

С.И.ШАПИРО  
А.Б.БОЙКО

---

*Программируемые*  
МИКРО-  
КАЛЬКУЛЯТОРЫ  
В ОБУЧЕНИИ

---

С.И.ШАПИРО  
А.Б.БОЙКО

---

*Программируемые*  
МИКРО-  
КАЛЬКУЛЯТОРЫ  
В ОБУЧЕНИИ



Москва «Радио и связь» 1989

ББК 32.97  
Ш23

УДК 621.322:621.382.049.77

Р е д е н з е н т ы: д - р т е х н . н а у к В . П . Д Ь Я К О Н О В ,  
канд. техн. наук Ю. В. ПУХНАЧЕВ

Редакция литературы по вычислительной технике

Шапиро С. И., Бойко А. Б.

Ш23 Программируемые микрокалькуляторы в обучении.—  
М.: Радио и связь, 1989. — 256 с.: ил.

ISBN 5-256-00319-4.

Эта книга об обучении программированию и о программировании обучения с использованием программируемых микрокалькуляторов (ПМК), о диалоге человека и ПМК в процессе совместного решения учебных, логических и игровых задач с опорой на сильные стороны каждого компонента системы. Для связи со школьным курсом информатики программы сопровождаются алгоритмами на учебно-алгоритмическом языке.

Для широкого круга читателей — школьников старших классов, учащихся ПТУ и техникумов и всех, кто заинтересован в повышении компьютерной грамотности.

ш 2404060000-066 152-89  
046(01)-89

ББК 32.97

Научно-популярное издание

Шапиро Самуил Иосифович

Бойко Алексей Борисович

## ПРОГРАММИРУЕМЫЕ МИКРОКАЛЬКУЛЯТОРЫ В ОБУЧЕНИИ

Заведующая редакцией Г. И. Козырева

Научный редактор А. Б. Бойко

Редактор Т. М. Толмачева

Художественный редактор Н. С. Шеин

Переплет художника А. К. Акименкова

Технический редактор З. Н. Ратникова

Корректор А. С. Дзуцева

ИБ № 1435

Сдано в набор 11.08.88

Подписано в печать 24.01.89

Т-05018 Формат 60×90<sup>1/16</sup> Бум. тип. № 2 Гарнитура литературная

Печать высокая Усл. печ. л. 16,0 Усл. кр.-отт. 16,0 Уч.-изд. л. 17,59

Тираж 40 000 экз. Изд. № 21666 Зак. № 150 Цена 1 р.

Издательство «Радио и связь». 101000 Москва, Почтамт, а/я 693

Московская типография № 5 ВГО «Союзучетиздат». 101000 Москва, ул. Кирова, д. 40

ISBN 5-256-00319-4

© Издательство «Радио и связь», 1989

## ПРЕДИСЛОВИЕ

Сегодня программируемые микрокалькуляторы (ПМК), еще не став общим достоянием, уходят в тень персональных компьютеров. Это, по-видимому, и несправедливо, и непрактично. Еще немало лет ПМК будут наиболее доступным видом вычислительной техники, и их потенциальные возможности должны быть реализованы до конца.

Книга С. И. Шапиро и А. Б. Бойко направлена на преодоление равнодушного отношения к ПМК, убедительно показывая, как много может сделать пытливый ученик с помощью ПМК и внимательного заинтересованного учителя. В условиях ограниченных ресурсов ПМК программирование в машинных кодах обладает хотя и запретной с точки зрения высокой методики, но от этого не менее притягательной силой. Здесь ощущение своей способности и изобретательности в управлении машиной, лежащей на вашей ладони, проявляется непосредственно и ярко. Кроме того, благодаря широкому использованию учебного алгоритмического языка книга позволяет сочетать абстрактную логику алгоритмизации с приключениями конкретного ПМК.

Может быть, некоторые методисты найдут поводы для придирок в отношении порядка и формы подачи материала, захотят поспорить по ассортименту предложенных задач. Надо, однако, учитывать, что это не казенный учебник, а книга для интересующихся программированием. К тому же энтузиазм и изобретательность авторов неподдельны, и нет сомнения в том, что не одну тысячу молодых и не очень молодых людей книга привлечет к информатике.

Академик А. П. Ериков

## ОТ АВТОРОВ

70-е годы отмечены рождением нового вычислительного средства — электронного программируемого микрокалькулятора (ПМК). Потребность в ПМК для научных, инженерных и других расчетов была столь значительной, что первая же модель «Электроника Б3-21» нашла широкий круг пользователей. ПМК быстро совершенствовались, в последние годы наблюдается сближение характеристик наиболее мощных ПМК и персональных ЭВМ — появилось такое вычислительное средство, как карманный компьютер.

Все более важной задачей становится подготовка кадров. Может случиться так, что подготовка кадров не будет успевать за компьютеризацией производства, а значит, нельзя будет добиться эффективного использования компьютеров. Одна из причин — это недостаточное количество школьных ЭВМ, поскольку вне апробирования на реальной ЭВМ в процессе обучения использование условного языка программирования нередко превращается в манипулирование символами. Об этом, например, предупреждает пчальный опыт изучения темы «Алгоритмы и программирование» в курсе алгебры 8 класса. Очевидно, что проигрывание алгоритмов на ПМК, доступных школе, может оказаться неоценимую помочь процессу обучения.

Одно из возражений противников использования ПМК в обучении программированию — у ПМК нет развитой логики, средств анализа ситуации. ПМК — не что иное, как «арифмометр», только электронный, «электронная логарифмическая линейка», просто «счеты новой конструкции».

Цель данной книги — продемонстрировать широкие возможности отечественных ПМК «Электроника» (МК-56, МК-54, Б3-34, МК-61, МК-52), прежде всего логические возможности на примере решения игровых задач. Нередко приходится слышать, что алгоритмы поиска ответа существуют только для математических задач, а решение логических и тем более игровых задач относится только к сфере догадки. Такая установка отсекает пути к анализу на ЭВМ, возможно, наиболее интересных задач. Мы хотели показать, что ПМК может успешно справляться с задачами и этого класса. Что же касается математических задач, то наиболее интересна их роль в формировании знаний, так как постановка познавательного численного эксперимента, отражающего динамику

возникновения понятий, — одна из первоочередных задач совершенствования процесса обучения.

