать, сколько времени длится цикл. Для этого я измерю время работы десяти, нет лучше ста циклов.

Я перевел калькулятор в режим вычислений и записал сотню в R0:100 P0. Потом взял свои часы с секундной стрелкой, засек время и быстро нажал В/О C/P. Через 41 секунду калькулятор остановился.

— Следовательно, один цикл выполняется за время 0,41 секунды.

— Это — «разрешающая способность» твоего таймера, — пояснил папа. — Минимальный временной интервал, который с его помощью можно отмерить. Не так уж плохо. А тебя устраивает, Лена?

— Ничего не понимаю, — сказала мама. — Как мне с помощью этой программы отмерить десять минут или полчаса?

— Да это же просто, — сказал я. — Нужно время в секундах делишь на 0,41, полученное число записываешь в нулевой регистр и запускаешь программу.

— По-моему, — посоветовал отец, — лучше несколько усложнить программу и избавить маму от лишних вычислений.

— Можно, — согласился я. — Тебе как удобнее задавать время, в минутах или секундах?

— Лучше в минутах.

Я немного подумал и переписал программу так:

00.P1 01. 6 02.0 03. X 04.IPA 05. X 06.P0 07.FLO 08.07 09.IP1

10.C/P 11.5P 12.00

— Ну и накрутил, — с уважением сказал дед. — Теперь объясни, что получилось.

— Введенное число в минутах записываем в P1. Затем умножаем его на 60 — это число записано прямо в программной памяти — и переводим таким образом в секунды. Потом делим его на «временную постоянную» нашего калькулятора. Но чтобы не делать лишней операции по перемене местами содержимого RХ и RУ, я записываю умножение на обратную величину, то есть в регистр. А перед началом работы требуется ввести результат деления числа 100 на время, в течение которого выполняется сотня циклов. Дальше все, как было раньше:



запись полученного числа циклов в Р0 и выполнение этих циклов. Перед остановкой я для удобства вывожу на индикатор величину требующегося времени в минутах.

— Это уже лучше, — сказала мама. — Мне нужно будет лишь ввести время в минутах?

— Да. А для проведения нового замера времени достаточно будет ввести лишь новую величину и нажать клавишу С/П. Возврат к началу программы происходит автоматически. Для этого в тексте записаны команды: БП 00.

— Работать с такой программой, правда, легче, но точность ее будет ниже, — сказал отец.

— Это почему же?

— Неужели ты сам не видишь, что выполнение первых шести команд и девятой тоже потребует дополнительное время? А оно в программе не учитывается.

— А как же его учесть?

— Наверное, можно так. Прикинуть, сколько времени работают эти команды, и вычесть его из задаваемого числа.

— Ну и насколько точнее будет при этом работать Андрюшкин таймер? — поинтересовалась мама.

— Секунд на пять — десять.

— Это мне не важно. За десять секунд пирог не сгорит. Так что программа Андрея меня вполне устраивает. Вот только погоди... Как я узнаю, что время истекло?

— Неужели ты не поняла? — удивился я. — Мелькание цифр на индикаторе кончится и там появится число введенных минут.

— Здорово, — сказала мама, правда, без особого энтузиазма. — Только зачем мне такой таймер? Я и на часы могу смотреть. Вот если бы калькулятор сигнал давал, звонок или гудок, тогда он мне помог бы.

Я загрустил. Жаль, конечно, программы, но мама была права. Эх, был бы у калькулятора звонок, как у будильника!

— Расстроился? — участливо сказал дедушка. — А я, кажется, знаю, как поправить дело. Где у вас транзистор?

Я, еще ничего не понимая, принес дедушке наш «Океан». Он включил приемник, установил на указателе диапазонов длинные волны, поставил рядом калькулятор и начал священных действовать. Сначала он запустил мою программу, поставив время ожидания одну минуту, потом стал крутить ручку настройки о радиоприемника. Приемник замолчал. Слышалось лишь тихое потрескивание. И вдруг раздался резкий звук.

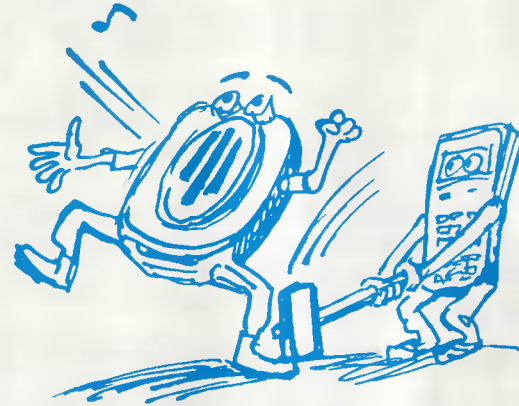
— Ничего сигнал? — торжествующе спросил дедушка.

Я ничего не понял и удивленно смотрел на него.

— Тебе не на меня, а на индикатор смотреть надо было. Тогда бы ты заметил, что звук раздался как раз в тот момент, когда калькулятор кончил считать и на индикаторе появилась единица.

— Выходит, приемник отреагировал на прекращение работы калькулятора? — спросил я.

— Конечно. Ведь калькулятор во время работы излучает сигнал определенной частоты. При остановке сигнал меняется, эту перемену и ловит приемник, откликаясь на нее звуком. Понял?



— Значит, приемник можно использовать как датчик звукового сигнала, связанный с калькулятором? Отлично!

— Ну вот, Лена, общими усилиями мы тебе из подручных средств создали настоящий таймер.

— Такой таймер меня вполне устраивает. Теперь каждую субботу приглашаю вас на пироги.

— А ты, Машенька, что загрустила? Пироги не нравятся? — спросила бабушка.

— Нравятся, но мне скучно.

— Бедная девочка, совсем тебя забыли взрослые со своим калькулятором.

— А он и мой тоже. Просто мне его не дают.

— Да как же не дают? — возмутился я. — А кто сложением и вычитанием занимался?

— Это несправедливо, — сказал дедушка. — Нужно отдать немедленно калькулятор Маше и пусть она нам тоже что-нибудь покажет.

— Хотите, я вам наш «кубик-рубик» покажу?

— Хотим, — ответила за всех бабушка.

— Только, пускай, Андрей мне введет программу. Вводить программы Маша самостоятельно еще не научилась и помогать ей в этом приходилось мне.

Я ввел программу «Числа по кругу» и положил перед Мацей листок с кодами.

— Ты запиши мне сначала числа, а потом я переставлять их буду.