Работа с ПМК должна сформировать у учащихся определенный стиль мышления: экономичность суждений, способность принимать верные решения в нестандартных ситуациях, а также терпение, сосредоточенность, умение разрабатывать вопросы до полной ясности, необходимой для сопоставления эффективных программ в условиях ограниченных вычислительных и языковых ресурсов ПМК. Учащийся познает цену эффективного алгоритма, приучается думать о возможностях оптимизации программы. Так закладывается алгоритмический стиль мышления.

Книга демонстрирует эффективность взаимовыгодного диалога человек — ПМК. В таком диалоге на долю человека приходится постановка задачи, формулирование гипотезы, зрительная оценка ситуации, на долю ПМК — численные расчеты вариантов. В ряде задач часть «человеческих» функций передается ПМК, что позволяет изучать границы его интеллектуальных возможностей. Это не только захватывающая игра, но и психологический эксперимент, моделирующий человеческое мышление.

Если идти от механизма обработки информации, то окажется, что человек и ПМК противостоят друг другу: бинарная логика микрокалькулятора чужда гибкой и многоплановой психике человека. Тем важнее показать человеко-машинный диалог, основанный на сильных сторонах каждого компонента такой системы. Решение на ПМК достаточно сложных логических и игровых задач иллюстрирует потенциальные возможности этого класса микропроцессорной техники, помогает учащимся преодолеть психологический барьер, укрепляет в них веру в творческие возможности человека.

Работая над книгой, мы не стремились приводить только «застегнутую на все пуговицы» оптимальную программу. Это лишь усложнило бы задачу начинающего. Первый вариант программы — часто лишь грубое, но достаточно близкое приближение к «человеческим» формам и способам решения задачи. Вслед за этим демонстрируется «мелкая доводка» в пределах избранной стратегии — процесс, когда программа шлифуется, сокращается. Кардинального улучшения программы, однако, удается достичь, лишь существенно изменив алгоритм. Чтобы проникнуть в творческую лабораторию создателя программы, нужно самостоятельно или под руководством выполнить всю черновую работу алгоритмизации.

Ориентируясь на применение ПМК в школе, мы изложили в ч. I основы программирования на ПМК. Первая часть, однако, не только работает на вторую, но имеет и независимое значение. Чтобы облегчить обучение и навести «понятийные мости», рассказ об основных командах ПМК сопровождается сопоставлением этих команд с командами учебно-алгоритмического языка (УАЯ), изу-

чаемого в школе. Это поможет начинающим составлять эффективные алгоритмы, что, в свою очередь, положительно скажется на интересе школьников — главном двигателе учения.

Учебно-алгоритмический язык занимает промежуточное положение между языками ПМК и компьютера. По своим выразительным возможностям он достаточно близок к языкам программирования высокого уровня, в нем удачно интерпретируются программы для ПМК. С этих позиций изучение программирования по схеме ПМК — УАЯ — персональный компьютер представляется особенно целесообразным. Алгоритм, обладая обозримостью и выразительными средствами, близкими к естественному языку, выполняет по отношению к программе роль машинонезависимых схем. С другой стороны, программы для ПМК наглядно показывают, как формальный блок алгоритма развертывается во вполне конкретные указания машины.

В сопоставлении с УАЯ наглядно выявляются неразвитость языка ПМК, ограниченность его памяти. Это не позволяет учащимся долго задерживаться на достигнутом уровне, активизирует их стремление освоить более сложную вычислительную технику. В то же время, приступая в 10 классе к изучению программирования на ЭВМ, учащиеся смогут опереться на уже полученные конкретные знания.

Все алгоритмы, используемые или разработанные авторами для решения игровых и логических задач, могут быть успешно преобразованы в программы для таких микроЭВМ, как, например, «Электроника МК-85», «Микроша», «Электроника БК-0010»; «Агат», «Электроника БК-0011», «Корвет». Поскольку ЭВМ обладают более широкими возможностями, изменяются и критерии оптимальности, выработанные при использовании ПМК, однако к рассмотренным алгоритмам полезно вернуться в 10 классе, программируя на Бейсике или других языках.

В книге предусмотрена единая методика описания программ. Алгоритм дается в словесном описании или алгоритмической нотации УАЯ, в более сложных случаях приводится его схема. Программы сопровождаются инструкцией по вводу исходных данных и комментарием к ней. Нередко поясняется структура программы, отдельные ее фрагменты. Для быстрой проверки правильности введенной программы приводятся контрольные примеры. Большинство программ дополнены заданиями для самостоятельного решения, пояснениями, ответами. К некоторым задачам разработаны рекомендации, как изменить программу, чтобы достичь нужного эффекта. В ряде случаев приводится теоретический материал, необходимый для понимания алгоритмов и программ. В приложении приведены наиболее сложные задачи школьного курса основ информатики и вычислительной техники.

Все программы разработаны для отечественных программируемых микрокалькуляторов «Электроника» МК-56, МК-54 и без ка-

ких-либо изменений подходят к ПМК «Электроника Б3-34», для которого накоплена богатая литература по программированию. С небольшими изменениями программы можно использовать и для новых моделей «Электроника» МК-61 или МК-52. Возможен перевод программ и для распространенных ПМК фирмы Hewlett-Packard, в том числе и для модели HP-41. Поскольку обозначения на клавишиах разных моделей микрокалькуляторов несколько различаются, приводим таблицу, которая поможет ввести программу в любой из перечисленных отечественных ПМК (табл. 1).

Авторы благодарны Курскому областному отделу народного образования за помощь в организации экспериментального обучения информатике в школах области по материалам этой книги, а также учителям, принимавшим участие в эксперименте, М. И. Данеличу и Т. Е. Коневой.

Таблица 1

МК-54, МК-56, МК-61, МК-52	Б3-34	Принятые в книге
P→X	IП	P→X
X→P	П	X→P
↔	XY	↔
B↑	↑	B↑
F <sub>sin</sub> <sup>-1</sup>	F arcsin	F sin <sup>-1</sup>
F <sub>cos</sub> <sup>-1</sup>	F arccos	F cos <sup>-1</sup>
F tg <sup>-1</sup>	F arctg	F tg <sup>-1</sup>
÷	÷	:
F A	F	F

## ВВЕДЕНИЕ

Эта книга адресована всем, осознавшим необходимость овладения компьютерной грамотностью. Конечно, невозможно в полной мере удовлетворить пожелания всех потенциальных читателей. Не существует, например, единого критерия оптимальности программы. С точки зрения инженера — это компактность, универсальность, скорость решения по программе. Для учащегося на первый план выдвигается простота, доступность, близость к естественным формам решения. Стержневым в книге по программированию, естественно, является понятие алгоритма.

Под алгоритмом понимают предписание, какие действия и в какой последовательности необходимо выполнять в ходе решения. С появлением первых ЭВМ понятие алгоритма обрело общенаучную значимость: машинные программы — не что иное, как алгоритмы, изложенные на языке ЭВМ.

Можно привести сколь угодно много примеров алгоритмов: решение квадратного уравнения, распознавание делимости чисел, нахождение пути при равноускоренном движении, вычисление молекулярной массы соединения, вывод спутника на орбиту и т. п. Все эти задачи имеют свои алгоритмы решения.

В алгоритмах учитываются возможности исполнителя — человека или ЭВМ. Один алгоритм рассчитан на скромные возможности калькулятора, другой — на память ЭВМ, такой, например, как «Агат», БК-0010. Объединяет алгоритмы одно — исполнитель (человек или ЭВМ) может решить задачу только в том случае, если будет тщательно исполнять все предписания алгоритма. В алгоритме каждая переменная имеет имя — обозначение, а при его исполнении, кроме того, получает конкретное значение, не обязательно словесное.

Итак, в соответствии с алгоритмом ЭВМ однозначно преобразует исходные данные и ставит им в соответствие некоторые результаты. Это напоминает школьное определение функции, с той лишь разницей, что в качестве переменных могут использоваться не только числа.

Алгоритм, адресуемый конкретному исполнителю, обладает следующими свойствами.

**Дискретность.** Алгоритм — это последовательность четко отчененных друг от друга шагов. Свойство дискретности встречается и в природных явлениях и процессах. Мы обнаруживаем его,

например, в механизме передачи наследственных признаков, где каждый конкретный признак организма определяется наследственным фактором, который при скрещивании не дробится, а передается как целое.

**Массовость.** Процесс исполнения алгоритма не зависит от содержания конкретной задачи. Если алгоритм относится более чем к одной задаче, то решения различаются только значениями переменных. Легко представить, какие трудности пришлось бы преодолевать, если для каждого квадратного уравнения приходилось бы придумывать особые способы решения, для каждой пары многозначных чисел — свой способ сложения. Все формулы в физике, математике и других науках удовлетворяют требованию массовости. Присваивая аргументам различные значения, получают различные реализации формул. Отличие алгоритма от его реализации такое же, как между формулой и арифметическим примером на ее применение.

**Понятность.** Чтобы исполнитель мог воспользоваться алгоритмом, его указания должны быть понятны. Многолетняя война в Лилипутии между «тупоконечниками» и «остроконечниками» за право диктовать, с какого конца разбивать яйцо, с тупого или острого, завершилась примирением на том, что разбивать яйцо следует с того конца, с какого удобнее. Как мы знаем, война в Лилипутии разгорелась с новой силой. И все потому, что не оговорили, с какого именно конца удобнее разбивать яйцо, т. е. указание алгоритма не удовлетворяет требованию понятности.

Однако, когда алгоритм формируется в расчете на человека, одна лишь научная корректность ничего не решает — необходимо учитывать психологические механизмы переработки информации человеком, в частности связанные с усвоением алгоритмов. Вот конкретный пример: «Произведение одночлена и многочлена тождественно равно сумме произведений этого одночлена на каждый член многочлена»\*. В этом правиле-алгоритме не сформулировано явно, как именно получают произведение. Правило демонстрирует знание вне действия. Учащиеся легко заучивают правило, но не умеют им пользоваться. Нарушено требование понятности алгоритмического предписания. Впрочем, в том же учебнике есть и алгоритм, где выполняется требование понятности:

«Чтобы умножить одночлен на многочлен, достаточно умножить одночлен на каждый член многочлена и полученные произведения сложить». Это уже руководство, позволяющее перейти от знания к действию.\*\*

**Доступность.** Только что обсуждавшееся свойство понятности ориентировано на человека и неприемлемо для ЭВМ. Однако и

\* Алгебра-6. — М.: Просвещение. — 1982.

\*\* В издании учебника 1985 г. замечание учтено. Однако аналогичные нарушения принципа алгоритмизации в обучении допущены на стр. 11, 78, 142, 147 и др.

для человека даже вполне понятное задание может оказаться практически невыполнимым, как, например, доступное и несложное для ЭВМ указание выполнить 200 сложений в секунду.

**Результативность.** Как бы долго ни продолжалось решение задачи, оно должно завершиться некоторым результатом в приемлемое для исполнителя время. Только для перечисления всех натуральных чисел от 1 до  $10^{10}$  человеку не хватит всей жизни. Современная ЭВМ справится с задачей всего за несколько часов. Как видим, для ЭВМ процесс результативен. Результативность есть не что иное, как свойство доступности, отнесенное не к одному действию, а к алгоритму в целом.

**Детерминированность.** Взаимосвязь действий при решении задачи, когда результат и процесс решения не зависят от исполнителя. Это свойство и позволяет передать решение автомату. Иногда детерминированность проявляется в поведении человека, особенно в динамических стереотипах. Рассказывают, что известный математик Давид Гильберт однажды, зайдя в спальню, чтобы переменить сорочку, снял пиджак, затем галстук, потом сорочку и в силу естественной последовательности действий ... лег в постель. Детерминированность присуща всем действиям ЭВМ. Для человека детерминированность, напротив, является антиподом гибкости мысли. Джанни Родари в «Приключениях Чиполлино» приводит случай, когда графский пес должен был открыть ворота парка, чтобы через них проник кавалерийский эскадрон. Но план рухнул: ворота оказались открытыми. Животно-растительному разуму добрых героев казалось не по плечу преодоление детерминированности в выполнении плана. С другой стороны, свойству детерминированности алгоритма мы обязаны возможностью передачи автоматам решения некоторых задач, требующих значительно го рутинного труда. За собой человек оставляет только творческую часть мыслительного процесса.

Логические машины Луллия и А. Н. Щукарева, счетные машины Паскаля и П. Л. Чебышева, программный автомат Ч. Бебиджа — вехи на пути к появлению ЭВМ. В отличие от всех предшествующих автоматов, ЭВМ, следуя введенной в нее программе-алгоритму, может работать без вмешательства человека — рассматривать и анализировать варианты, отбирая по заданным критериям оптимальные.