Я заполнил стек числами: 1, 2, 3 и 4 и отдал калькулятор сестренке.

— Вот смотри, бабушка, если нажать на клавишу 6, то поменяются местами первые две цифры, а если на 7, то все цифры переедут по кругу — первая на место последней, вторая станет первой, третья второй, а четвертая попадет на место третьей. Поняла?

— Не совсем. А что я должна делать?

— Нужно переставить цифры так, чтобы сначала была четверка, потом тройка, двойка и единица.

— Ну, давай попробую.

Бабушка взяла калькулятор и стала опасливо нажимать на клавиши. Цифры замелькали.

— Смотри-ка, правда, интересно.

Минут через десять бабушка достигла цели и очень обрадовалась.

— А я уже умею все переставить за четыре хода, — гордо сказала Маша. — Вот, смотри.

Она пододвинула к себе калькулятор и бойко стала нажимать на клавиши. Через пару минут все числа стояли на нужных местах.

— Очень ловко. Но давай все же пойдём посмотрим твоих кукол. Ты с калькуляторами про них не забыла? А ты, Лена, обещала мне показать свитер, который связала с помощью калькулятора.

Когда женщины вышли, папа обратился к дедушке. — А вы, Николай Иванович, сыграть не хотите? Для вас у меня есть игра очень интересная. На «лис» не желаете поохотиться?

— Как это?

— Вы слышали, наверное, про соревнования по ориентированию?

— Слышал.

— Так вот, я спрячу в «лесу» пять «лис», а вы, по этому «лесу» двигаясь, попытаетесь запеленговать и найти «лис». Для этого нам понадобится листок бумаги в клеточку размером  $9 \times 9$  квадратов. Вы будете выбирать какой-нибудь квадрат и проверять, попала ли на линию пеленга какая-нибудь «лиса». На индикаторе при этом будет показано, успешен ли пеленг или нет и сколько «лис» сидит на линиях пеленга. Ваш «приемник» одновременно будет проверять все вертикали, горизонталы и диагонали, проходящие через точку, где вы находитесь, и будет сообщать, сколько «лис» в данный момент на всех направлениях находится.

Когда вы попадете в один квадрат с «лисой», то на индикаторе появится сигнал успеха: 1.111111 — 01.

## ОХОТА НА ЛИС

	0	1	2	3	4	5	6	7	8	9
0	ПД	5	ПО	1	0	П7	ПП	71	КП1	ПП
1	71	ИП7	÷	КИП†	+	КП†	FL0	06	Сх	С/П
		ИП7	÷	ПА	КИПА	ХУ	ИПА	—	ПВ	Сх
3	П6	5	ПО	ИПС	КИП†	—	FX=0	42	9	F1/X
4	БП	19	FBx	П8	КИП8	ХУ	ИП8	—	ИПВ	—
5	ИП7	x	FX≠0	65	ИПА	ИП8	—	FX≠0	65	÷
6	FX <sup>2</sup>	1	—	FX=0	66	КИП6	FL0	33	ИП6	БП
7	19	ИПД	1	1	x	FP	+	ПД	КИПД	ХУ
8	ИПД	—	ПД	9	x	1	+	ПС	КИПС	ИПС
9	В/О									

## Инструкция

1. 0, **ЧЧММ** (ЧЧ — часы, ММ — минуты) В/О С/П «0».

2. ху (двузначное число, х и у — координаты охотника в лесу) С/П «число запеленгованных лис» или «1,111111 — 01», если охотник поймал лису.

3. Перейти к п. 2.

Я этой программы раньше не видел и с интересом стал наблюдать за дедушкиной охотой.

Перед ее началом папа ввел число 0.1849. Я уже знал, что это нужно для запуска генератора псевдослучайных чисел. Число было дробным.

После нуля целых записывались четыре цифры — по две на часы и минуты. После запуска программы прошло минуты две. Наконец, на индикаторе появился ноль.

Игра протекала так.

Дедушка ввел 34, то есть координаты квадрата в третьем ряду и четвертом столбце. На индикаторе появилось 2.

— Повезло, — сказал он. — Пойдем дальше.

Он набрал 64 С/П и на индикаторе засветились единицы: 1.111111 — 01.

— Николай Иванович, — сказал папа, — если и дальше так пойдет, то ни в каком лесу от вас не спрячешься.

Дедушка ничего не ответил и продолжал сосредоточенно нажимать клавиши: 37 С/П — на индикаторе 2, 67 С/П — 3.

Я решил, что игра вот-вот закончится, но неожиданно она затянулась — дедушке перестало везти.

Игра складывалась таким образом:

99 С/П «1», 11 С/П «1», 91 С/П «4», 97 С/П «2», 61 С/П «3», 19 С/П «2», 94 С/П «1.111111 — 01»

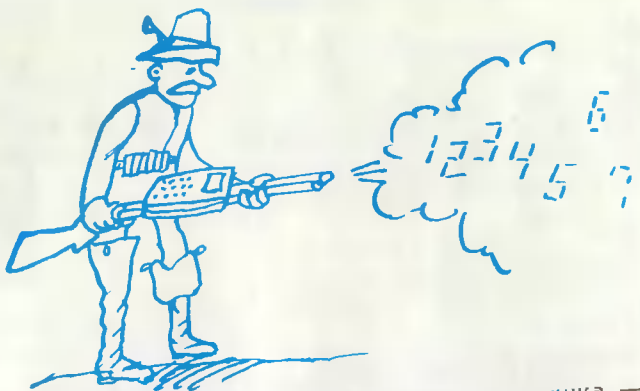
Наконец-то! «Лисы» были окружены, и дедушка стал их вылавливать одну за другой:

73 С/П «3», 83 С/П «2», 82 С/П «1.111111 — 01».

— Попалась, голубушка, — приговаривал дедушка.

51 С/П «1», 21 С/П «1.111111 — 01», 71 С/П «3», 76 С/П «1.111111 — 01»





— Кажется, всех поймал, — вдруг объявил дедушка. — Интересная игра. Да и программа, наверное, тоже интересная. Расскажи-ка, Игорь, как она работает?

— Первым делом, — начал папа, — калькулятор прячет в «лесу» пять «лис». Делает он это с помощью генератора псевдослучайных чисел.

— Это, когда мы часы и минуты вводим? — спросил я.