ЭВМ оказалось по плечу решение многих задач, считавшихся ранее доступными только для человека. В 1968 г. впервые в нашей стране было зарегистрировано научное открытие тонких высокотемпературных слоев с особыми свойствами в плазме, движущейся в магнитном поле. Условия возникновения слоев были предсказаны ЭВМ при моделировании процесса в вычислительном эксперименте. Опытный вычислитель выполняет 600 операций в день, т. е. 10 млн. операций за 50 лет непрерывной работы. Быстро действующая ЭВМ сделает это за доли секунды. В день она

производит больше вычислений, чем человек мог бы сделать за несколько тысяч лет. Быстродействие ЭВМ растет с каждым годом, позволяя решать все новые и новые задачи, что ранее было невозможно из-за невероятного объема необходимых расчетов.

Одним из научных направлений, возникших на базе современной вычислительной техники, является информатика, изучающая структуру и общие свойства научной информации: закономерности ее создания, передачи, преобразования и использования в различных сферах человеческой деятельности. Информатика изучает как качественную, так и количественную сторону процесса организации информации. Однако информатика не разрабатывает критерии оценки истинности, новизны, полезности информации, методов ее переработки с целью получения новой информации. Как видим, в информатику не входят процессы человеческого творчества.

Однако в обучении информатике и с помощью информатики нельзя не учитывать особенности человеческого восприятия, поскольку терминал (окончное устройство ЭВМ) не есть конечный пункт системы обучающийся — ЭВМ. Добиваясь перехода от обучения программированию к программированию обучения, нельзя игнорировать психологические механизмы переработки информации человеком. Для эффективного использования алгоритма обучающемуся необходимо, чтобы алгоритмический язык был по возможности ближе к родному языку, что следует из теории известного психолога Л. С. Выготского. Именно такой, учитывающей особенности человеческого мышления, принципложен в основу пробного учебного пособия «Основы информатики и вычислительной техники» [12].

Учебно-алгоритмический язык — это система обозначений и правил для единообразной записи алгоритмов. Язык ориентирован на учащегося, являясь переходной формой от естественного языка, на котором они разговаривают, к программам для ЭВМ, содержит основные конструкции человеческого языка: знаки, слова, предложения, тексты и математическую символику: числа, функции, математические операции, отношения равенства, неравенства. Неделимым базовым элементом УАЯ является знак. Со-вокупность знаков — алфавит — используется для образования других разрешенных конструкций: слов, команд, программ и др. В УАЯ строится алгоритм работы с величинами. Величина имеет имя и значение. Если величина принимает разные значения в процессе реализации алгоритма, то она является переменной. В противном случае ее считают константой. Система обозначений, принятая для формальной записи алгоритмов, называется алгоритмической нотацией. Она удобна для описания и анализа алгоритмов в понятиях человека и близка к машинным языкам. Единая нотация возможна потому, что во всех алгоритмах есть общие слова, которые имеют один и тот же постоянный смысл. Их принято на-

зывать служебными или ключевыми. Следуя нотации УАЯ, будем записывать ключевые слова сокращенно и выделять их полужирным шрифтом. Два ключевых слова мы уже знаем: начало (**нач**) и конец (**кон**), с другими (алгоритм (**алг**), аргумент (**арг**), результат (**рез**), (**если**), (**то**), (**иначе**), (**все**), начало цикла (**нц**), конец цикла (**кц**)) можно познакомиться по учебным пособиям [12, ч. I, II].

В алгоритмах решения логических и игровых задач используются три дополнительных обозначения, необходимые для организации диалога человек — ЭВМ. Это ввод (**ввод**), вывод (**выв**) или индикация промежуточных и окончательных результатов и комментарий (**ком**), или пояснение значений переменных или отдельных фрагментов программы.

В главе 1 ключевые слова и составленные с их помощью команды сопоставляются с операторами языка ПМК. В дальнейшем будем понимать программирование на ПМК как кодирование в понятиях ПМК действий более высокого уровня, записанных в нотации УАЯ.

## Глава 1

### ПРОГРАММИРУЕМЫЙ МИКРОКАЛЬКУЛЯТОР «ЭЛЕКТРОНИКА МК-56»

#### 1.1. ПРИНЦИП РАБОТЫ

В основу ПМК положен принцип «бесконечная лента памяти». В кольцо, по которому непрерывно движется информация, включены устройства для обработки этой информации: два микроконтроллера, выполняющие необходимые вычисления, и два сдвиговых регистра, играющих роль динамической памяти. В микроконтроллерах имеется арифметико-логическое устройство (АЛУ), предназначенное непосредственно для вычислений.

Два числа, участвующие в операции, попадают в операционные регистры X и Y. Все операции могут выполняться только над содержимым этих регистров. Результат вычисления направляется в регистр X и одновременно выводится на индикатор. К операционным примыкают еще два регистра, Z и T. Все вместе они образуют стек. Числа в стеке в процессе вычислений сдвигаются в одном из двух направлений: вверх  $PX \rightarrow PY \rightarrow PZ \rightarrow PT \downarrow$ , когда в стек вызывается новое число, и вниз  $PT \rightarrow PZ \rightarrow PY \rightarrow PX \downarrow$  в момент выполнения операции. Часто к стеку причисляют регистр X1, куда при выполнении операций записывается прежнее содержимое регистра X.

В специальной области микроконтроллера — постоянном запоминающем устройстве (ПЗУ) — собрана вся информация, позволяющая ПМК «понимать» нажатия клавиш, вести вычисления — функционировать так, как того требует звание вычислительного устройства. Эту информацию нельзя стереть, отключив питание. Все то, что теряется при выключении ПМК — данные, программа, находится в оперативном запоминающем устройстве (ОЗУ). Оперативная память у ПМК «Электроника МК-56» динамическая, кольцевая, поэтому информация обрабатывается как

бы «на лету», нужные данные извлекаются от бесконечной их вереницы.

В кольцо замкнуты две области памяти — регистровая и программная. В регистровой памяти хранится содержимое 14 регистров — 0, 1, 2, ..., 9, А, В, С, Д (последним четырем регистрам соответствуют числа от 10 до 13). В каждый из них можно записать любое необходимое число, чтобы использовать их в дальнейших вычислениях. Программная память служит для хранения кодированной записи текста программы. Текст, как и данные, можно вводить с клавиатуры. Команды нумеруются числами от 00 до 97 — адресами команд. Нетрудно подсчитать, что всего в программе может быть 98 команд.