— Да. Сам генератор — это подпрограмма, начинающаяся с команды 71. Основная программа обращается к этой подпрограмме в цикле (команды 06 — 17) пять раз. Таким образом в адресуемые регистры заносятся координаты «лис». В какие регистры, я говорить не буду, иначе трудно будет удержаться от соблазна перед началом игры подсмотреть их содержимое. А тогда играть неинтересно будет. После того, как «лисы» спрятаны, программа очищает индикатор и предлагает начать игру.

Введенное охотником двузначное число — предполагаемые координаты «лисы» — разделяется на два числа: отдельно X-координата и Y-координата, которые записываются в регистры PA и PB. Этим занимаются команды 20 — 28. Причем, для отделения целой части от дробной используются команды косвенной адресации.

Затем начинаются проверки. Сначала на совпадение введенных и «задуманных» координат (команды 33 — 37). Если координаты совпали, то есть, «лиса» обнаружена, то команды 38 и 39 выдают на индикатор сигнал успеха: 1,111111 — 0 и управление передается на команду по адресу 19. Охотнику предлагается ввести новый пеленг.

Если «лиса» не найдена, начинается проверка по направлениям. Сначала по горизонтали и вертикали, потом по диагонали. Этим занимаются команды 42 — 67. Они подсчитывают, сколько «лис» оказалось на линиях пеленга, и накапливают результат в регистре P6.

После этого, число запеленгованных «лис» выводится на индикатор и охотник может делать следующий ход.

Тут в комнату вошла бабушка.

— Не засиделись ли мы с тобой, дед?

— А штуку эту, — обратилась она к папе, — вы нам обязательно подарите.

— Как обещал, с завтрашнего дня начну заходить в магазин. Ведь теперь программируемый калькулятор не сразу купишь.



## Вечер пятый

Рассказывает Игорь Петрович

Полгода с микрокалькулятором. Калькулятор предсказывает погоду. Что такое структурное программирование? С микрокалькулятором — на рынок. Как количественно оценить качество? Калькулятор — кассир. Нужно ли оптимизировать программы? Чего не может калькулятор? Покупаем домашний компьютер.

Прошло полгода со дня приобретения микрокалькулятора. Первые радости, удивления, недоумения, восторги миновали. Понемногу все мы привыкли к калькулятору и он стал восприниматься как рядовой предмет домашнего быта, вроде пылесоса или магнитофона.

Дочке он уже стал надоедать. Считать с его помощью она, правда, научилась и довольно сносно. Во всяком случае таблицу умножения освоила.

Изредка она берет калькулятор, чтобы поиграть. Но так как самой вводить программы ей пока еще трудно, а старшие не всегда рядом, то и играет она с ним нечасто. Реже, чем с куклами.

Сын же использует калькулятор частенько и по делу. После того как он убедился, что с помощью калькулятора решать квадратные уравнения значительно легче, чем без него, да и просто вычисления по формуле можно провести быстрее, чем на бумаге, он постоянно кладет калькулятор рядом с собой, когда делает уроки. Любит и поиграть с ним. Причем особенно интересуют его не процесс игры, а сами игровые программы. Он создал их уже десятка полтора. А полигон для обкатки — весь класс. Калькуляторы есть уже у многих, и ребята организовали что-то вроде клуба. Обмениваются программами, алгоритмами, приемами программирования. Устраивают настоящие «калькуляторные» матчи.

Жена тоже привыкла к «игрушке». Свитера, связанные с помощью калькулятора, появились уже не только на Андрее, но и на Маше. Сейчас моя очередь. Каждый вечер Лена придирчиво подсчитывает калорийность съеденной пищи и планирует рацион на завтра. Как я понял, тучность теперь никому из нас не грозит.

Неожиданно для меня и родители Лены, теща с тещей, тоже полюбили электронного помощника. Особенно после того, как





ПРОГНОЗ ПОГОДЫ

0	1	2	3	4	5	6	7	В	9
0 Сх	4	6	ПД	5	ПО	1	П6	ИП6	ВП
1 5	С/П	КП6	FL0	ОВ	ИП4	2	—	Fx=0	24
2 1	ПВ	БП	43	ИП3	Fx≠0	28	1	ИП5	+
3 ПВ	ИП4	Fx=0	64	ИП5	+	ПА	—	Fx=0	55
4 ИП2	Fx≠0	53	1	ИП2	ИП6	2	—	F10x	х
5 ИПВ	+	С/П	3	КБПД	ИП6	3	—	Fx≥0	62
6 4	КБПД	5	КБПД	ИП6	КБПД	6	КБПД	72	ИП2
7 Fx=0	76	2	ИП2	—	—	—	—	—	—

Инструкция

1. В/О С/П «10000» ответ на 1-й вопрос С/П «20000» ответ на 2-й вопрос С/П... «50000» ответ на 5-й вопрос С/П (допустимые ответы даны на с. 93) «АОВ» (смысл величин А и В расшифровывается на с. 93).
2. Продолжение работы: перейти к п. 1.

мы подарили им такой же, как наш. Маша каждый раз, когда они приходят, спрашивает, как поживает у них братик нашей машинки. Мария Петровна в свою очередь очень любит задавать Андрюше вопросы типа: «Какой день недели был 17 февраля 1931 года?» или «Сколько дней осталось до 31 декабря 1999 года?». Андрею эта игра тоже нравится. Еще больше ему нравится удивлять бабушку своими успехами. Недавно он сразил Марию Петровну, познаколив ее с калькуляторным прогнозом погоды. Дело было так. Андрей нашел программу прогноза в журнале «Наука и жизнь», разобрался в ней и решил удивить стариков. И вот как-то после чая Мария Петровна вздохнула: «Ох, что-то кости ломит, видать к дождю», он тут же заявил: «Сейчас проверим твой прогноз по калькулятору».

Бабушка, как всегда, очень удивилась. — Что же тут непонятного, — серьезно возразил Андрей, — есть много факторов, которые определяют погоду. Конечно, все учесть на калькуляторе невозможно. Я читал, что прогноз погоды рассчитывают на больших вычислительных машинах. Но известно, что можно определить погоду по тому, как солнце заходит, откуда ветер дует, как ведет себя барометр. Программа, которую я нашел, как раз учитывает поведение барометра и погоду в данный момент.