## 1.2. ВВОД И ПРЕДСТАВЛЕНИЕ ЧИСЕЛ. ОБРАЩЕНИЕ К РЕГИСТРАМ ПАМЯТИ

Для управления режимами работы и ввода в ПМК данных и команд предназначены 30 клавиш на передней панели. Однако с их помощью можно выполнить или закодировать около 200 различных операций. Многие операции вводятся нажатием клавиш в различных комбинациях. Обозначения части операций вынесено на панель пульта управления: их можно видеть *на*, *над* и *под* клавишами. Клавиши объединены в три секции: центральную — 12 клавиш, левую — 10, правую — 8.

Включите ПМК в сеть и подготовьте его к работе (выключатель питания расположен справа над индикатором). Если ПМК исправен, в левой части индикатора появится 0.

Ввод чисел всегда производится в операционный регистр X, т. е. на индикатор. Для этого предназначены все клавиши центральной секции и одна клавиша правой секции. Речь идет о цифровых клавишах 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 и клавише /—/ (Смена знака мантиссы или порядка), а также клавише ВП (Ввод порядка). Нажмите последовательно 2 5 4, на индикаторе высветится 254. Чтобы очистить регистр X (и индикатор), нажмите красную клавишу CX (правая секция), на индикаторе появится 0. Можно ввести десятичную дробь, отрицательное число. Рис. 1 поможет потренироваться вводить числа. Важно привыкнуть вначале вводить число, а лишь затем знак, если число отрицательное, поскольку традиционный порядок ввода ПМК не допускает.

Для отображения чисел ПМК снабжен 12-разрядным индикатором. Десятичные дроби, не начинающиеся с нуля и имеющие не более восьми цифр, представляются в естественной форме, т. е. показания индикатора могут выглядеть так: 3.1415926 или 10. Наибольшее по абсолютной величине число, которое можно записать в ПМК в естественной форме, это 99999999 или  $10^8 - 1$ ,

Требуется ввести	Нажимаемые клавиши	Показания индикатора
123	1 2 3	123.
12,3	1 2 . 3	12.3
$12,3 \cdot 10^{-4}$	1 2 . 3 BT 4 /—/	12.3 -0,4
$-0,123 \cdot 10^{-2}$	0 . 1 2 3 /—/ BT 2 /—/	-0.123 -0,2

Рис. 1

наименьшее — 1. Таким образом, естественная форма сохраняется в диапазоне  $1 \leq x \leq 99999999$ .

Для отображения других чисел используется представление в форме с плавающей точкой. Известно, что любое десятичное число можно записать в виде  $x = m \cdot 10^n$ , где  $m$  — мантисса,  $n$  — порядок числа. Запись неоднозначна:  $0,0356 = 0,356 \cdot 10^{-1} = 0,000356 \times 10^2$  и т. д. Условимся фиксировать точку после первой не равной нулю значащей цифры. Тогда  $0,0356 = 3,56 \cdot 10^{-2}$ . Это и есть представление числа в форме с плавающей точкой, где  $m = 3,56$ ,  $n = -2$ . Три крайних разряда в правой части индикатора служат для индикации порядка: первый разряд — знаковый, второй и третий отображают порядок числа от —99 до 99. На индикаторе могут отображаться числа в промежутке от  $10^{-99} \leq x \leq 9,999999 \cdot 10^{99}$ .

Описанное представление чисел называют нормализованным. В какой бы форме числа ни вводились, ПМК автоматически нормализует их. Набрав, например, 0,036 и нажав клавишу В↑ (Копирование в регистре Y), получим  $3,6 \cdot 10^{-2}$  — число нормализовано. На индикаторе это число выглядит так: 3.6 —02. Еще пример:  $-25 \cdot 10^4$  — нажимаем 2 5 /—/ ВП 4, после нормализации (В↑) получаем —250000.

Для хранения оперативных данных предусмотрена регистровая память, состоящая из 14 адресуемых регистров: 0, 1, ..., 9, А, В, С, Д. Числа в регистры могут попадать только из регистра X. Чтобы запомнить число в регистрах с 0 по 9, необходимо нажать клавиши сначала X→P, а затем соответствующую цифровую. Чтобы записать число в регистры А, В, С и Д, необходимо нажать клавишу X→P, а затем ту из клавиш, под которой сделана соответствующая надпись. Каждое обращение к регистру командой X→P уничтожает его прошлое содержимое и записывает в него текущее содержимое регистра X.

Пример. Записать число 23 в регистр 4 и число 3 в регистр С. Нажимаем клавиши 2 3 X→P 4 — в Р4 и на индикаторе число 23; 3 X→P С — в РС и на индикаторе число 3.

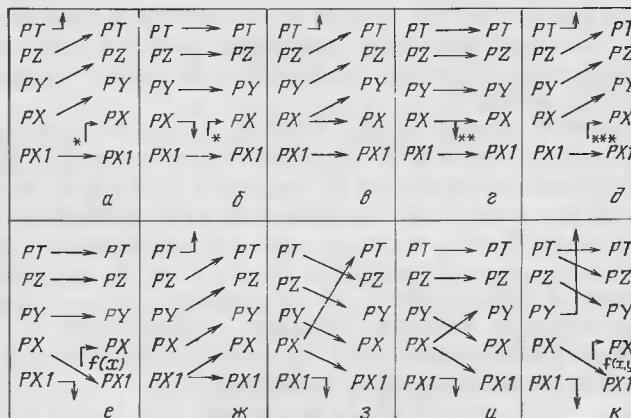
Составляя алгоритм, часто используют запись С := 3, обозначающую операцию присвоения регистру С в качестве содержимого числа 3. Можно встретить и такую запись (С) = 3, означающую, что содержимое регистра С равно трем.

Чтобы вызвать число из регистра памяти в операционный регистр X (на индикатор), следует воспользоваться клавишей П→X. Тренируясь во вводе чисел, мы записали число 23 в регистр 4. Теперь можно вызвать его в регистр X (и на индикатор), нажав клавиши П→X 4. Этую операцию будем обозначать как X := (4), где 4 — номер регистра.