Я был очень доволен, что сын нашел эту программу. Все расчеты в ней, по существу, нет. Ее можно назвать логическим. Все действия, которые происходят в ней, — это сравнения, выбор того или иного направления работы. Кроме того, необычен решен вопрос о вводе информации. Калькулятор выводит не сказки: номер вопросов, на которые нужно давать ответы. Что не спутать вопросы с ответами, первые выводятся в виде пя-

значных чисел. Например, 10000 — 1-й вопрос, 20000 — 2-й и т.д. Вопросы появляются после запуска программы клавишами В/О, С/П. Отвечать на них нужно следующим образом:

1. Состояние погоды сегодня: 0 — хорошая, 1 — плохая;
2. Давно ли стоит такая погода: 0 — недавно, 1 — давно;
3. Показания барометра: 0 — падают, 1 — растут, 2 — не меняются;
4. Меняются эти показания: 0 — быстро, 1 — медленно, 2 — вообще не меняются;
5. Время года: 0 — лето, 1 — зима, 2 — весна, 3 — осень.

После ввода последнего числа на индикаторе появляется «прогноз». Он закодирован в виде АОВ, где А и В — цифры.

Первая цифра показывает характер погоды, вторая — ее продолжительность. Расшифровываются эти символы так.

А: 1 — плохая погода, 2 — хорошая, 3 — ливень, гроза, 4 — сильный ветер, 5 — оттепель, снегопад, 6 — северный ветер, мороз.

В: 0 — продержится такая погода недолго (меньше суток), 1 — несколько дней, 2 — будет стоять долго.

Андрей быстро посмотрел на барометр и ввел данные в программу: погода хорошая (0), стоит давно (1), показания барометра падают (0) и довольно быстро (0), на дворе весна (2).

Немного «подумав», калькулятор выдал: 401.

— Это значит, — сказал Андрей, — будет сильный ветер в течение нескольких дней.

— Ты смотри-ка, — удивилась Мария Петровна, — похоже! А насчет дождя твой калькулятор ничего не говорит?

— Нет, — ответил Андрей. — Но если сильный ветер будет, то, наверное, и дождь тоже. Значит, и твой прогноз, бабушка, верен.

Я промолчал. Недавно калькулятор выдал нам «хорошую погоду в течение нескольких дней», а на следующий день пошел снег с дождем.

Конечно, дело не в этом. После разбора этой программы Андрей стал лучше разбираться в косвенной адресации, подпрограммах и познакомился с хорошим приемом организации диалога.

Вообще, пожалуй, калькулятор полезнее всего сыну. И дело не только в том, что он научился решать несложные задачи, главное, у него стал появляться интерес к программированию. Он все чаще и чаще задает мне вопросы, непосредственно к калькулятору отношения не имеющие.

— Папа, а что это такое — структурное программирование? — спросил он недавно. Запомнил, как мы говорили об этом с Николаем Ивановичем, когда они с бабушкой были у нас в гостях. Хотя принципы структурного программирования вряд ли применимы к микрокалькулятору, я все-таки решил, что представ-

О структурном программировании

На смену языкам команд высокого уровня, но психология программиста по-прежнему оставалась старой. По-прежнему большим достижением считалось создание сверхоптимизированных программ. Программисты рассматривали свою деятельность исключительно как искусство, не поддающееся строгому анализу.

Тут-то теоретики программирования стали задумываться над вопросом «Что делать дальше?». Ясно было, что, работая по-старому на новых машинах далеко не уедешь. Необходимо было пересматривать основные концепции программирования. Тогда-то и появился термин «структурное программирование». Приверженцы нового метода, среди них один из наиболее известных специалистов голландский профессор Э. В. Дейкстра, критику старых методов программирования сосредоточили сначала на использовании оператора GO TO (по-русски «идти на»). Так почти во всех языках программирования называли оператор перехода. На языке нашего калькулятора ему соответствует команда «БП». Так вот, они считали, что именно он превращает программы в «тарелки макаронов». Если же его исключить и оставить только три конструкции — следование, повторение и выбор, то появится возможность писать и отлаживать программы, состоящие из отдельных, независимых блоков. Это все равно, как строить дом не из кирпичей, а из отдельных функциональных блоков, что, конечно же, будет быстрее и дешевле.

Правда, при этом программы станут несколько длиннее, будут чуть дольше. Однако для современных машин с огромным объемом и большими объемами памяти, это, наверное, не так страшно. Впоследствии оказалось, однако, что сам по себе оператор GO TO не уже вреден. Путаница происходит от его неразумного использования. То так же нельзя сказать, что вреден, например, огонь, поскольку он способен уничтожить леса и дома. Но играть с ним вредно. Надеюсь, это сравнение понятно? Когда ты станешь изучать алгоритмические языки, то, по всей вер-

ри работе с калькулятором полезно следовать, если есть некоторый резерв памяти. В противном же случае можно ими и пренебречь. В конце концов микрокалькуляторные программы — действительно «микропрограммы» и чаще всего не так уж трудно разобраться. А почувствовать преимущества структурного программирования ты полностью сможешь, когда начнешь программировать больших ЭВМ.



Закончив лекцию, я в очередной раз задумался о месте программируемых калькуляторов в иерархии ЭВМ. С одной стороны, учить программированию с их помощью довольно трудно, особенно в свете идей структурного программирования — уж больно неудобен их язык. С другой — чем говорить абстрактно об алгоритмах, циклах и выборах, лучше показать примеры их реализации даже на неудобном языке микрокалькулятора. Наконец, почему надо обязательно равняться на большие ЭВМ? Почему просто не рассматривать калькулятор как одно из счетных устройств в ряду арифмометров, логарифмических линеек и тому подобных приспособлений, только более универсальное, точное и быстрое.

Для меня, кстати, калькулятор превратился в верного помощника в работе. После того как я обзавелся справочниками, содержащими программы для калькулятора (жаль только, что их очень мало), и сам написал десятка полтора программ для решения часто встречающихся у меня счетных проблем, калькулятор стал мне просто необходим.

Сегодня вечером мы решили всей семьей собраться вместе и поделиться успехами. Свои сюрпризы я, как обычно, решил обыграть и потому обратился к жене с вопросом: «Ты, вроде, на рынок завтра собралась?».

— Да. Рекомендуешь взять с собой калькулятор? На рынке он не понадобится.

— Напрасно ты так думаешь. Попробую тебя переубедить. Когда ходишь по рынку, то самая главная проблема — проблема выбора. Хочется сделать покупку получше и подешевле, а кругом всего столько, что глаза разбегаются. Так вот в этой проблеме калькулятор может помочь. Точнее, не сам калькулятор — он может сделать только то, что мы в него зложим, а программа, которую я нашел в журнале.

— Интересно, что это за программа?

— Действительно, программа интересная. Она может пригодиться не только на рынке. Кстати, Андрей, думаю, и тебя она заинтересует.

#### С МИКРОКАЛЬКУЛЯТОРОМ НА РЫНОК

0	1	2	3	4	5	6	7	8	9
4	п0	с/п	кп1	ф0	ф0	03	+	+	+
1	4	п0	ф0	пп	90	ип2	+	ип3	пп
87	ип3	+	ип2	пп	В7	ип1	+	ип3	ип2
пп	В8	ипд	ип7	ип9	÷	ипс	х	ипв	+
ипА	ип7	х	ип6	÷	п3	ипв	п2	ипс	ипА
ип9	х	—	ипА	÷	ип6	÷	ипА	ип7	х
—	ипв	÷	п4	ф0	х	ип5	х	ху	п5
с/п	+	ип4	ип2	+	Бп	70	ип1	+	ип3
х	+	+	ип4	÷/0					—
4	÷	кп0							

#### Инструкция

1. В/О С/П «4» Ц (1,1) ↑ Ц (5,1) ↑ Ц (1,5) ↑ Ц (5,5) С/П.
2. признак 1 ↑ признак 2 С/П «цена».
3. Продолжение работы: к п. 2.

— Допустим, ты собралась покупать яблоки. Примем, что их качество оценивается лишь по двум параметрам — по размеру и сочности. Разложим цену яблок на составляющие, зависящие от этих признаков, учтем и постоянную добавку, не зависящую от качества: сюда входят, например, расходы на транспортировку и хранение. Предположим для простоты, что зависимость цены от каждой из этих составляющих линейная и что коэффициент пропорциональности тоже линейно зависит от другой составляющей. Так для цены Ц получается несложная формула:

$$Ц(x,y) = A + (B + by)x + (C + cx)y = A + Bx + Cy + Dxy.$$

— А что это за  $x$  и  $y$ ?

— Уровни качественных признаков. Глядя на конкретные яблоки, ты сама станешь определять эти уровни по пятибалльной системе.

— А откуда я возьму все эти  $A$  и  $B$ ,  $C$  и  $D$ ?

— Для этого я предлагаю довольно хитрую процедуру. Ты отыскиваешь на рынке яблоки, которым можно поставить предельные баллы, узнаешь их цены и составляешь такую таблицу:

Размер X	Стоимость Y	Цена Ц
1	1	0,50
5	1	1,20
1	5	0,70
5	5	2,00

В/О С/П Ц(1,1) ↑ Ц(5,1) ↑ Ц(1,5) ↑ Ц(5,5) С/П

Калькулятор работает несколько секунд и находит коэффициенты  $A$ ,  $B$ ,  $C$  и  $D$ .

— Хорошо. Но что я с ними буду делать?

— Вот тут-то и начинается самое интересное. Ты внимательно рассматриваешь яблоки у всех продавцов. Самые низкокачественные тебя, естественно, не привлекут. Самые высококачественные, скорее всего, покажутся слишком дорогими. Но вот у двух продавцов ты видишь яблоки, которые вполне устраивают тебя и по цене, и по качеству. Оба просят по 1 рублю 30 копеек за килограмм. Допустим, яблоки первого ты оцениваешь по размеру и сочности в 3 и 4 балла соответственно. Вводи эти числа в калькулятор и запускай его:  $x \uparrow y$  С/П. Расчет

дает 1,23 рубля. Это меньше, чем назначенные за них продавцом 1,30 рубля. То есть продавец завышает цену своего товара. Теперь введи в калькулятор баллы, которыми ты оцениваешь яблоки второго продавца — пусть эти оценки равны соответственно 4 и 3. Результат расчета 1,35 рубля. Покупай эти яблоки — ведь ты заплатишь за них меньше, чем они стоят. И вообще предпочитай такой товар, за который запрашивают цену, максимально отклоняющуюся вниз от твоей оценки. Такая покупка принесет тебе наибольшее удовлетворение как самая экономная.

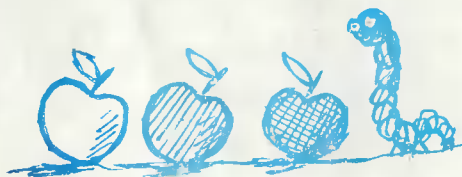
— Ну, ладно, — сказала жена. — Теперь я спокойна. Возьму с собой калькулятор, введу в него на рынке программу, и самая экономная покупка мне обеспечена. А тебе не приходило в голову, что иногда хочется выбрать просто самые красивые яблоки?

— Извини за каламбур, но ты попала прямо в яблочко. Как же ты собираешься выбирать самые красивые?

— Посмотрю. Какие больше понравятся, те и куплю.

— Давай проведем эксперимент. Андрей, принеси, пожалуйста, из кухни вазу с яблоками.

Когда ваза появилась на столе, я отобрал из нее десяток яблок, и выложил их рядом.



— Выбери, пожалуйста, самое красивое, — обратился я к Лене.

— Вот это.

— А может быть это? — хитро сказал я, и положил рядом не менее аппетитное яблоко.

— Может и это, — неохотно согласилась жена.

— Судя по твоей нерешительной интонации, не так-то просто сделать выбор, когда признаки нельзя измерить. А что, если предложу вам разложить все эти яблоки по степени красоты? — обратился я к домочадцам. — Справитесь ли вы с такой работой?

— Ну, не интригуй. Говори сразу, что за программу приговаривал. То, что она интересная, я не сомневаюсь, у тебя все программы интересные.

— Я написал программу, которая занимается довольно сложным делом: она «шкалирует» объекты, то есть располагает их по степени возрастания какого-либо признака. Сложность подобного процесса в том, что признак, по которому надо «шкали-

ровать», не поддается измерению. Как, например, измерить красоту или вкус того же яблока. Как оценить, какая книга интереснее или какой фильм лучше?

— И как же справится с такой работой калькулятор? Что ему нужно будет давать яблоки на пробу или показывать кинофильмы?

— Нет, конечно, это сегодня никакой машине будет не по зубам, да и в ближайшее время, пожалуй, тоже. Однако калькулятор может помочь в анализе наших ощущений, наших оценок. Ведь одно дело — расположить десяток яблок по соответствующему критерию, а другое — выбрать из двух яблок то, которое больше понравится.