### 1.3. ОРГАНИЗАЦИЯ ВЫЧИСЛЕНИЙ В СТЕКЕ

Свое название стек получил из-за организации записи и считывания информации по принципу «последним зашел — первым вышел». Рассмотрим, что это значит. Стек состоит из четырех регистров — X, Y, Z, T. К стеку может быть также отнесен регистр X1 — регистр предыдущего результата, поскольку он выполняет схожие функции. Работу стека удобно рассматривать в типовых ситуациях (рис. 2, а—к). На каждом рисунке в левой позиции — содержимое регистров стека до выполнения операции, в правой — после.

После включения регистры стека и регистр предыдущего результата находятся в нулевом состоянии. Набор числа на индикаторе означает заполнение регистра X по схеме рис. 2, а либо 2, б. Последний случай имеет место, если перед набором числа была нажата клавиша В↑. Нажатие клавиши В↑ автоматически нормализует число, находящееся в регистре X (на индикаторе), и передает копию числа, находящегося в регистре X, в регистр Y. При этом содержимое регистра T стирается из памяти, содер-



\* — вводим с клавиатуры

\*\* — копируем в память

\*\*\* — копируем из памяти

f(·) — вычисляется по микропрограмме

Рис. 2

жимое (прежнее) РY переписывается в РZ, а содержимое РZ — в РT (рис. 2, в).

Если два числа вводятся с клавиатуры подряд, то они разделяются командой В↑, перемещающей первое число в РY, иначе вместо двух получится одно число. Если число вводится после некоторой операции (арифметической, вычисления функции, ↔, X→P, P→X и др.), то оно само перемещает предшествующее содержимое РX в РY согласно рис. 2, а.

Запись числа в адресуемый регистр памяти не изменяет содержимого стековых регистров и регистра X1, информация копируется из РX в заданный регистр (рис. 2, г). Копирование из адресуемого регистра памяти вызывает изменение содержимого стека в соответствии со схемой (на рис. 2, д). При выполнении действий с одной переменной (одноместной операции) значение функции записывается в регистр X, а сам аргумент переписывается в регистр X1, число, ранее находившееся в РX1, при этом теряется (рис. 2, е).

Вызвать число из регистра XI в регистр X можно командой FBx (Восстановление икса). При этом (как видно из рис. 2, ж) содержимое РX1 копируется в РX, происходит сдвиг информации «вверх» по стеку, а содержимое РT теряется. Кольцевое перемещение информации в стеке достигается набором команды FO и сопровождается систематической сменой информации в ячейках стека (рис. 2, з).

Двухместные операции (+; —; ×; : ; ↔) требуют для размещения операндов регистры X и Y, результат записывается в РX; операнд из РY теряется, операнд, бывший в РX, отправляется в РX1, замещая его прежнее содержимое (рис. 2, к). Движение информации в стеке после нажатия ↔ отражено на рис. 2, и. Использованием является двухместная операция  $x^y$ , вычисление которой изменяет содержимое стека в соответствии с рис. 2, е.

Обратите внимание: при сдвиге информации «вниз» по стеку содержимое РT копируется в РZ, т. е. в РT остается как бы «след» числа, находившегося в нем последним до «спуска» в РZ.

Для вычисления встроенных функций используется префиксная клавиша F, знакомая нам по команде FBx. Нажав эту клавишу, вы указываете ПМК, что требуется выполнить команду, обозначение которой нанесено над второй нажимаемой клавишей.

Пример. Вычислить  $\sqrt{5}$ , tg (0,3).

Нажимаемые клавиши

Показание индикатора

5	F	$\sqrt{\phantom{x}}$	2.2360679	
0	.	3	F tg	5.2360355—03

ПМК вычисляет элементарные функции по микропрограммам, хранящимся в ПЗУ одного из микроконтроллеров.

Перед вычислением тригонометрических функций следует проверить положение переключателя Г—Р—ГРД, расположенного справа на передней панели. Если переключатель находится в крайнем левом положении, то ПМК производит операции в градусах, если в среднем — в радианах, в крайнем правом положении — в градах (последовательность указана для МК-56, в других ПМК она может быть иной). Единица измерения углов 1 град — это  $1/400$  угловой меры окружности, в окружности 400 град, т. е. в прямом угле 100 рад,  $1^\circ = 10/\pi$  град.

Стек помогает решать следующие задачи. Рассмотрим примеры решения этих задач.

**Задача 1.** Возвведение числа в целую положительную степень. Использование стека позволяет получить результат более точный, чем при использовании встроенной функции  $Fx^y$ . Чтобы набрать число, которое необходимо возвести в степень, нужно трижды нажать клавишу  $B\uparrow$ , заполнив стек копиями исходного числа, а затем  $n - 1$  раз нажать клавишу умножения ( $n$  — это показатель степени). При этом число будет передвигаться по стеку в соответствии со схемой на рис. 2,к. В программе этот метод не рационален. До степени не больше шестой он может использоваться в модернизированной форме (табл. 2).

Таблица 2

Вычислить	Нажимаемые клавиши
$x^3$	$B\uparrow F X^2 \times$
$x^4$	$F X^2 F X^2$
$x^5$	$B\uparrow F X^2 F X^2 \times$
$x^6$	$F X^2 B\uparrow F X^2 \times$
$x^8$	$F X^2 F X^2 F X^2$
$x^9$	$B\uparrow F X^2 F X^2 F X^2 \times$

Пример.  $2^5 ? 2 B\uparrow B\uparrow B\uparrow \times \times \times$  Результат «32» (здесь и далее кавычки означают «читаем на индикаторе»).

**Задача 2.** Алгебраическое сложение произведений и частных от деления двух и более чисел. Числа вводятся слева направо, т. е. в естественной последовательности.

Пример.  $2 \cdot 3 - 8/5 = ?$

Результат: 4,4.

Нажимаемая клавиша	2	$B\uparrow$	3	$\times$	8	$B\uparrow$	5	:	-
Схема на рис. 2	a	v	b	k	a	v	b	k	k

**Задача 3.** Нахождение алгебраической суммы квадратов чисел. Вычислить выражение вида  $\sqrt{x^2 + y^2}$ .

Выполняется следующая цепочка действий:  $x FX^2 y FX^2 + FV^-$ . Для  $x=3, y=4$  получается 5.

**Задача 4.** Деление алгебраической суммы на алгебраическую сумму. Вычислить выражение  $\frac{x_1 - x_2 x_3}{x_4 x_5 + x_6}$ , где  $x_1, x_2, x_3, x_4, x_5, x_6$  — любые числа. При вычислениях сохраняется естественный порядок ввода чисел.