— А если выбрать трудно.

— Алгоритм анализа и это предусматривает. В таком случае можно считать каждый экземпляр из пары равноценным.

Метод, положенный в основу программы, называется методом попарных сравнений. Все объекты сравниваются попарно друг с другом, результаты этих сравнений анализируются, и калькулятор на основе этого анализа выносит свой «вердикт», а проще говоря, приписывает каждому объекту определенный вес, своего рода количественное выражение качества.

Программа построена так, что сначала она задает вопросы, то есть показывает номера объектов, которые нужно сравнивать. В ответ нужно набрать номер предмета, который кажется предпочтительнее, или 0, если предметы кажутся одинаковыми.

Каждый вопрос представляет собой цепочку цифр, где номер одного объекта отделен от номера другого тремя нулями, например 50002 означает сравнение предметов 5 и 2. Появление сообщения 10001 является сигналом об окончании сравнения. Теперь надо нажать клавиши БП 42 и С/П и подождать появления нуля на индикаторе. На этом работа программы заканчивается и остается извлечь результаты, накопленные в адресуемых регистрах: вес первого объекта — ИП1, второго — ИП2 и т. д.

Понятно? Теперь давайте перенумеруем объекты нашего исследования. Вот тебе, Маша, острая палочка. Цифры ты знаешь, наноси их на яблоки, а Андрей пока введет программу.

#### ШКАЛИРОВАНИЕ ОБЪЕКТОВ

	0	1	2	3	4	5	6	7	8	9
0	ПД	ПО	1	КП†	FL0	03	ИПД	ПС	ПО	КИПО
1	ИПО	ВП	4	ИПС	+	С/П	Fx=0	25	FL0	10
2	ИПС	1	—	БП	07	ИПС	—	Fx=0	35	КИПС
3	1	+	КПС	БП	18	КИП†	1	+	КП†	БП
4	18	ИПД	ИПО	0	КИП†	+	FL0	44	ПС	ИПД
5	ПО	КИП†	ИПС	÷	КП†	FL0	51	Cx	С/П	



## Инструкция

1. число анализируемых предметов В/О С/П.
  2. «А000Е» ответ (А или В) С/П.
  3. Повторять п. 2 до появления сообщения «1, 11». После этого перейти к п. 4.
  4. БП 41 С/П «0» ИП1 «условный вес предмета 1» и т. д.
- Примечание. Число предметов не должно превышать 11.

Свою работу дети закончили почти одновременно. Маша, сопя от напряжения, вывела на последнем яблоке цифру 10, а Андрей перевел калькулятор в режим вычислений.

Я набрал на клавиатуре число объектов 10, нажал клавиши В/О С/П и после того, как на индикаторе появилось сообщение 900010, отдал калькулятор жене.

Попарное сравнение десятка яблок заняло около десяти минут. Я не сообразил сразу, что число сочетаний из десяти по два равно 45. Надо было бы ограничиться пятью, но отступать было некуда. Наконец работа закончилась, и все заметно повеселели. Яблоки выстроились по ранжиру.

— Действительно, интересно, — сказала жена. — Но где это может пригодиться на практике, ума не приложу.

— А скажи, как по твоему можно сравнить, скажем, розы «Бель Блонде» по степени насыщенности цвета лепестков?

— Наверное, никак. Но опять-таки, зачем?

— А вот зачем. Было известно количество удобрений, подсыпанных в горшочки, где росли розы. После ранжирования роз по насыщенности цвета удалось построить эмпирическую зависимость этой насыщенности, выраженной в условных весах, от количества граммов удобрений. Вот и практический пример использования метода. Ясно?

— Ясно, — ответил сын. — Только, что такое «эмпирический»?

— Эмпирический, значит, взятый не из теоретических соображений, а из опыта, из наблюдений. Кстати, если уж об этом зашла речь, просматривая материалы журнала «Наука и жизнь», я наткнулся на любопытную эмпирическую зависимость между размером спелого арбуза и его весом, выведенную одним школьником.

Я быстро пролистал стопку вырезок, лежащих на письменном столе, нашел нужную и показал сыну:

$$m \approx 17 \cdot 10^{-6} L^3.$$

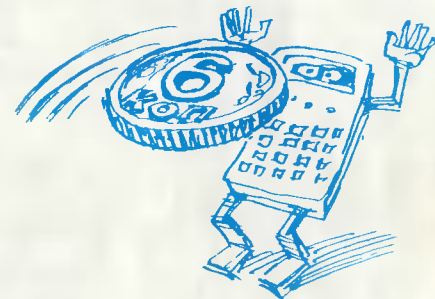
здесь  $m$  — масса в килограммах, а  $L$  — длина обхвата в сантиметрах.

Если построить эту зависимость, то можно получить «кривую спелости». Этот рисунок дан в журнале. Остается взвесить арбуз, измерить длину его «экватора» и научным способом определить, стоит ли его покупать.

— Ну, арбузы пока еще не поспели, — заметила жена, — но ты, Андрей, когда они появятся, будешь ходить за арбузами только с калькулятором.

— А мне мама покупки делать еще не доверяет, — жалобно сказала Маша.

— Я думаю, скоро мама позволит тебе самостоятельно ходить в магазин. Считать ты уже научилась. Впрочем, для лучшей тренировки у меня есть новая программа. Она будет «задумывать» какое-нибудь число — сумму от одного до двух рублей, а ты должна будешь отсчитать столько же монет. Пока не наберешь нужную сумму, на индикаторе будет появляться — 1. Если ты отсчитаешь денег больше, чем придуманная калькулятором сумма, он покажет тебе на индикаторе 1. В этом случае надо брать сдачу, то есть вводить числа со знаком «—». Наконец, когда сумма, предложенная калькулятором, будет угадана, она появится на индикаторе, а за угаданным числом загорятся пять единиц, как пять восклицательных знаков, что в этой программе означает «Спасибо!». Для нового сеанса отгадывания достаточно нажать С/П.



Учи, если попытаешься набрать число, которое не соответствует какой-нибудь монетке, скажем 6 или 30, то калькулятор его не пропустит.

О твоей промашке он сообщит словом ERROR на индикаторе. Так что учись использовать для расчета только определенные монетки.

Вот текст программы. Сейчас Андрей введет его и ты сможешь поупражняться.

Я взял калькулятор и ввел программу «Оплата счета».