Пример.  $(7 - 2 \cdot 4) / (5 \cdot 3 + 6) = ?$

Нажимаемая клавиша	7	$V\uparrow$	2	$V\uparrow$	4	$\times$	-	5	$V\uparrow$	3	$\times$	6	+	:
Схема на рис. 2	a	v	b	v	b	k	k	a	v	b	k	a	k	k

Результат:  $-4,7619047 \cdot 10^{-2}$ .

**Задача 5.** Вычисление алгебраической суммы, произведения и частного функций. В зависимости от вида выражения рекомендуем пользоваться методом, показанным в примере 1 или 2.

Пример 1.

Нажимаемая клавиша	5	$F \lg$	3	$F \lg$	7	$FX^2$	$\times$	+
Схема на рис. 2	a	e	a	e	a	e	k	k

Результат: 24,077909. Восстановите условие примера.

Пример 2.  $\exp(\cos(\lg \sqrt{3})) = ?$  (в традиционной школьной записи  $e^{\cos(\lg \sqrt{3})}$ ). Переключатель Р—Г установить в положение Р. Здесь вычисления следует производить справа налево.

Нажимаемая клавиша	3	$FV^-$	$F \lg$	$F \cos$	$F e^x$
Схема на рис. 2	a	e	e	e	e

Результат: 2,6423776.

**Задача 6.** Выполнение вычислений со скобками. Калькулятор со стеком иногда называют устройством с изменяемой бескодочной записью. Важно, производя вычисления, следить за глубиной заполнения стека, чтобы не допустить потери необходимой информации из регистра Т.

Пример.  $2+7\cdot 3+6\cdot(5-4)=?$

Нажимаемая клавиша	2 В↑ 7 В↑ 3 × + 6 В↑ 5 В↑ 4 — × +
Схема на рис. 2	а в б в б к к а в б в б к к к

Результат: 29.

Следует стараться заполнить стек каждый раз на возможно меньшую глубину. Это достигается ценой потери удобства естественного порядка ввода данных. Решим этот же пример, вводя данные в соответствии с порядком выполнения действий:

Нажимаемая клавиша	7 В↑ 3 × 2 + 5 В↑ 4 — 6 × +
Схема на рис. 2	а в б к а к а в б к а к к

Результат: 29.

Нетрудно заметить, что, отказавшись от ввода чисел по порядку, мы выиграли в количестве нажатий. Превышение глубины стека приводит к неверному результату.

Пример.  $2+3+4+5+6=?$

Нажимаемая клавиша	2 В↑ 3 В↑ 4 В↑ 5 В↑ 6 + + + +
Схема на рис. 2	а б в б в б в б в к к к

Результат: 21.

При умелом использовании стек кажется бесконечным. Так, выражение вида  $(x_1+x_2(x_3+x_4(x_5+\dots x_{i-2}(x_{i-1}+x_i))\dots))$  может быть вычислено следующим образом:

Нажимаемая клавиша	$x_i$ В↑ $x_{i-1}$ + $x_{i-2}$ ×   ... $x_3$ + $x_2$ × $x_1$ +
Схема на рис. 2	а в б к а к   а к... а к а к а к

#### 1.4. ЧТО НЕЛЬЗЯ ВЫЧИСЛИТЬ НА ПМК

Если попытаться поделить любое число на нуль, извлечь корень из отрицательного числа или возвести отрицательное число или нуль в произвольную степень, на индикаторе появится

ЕГГОГ — сообщение об ошибке. Это произойдет и в ряде других случаев: если вычислять  $\lg x$  или  $\ln x$  для  $x \leq 0$ ,  $\operatorname{tg} x$  при  $x = \pi/2 \pm \pi n$ ,  $\sin^{-1} x$  и  $\cos^{-1} x$  для  $|x| > 1$ . Аналогичное сообщение получится и в том случае, когда результат выполнения операции превышает  $9,9999999 \cdot 10^{99}$ .

Важной характеристикой вычислительных устройств наряду с емкостью памяти является скорость работы — быстродействие. Арифметические операции и операции  $x^2$ ,  $\sqrt{x}$ ,  $1/x$  выполняются в среднем за полсекунды, а остальные функции — за 2—3 с. Такое низкое быстродействие ПМК объясняется особенностями устройства его памяти, замкнутой в кольцо, однако оно вполне достаточно для решения тех задач, которые умещаются в его ограниченную память.

#### УПРАЖНЕНИЯ

1. Найдите решение, не записывая промежуточных результатов. Важно уделять внимание движению чисел в сетке.

Пример.  $2,85+4,739?$  Ответ: 7,589. Решение: 2.85 В↑ 4.739 +.

1.1.  $4,76/0,284?$  «16.760563».

1.2.  $(3,15-2,96 \cdot 12,125)^2?$  «1071.9076».

1.3.  $\lg 7,6-(3 \sin^2 25^\circ)/2?$  «0.61290428». Переключатель установите в положение Г. Введите 7.6 F lg 3 В↑ 25 F sin FX<sup>2</sup> ... Если проследить продвижение чисел в стеке, то нетрудно заметить, что в регистре X записано число  $\sin^2 25^\circ$ , в регистре Y — число 3, а в регистре Z — число  $\lg 7,6$ . Чтобы получить верный ответ, осталось нажать клавиши  $\times$  2 : —.

1.4.  $3,96e^2 + \sqrt{10/tg 2}?$  «28.569691». Переключатель установите в положение Р.

1.5.  $\sqrt{8,71 \cdot 2,96 - 3,74}?$  «4.6948482».

1.6.  $0,479^2\pi?$  «0.72081014».

2. Вычислите площадь треугольника, если заданы три его стороны:  $a = 2,87$ ,  $b = 3,76$ ,  $c = 1,984$ . Для решения удобно воспользоваться формулой Герона  $S = \sqrt{p(p-a)(p-b)(p-c)}$ , где  $p$  — полупериметр. Если решить задачу в общем виде, то последовательность нажимаемых клавиш останется неизменной, какими бы значениями  $a$ ,  $b$ ,  $c$  мы не задавались. Поместим данные из нашего примера в соответствующие регистры: 2.87 X→ПА, 3.76 X→ПВ, 1.984 X→ПС. Значение полупериметра направим в РД:  $p$  X→ПД.