## ОПЛАТА СЧЕТА

0	1	2	3	4	5	6	7	В	9
0 ПД	5	0	П1	2	0	П2	1	0	П3
1 5	П4	3	П5	2	П6	1	П7	ИПД	П1
2 1	х	Зп	+	ПД	КПД	ХУ	ИПД	—	ПД
3 2	F10 <sup>x</sup>	х	ИП1	+	КПС	Сх	ИПД	ПВ	С/П
4 ПА	Fх<0	44	/-/	↑	↑	↑	7	ПО	ЕО
5 КПИ↑	—	Fх*0	60	FLO	49	ВП	7	БП	10
6 ИПА	ИПВ	+	ПВ	ИПС	—	Fх*0	77	БП	10
7 1	/-/	БП	39	1	БП	39	ИПС	Fх<0	74
В +	С/П	БП	1В					9	F1/X

## Инструкция

1. **0,4ЧММ** (ЧЧ — часы, ММ — минуты) В/О С/П «0».
2. **номинал монеты** С/П «ответ».
3. Если ответ «—1», сумма еще не набрана. Перейти к п. 2. Если ответ «1», перебор. Перейти к п. 2 и вводить новый номинал со знаком «—», то есть получать сдачу. Если ответ «число, 11111», сумма оплачена полностью.
4. Для новой игры: С/П «0» и перейти к п. 2.

**Примечание.** Если введенное число не соответствует принятым номиналам монет, то появляется сообщение: «ЕГГОГ». Введенное число не засчитывается. Перейти к п. 2.

После ввода программы Андрей перевел калькулятор в режим вычислений, посмотрел на часы, набрал на клавиатуре 0,1845, нажал В/О и С/П и отдал калькулятор Маше.

Пока Маша пыталась «оплатить чеки», Андрей достал испанные листочки бумаги и обратился ко мне. «Помнишь, папа, когда приходили твои друзья, ты показывал им формулу из одной статьи».

— Помню, — сказал я, посмотрев на его записи. — Это зависимость коэффициента устойчивости транзистора в схеме с общей базой от разных параметров.

— Так вот, я написал программу, которая позволяет вести расчеты по этой формуле. Причем я решил максимально использовать свойства стека. Программа получилась довольно короткая. Хочу ее тебе показать.

Сын разложил листочки по порядку и начал рассказывать. — Твоя формула имела такой вид:

$$C = \frac{\sqrt{1 + 2\beta \cos \varphi + \beta^2} \sqrt{\beta^2 + 2\beta \cos \varphi}}{x(1 + 2\beta \cos \varphi + \beta^2) - \beta \sin \varphi}.$$

Прежде всего я решил, что вводить все три параметра удобно сразу и непосредственно в стек, а не в адресуемые регистры. Пользователю при этом нужно будет нажимать меньше кнопок, а что касается программной обработки, все числа будут прямо под рукой. Далее, как рекомендуют в некоторых книжках, я выделил одинаковые куски из формулы, чтобы не считать их дважды, и, наконец, записывая программу, я рядом с командами писал содержимое регистров стека, как ты учил. Все это позволило мне довести длину программы до 37 команд, хотя первый вариант был почти вдвое длиннее. По-моему, короче написать программу просто невозможно. Мне хочется услышать твое мнение.

И сын показал текст программы.

## РАСЧЕТ ПО ФОРМУЛЕ

	0	1	2	3	4	5	6	7	8	9
0	Fcos	FBx	Fsin	F0	XY	P2	x	P4	XY	P3
1	x	2	x	ИП2	Fx <sup>2</sup>	P5	+	F√	ИП4	2
2	x	1	+	ИП5	+	P5	F√	x	XY	ИП2
3	x	ИП3	ИП5	x	XY	—	÷	С/П		

## Инструкция

1.  $x \uparrow \beta \uparrow \varphi$  В/О С/П «С».
2. Для новых расчетов перейти к п. 1.

**Примечание.** Формула, расчет по которой запрограммирован, приведена на с. 102.

— Ну, что ж, хорошая работа, — сказал я, проглядев программу. — Однако позволь спросить тебя, сколько времени заняло у тебя программирование?

— Дня два, — подумав, ответил Андрей.

— Теперь представь себе, что по этой формуле нужно было получить результаты всего один раз, то есть просчитать одну кривую? Так не лучше ли (для дела, разумеется) написать программу, пусть вдвое более длинную, зато потратив на это всего час и еще час, чтобы получить все результаты?

— Но ты же сам говорил, что нужно стремиться делать более короткие программы?

— Да, если их делать быстро, а не тратить на это несколько дней.

— Сын так старался, — вступила за него жена.

— Обижаться не надо. Надо учиться. К тому же я говорю пока лишь об одной ситуации. Возможна и другая, когда к программе будут обращаться многократно, причем разные люди. Такие программы не грех как следует отработать. В этом случае экономия даже на одной лишней команде, на одном лишнем нажатии клавиши помножается на число использований программы, и в этом случае несколько дней работы обернется в будущем большой экономией времени.

С этой точки зрения результат твой хорош. Правда, я не стал бы утверждать, что достигнутая тобой длина программы — абсолютный минимум. Недаром говорят, нет такой программы, которую нельзя было бы сократить хотя бы на одну команду. Думаю, и твой рекорд — не исключение. Но есть у меня и более серьезные замечания.

Я понимаю, что ты делал эту программу как пример. Но если бы делал ее для использования инженерами, то надо было бы предварительно узнать, удобно ли им ею пользоваться?

Кстати, помнишь, мы обсуждали как-то с моими друзьями проблему постановки задачи? Сергей Владимирович говорил



тогда, что часто специалист не может толком объяснить, что ему нужно. Иногда бывает и наоборот. Математик, делая работу для специалиста, не представляет себе, зачем нужна создаваемая им программа и удобно ли будет потребителю ею пользоваться. Так вот ты выступил в роли такого математика. К примеру, почему ты решил, что каждый раз нужно вводить все три параметра? Представь себе, что специалисту потребовалось построить графики зависимости этой функции от  $\beta$ , рассматривая  $\varphi$  и  $x$  как параметры. В этом случае следовало бы изменить как ввод, так и структуру программы. Например, таким образом. Вводить сначала  $\varphi$  и  $x$ , вычислять выражения, зависящие только от них, запоминать их в адресуемых регистрах, а уж затем вводить значение  $\beta$ . Начало программы, работающей в таком режиме, могло бы выглядеть так:

00.П0 01. XY 02. Fcos 03.FBx 04. Fsin 05. П1  
06. F $\Phi$  07.2 08. X 09. П2 10. X 11. П3 12. C/П

Таким образом все члены формулы, не зависящие от  $\beta$ , вычислены и записаны в адресуемые регистры: X в П0,  $\sin\varphi$  в П1,  $2\cos\varphi$  в П2 и  $2X\cos\varphi$  в П3. Теперь для каждого следующего значения  $\beta$  эти величины вычислять не нужно, достаточно их вызывать из соответствующих регистров. Таким образом сократится время работы программы, а это для пользователя важнее всего.

Можно было бы решить эту проблему и по-другому. Важно каждый раз с самого начала четко представлять себе, зачем и для кого пишется программа, и исходя из этого стараться обеспечить максимальные удобства пользователю. Это непреложный закон для любого программиста, и чем раньше ты поймешь его, тем лучше будет и для тебя, и для твоей дальнейшей деятельности. В противном случае могут произойти довольно забавные казусы. В своей книге «Методология программирования» польский математик В. Турский описывает один из них. Он вспоминает, как еще на заре программирования один его приятель попросил проделать несколько вычислений. Он запрограммировал формулы, просчитал все на машине и отдал приятелю несколько страниц скучных цифр. Каково же было удивление В. Турского, когда он через несколько дней, случайно забежав в комнату приятеля, обнаружил, что тот сидит над толстым томом тригонометрических таблиц и трудолюбиво вычисляет синусы и косинусы для полученных чисел. «Как оказалось, он не знал, что вычислительная машина умеет вычислять синусы и косинусы, ну а я не знал, что это ему нужно», — заканчивает свой рассказ математик.

— Ну ты, кажется, прочел незапланированную лекцию, — недовольно сказала жена, — так можно отбить у Андрея охоту к калькулятору.

— А мне кажется, нет. Просто я поговорил с Андреем как профессионал с профессионалом. Его квалификация уже позволяет ему делать серьезные, рабочие программы. Вот я и высказал, как говорят, деловые претензии. Так всегда обсуждают серьезные работы. А, Андрей? Не обижаешься на критику?

Сын приободрился:

— Чего ж тут обижаться? Я же не знал, как пользуются этой формулой.

— Ну вот, видишь? — успокоил я жену. И снова обернулся к сыну:

— Не отбил я у тебя охоту к программированию?

— Нет, конечно! Мне ведь и так интересно программировать, даже если по моим программам никто никаких задач решать не станет.

Интересно.

Не просто полезно. В конце концов в быту пылесос с полотером даже полезнее.

Интересно. Интересно всем.

Маше — для нее калькулятор — это и доброжелательный учитель, и увлекательная непредсказуемая игра.

Андрею — для него это собственная ЭВМ. Когда он придумывает программы и затем реализует их на калькуляторе, он кажется себе одновременно ученым, исследователем, мореплавателем и космонавтом. Кроме того, как это ни парадоксально, но недостаточно подробная, да и чего греха таить, не всегда правильная инструкция к калькулятору, заставляет пользователя самостоятельно взяться за его изучение.

Интересно Лене. Кроме пользы калькулятора при вязании или выпечке пирога, появляется момент приобщения к самой современной технике, о которой раньше она знала только понаслышке.

Интересно родителям Лены.

Марии Петровне калькулятор — неожиданный помощник в самых разнообразных ситуациях.

Николаю Ивановичу — воспоминание о молодости.

Признаюсь, что работать с калькулятором интересно и мне. Приходится искать неочевидные решения, задумываться о стро-гой постановке задачи, оптимизации алгоритмов.

И все-таки.

Многого калькулятор не может. Прежде всего нет у него возможности работать с нечисловой информацией, нет дисплея, позволяющего получать результаты в наглядной форме и в прямом смысле слова разговаривать с машиной. Нет внешнего устройства, дающего возможность не только записывать программы, но и вести каталог их, держать там «записную книжку», справочники, в которых легко отыскать нужную информацию. Нет и печатающего устройства, которое позволило бы

не только фиксировать цифры, но и создавать таблицы и рисунки, печатать тексты. . .

— Ну и что ты предлагаешь? — услышал я вдруг голос жены. Я и не заметил, что стал думать вслух.

— Есть уже персональные вычислительные машины, возможности которых несравнимо выше, чем у калькуляторов. С их помощью можно учить не только арифметику, но и другие предметы, иностранные языки, например. Можно играть с ними в шашки и в шахматы, а можно и в другие электронные игры, очень интересные к тому же. Можно моделировать большинство игровых автоматов. Можно вести каталоги и дневники, можно создавать и редактировать тексты. В общем, можно очень много. Неплохо бы такую ЭВМ, которая называется домашний персональный компьютер, иметь и нам. Вот купим велосипед для Андрея. . .

— Я могу еще и на старом покататься, — отозвался сын.

— Так решено? Покупаем компьютер?



## Рекомендуемая литература

1. Дьяконов В. П. Справочник по расчетам на микрокалькуляторах. — М.: Наука, 1986. — 224 с.
2. Кибернетика. Микрокалькуляторы в играх и задачах. — М.: Наука, 1986. — 160 с.
3. Пойа Д. Математическое открытие. — М.: Наука, 1976. — 448 с.
4. Пухначев Ю. В., Данилов И. Д. Микрокалькуляторы для всех. — М.: Знание, 1986. — 190 с.
5. Романовский Т. Б. Микрокомпьютер в школе. — Рига: Звайгзне, 1986. — 239 с.
6. Трохименко Я. К., Любич Ф. Д. Инженерные расчеты на программируемых микрокалькуляторах. — Киев: Техника, 1985. — 328 с.
7. Турский В. Методология программирования. — М.: Мир, 1981. — 265 с.
8. Цветков А. Н., Епанечников В. А. Прикладные программы для микроЭВМ «Электроника БЗ-34», «Электроника МК-56», «Электроника МК-54». — М.: Финансы и статистика, 1984. — 175 с.
9. Чакань А. Что умеет карманная ЭВМ. — М.: Радио и связь, 1982. — 194 с.
10. Данилов И. Д. Секреты программируемого микрокалькулятора. — М.: Наука, 1986. — 160 с.
11. Славин Г. В. Программирование на программируемых микрокалькуляторах типа «Электроника БЗ-34». — Таллин: Валгус, 1985. — 126 с.