Запишем цепочку клавиш, которые придется нажать:

2.87 X→ПА 3.76 X→ПВ + 1.984 X→ПС + 2 : X→ПД П→ХД П→ХА — X→П→ХД П→ХВ — X→П→ХД П→ХС — X→F V.

Верный ответ:  $S = 2,8043621$ .

Вычислите площадь треугольника с другими длинами сторон.

#### 1.5. РЕЖИМ ПРОГРАММИРОВАНИЯ. ВВОД ПРОГРАММЫ И ЕЕ ОТЛАДКА

ПМК может работать в одном из двух режимов: вычислений и программирования. После включения ПМК находится в режиме вычислений. До сих пор мы работали с ПМК только в этом режиме.

Чтобы ввести в ПМК программу, нужно перевести его в режим программирования. Делается это нажатием двух клавиш F и ПРГ. Тотчас же в правом углу индикатора появится 00. Это значит, что счетчик адресов установлен на нулевую отметку и команда, которую мы введем первой, запишется в ячейку с адресом 00. В любой момент можно возвратиться в режим вычислений, нажав клавиши F и АВТ.

В режиме программирования в ПМК вводят текст программы, можно заменять отдельные фрагменты программы при ее отладке (коррекции). После набора очередной команды содержимое счетчика адресов автоматически увеличивается на единицу.

Когда программа введена полностью, ее можно проконтролировать, нажимая клавишу  $\leftarrow$  ШГ. Только убедившись в отсутствии ошибок, можно переходить в режим автоматических вычислений.

Любая программа, т. е. описание алгоритма на языке вычислительного устройства, состоит из операторов. *Оператором* называют символ или группу символов, обеспечивающих выполнение одной элементарной операции. Оператор подобен слову в языке: прибавить к нему другие слова можно, а дробить так, чтобы сохранился смысл, нельзя.

Если в режиме программирования нажимать клавиши в естественном порядке, воспроизводящем ход расчетов вручную, в ПМК будут вводиться соответствующие операторы. Так, нажатиями

Код	0	1	2	3	4	5	6	
0	0	1	2	3	4	5	6	
1	+	-	$\times$	:	$\leftrightarrow$	$F10^x$	$Fe^x$	
2	$F\pi$	$F\sqrt{ }$	$FX^2$	$F1/X$	$FX^y$	$F0$	$K0'*$	
3	$K0'''*$	$K X *$	$K3H^*$	$K0'^*$	$K[X]^*$	$K\{X\}^*$	$K \max^*$	
4	$X \rightarrow \Pi 0$	$X \rightarrow \Pi 1$	$X \rightarrow \Pi 2$	$X \rightarrow \Pi 3$	$X \rightarrow \Pi 4$	$X \rightarrow \Pi 5$	$X \rightarrow \Pi 6$	
5	$C/\Pi$	$B\Pi$	$B/O$	$\Pi\Pi$	$KНОП$			
6	$\Pi \rightarrow X 0$	$\Pi \rightarrow X 1$	$\Pi \rightarrow X 2$	$\Pi \rightarrow X 3$	$\Pi \rightarrow X 4$	$\Pi \rightarrow X 5$	$\Pi \rightarrow X 6$	
7	$KX \neq 00$	$KX \neq 01$	$KX \neq 02$	$KX \neq 03$	$KX \neq 04$	$KX \neq 05$	$KX \neq 06$	
8	$KБП0$	$KБП1$	$KБП2$	$KБП3$	$KБП4$	$KБП5$	$KБП6$	
9	$KX \geq 00$	$KX \geq 01$	$KX \geq 02$	$KX \geq 03$	$KX \geq 04$	$KX \geq 05$	$KX \geq 06$	
-	$KПП0$	$KПП1$	$KПП2$	$KПП3$	$KПП4$	$KПП5$	$KПП6$	
L	$KX \rightarrow \Pi 0$	$KX \rightarrow \Pi 1$	$KX \rightarrow \Pi 2$	$KX \rightarrow \Pi 3$	$KX \rightarrow \Pi 4$	$KX \rightarrow \Pi 5$	$KX \rightarrow \Pi 6$	
C	$KX < 00$	$KX < 01$	$KX < 02$	$KX < 03$	$KX < 04$	$KX < 05$	$KX < 06$	
G	$KП \rightarrow X 0$	$KП \rightarrow X 1$	$KП \rightarrow X 2$	$KП \rightarrow X 3$	$KП \rightarrow X 4$	$KП \rightarrow X 5$	$KП \rightarrow X 6$	
E	$KX = 00$	$KX = 01$	$KX = 02$	$KX = 03$	$KX = 04$	$KX = 05$	$KX = 06$	

двух клавиш F и sin в программу будет введен оператор  $Fsin$ , необходимый чтобы вычислить синус числа, находящегося в РХ. Известные нам операторы ввода числа в адресуемую память  $X \rightarrow \Pi N$  и извлечения числа из памяти  $\Pi \rightarrow X N$  будем называть *операторами присваивания*. К операторам присваивания относится также набор числа с клавиатуры в РХ.

Программа не может располагаться в ПМК иначе, как в виде отдельных команд, каждая из которых занимает свою ячейку программной памяти. Шаги программы нумеруются двузначными числами от 00 до 97 (в новых моделях ПМК 00 — 99, —0, —1, —2, —3, —4). Эти числа принято называть *адресами команд*.

Команды, которые вводятся в ПМК в режиме программирования, отображаются на индикаторе в закодированном виде. Коды команд состоят из цифр и знаков —, L, C, Г, Е (табл. 3). При вводе команды ее код отображается в левом углу индикатора. Для контроля высвечиваются коды двух предшествующих команд, в правом углу индикатора — адрес команды, следующей в программе за текущей.

Пример. На индикаторе 124—6С 15. Это означает, что последней введена команда умножения (код 12). Она размещена в программе по адресу 14, две предшествующие команды:  $X \rightarrow \Pi A$  и  $\Pi \rightarrow XC$ .

Если код на индикаторе не соответствует введенной команде, это значит, что при вводе допущена ошибка. В этом случае, на-

Таблица 3

7	8	9	-	L	C	Г	Е	
7	8	9	.	/—/	ВП	CX	B↑	FBX
$F\lg$	$F\ln$	$F\sin^{-1}$	$F\cos^{-1}$	$F\tg^{-1}$	$F\sin$	$F\cos$	$F\tg$	
$K\Lambda^*$	$K\vee^*$	$K\oplus^*$	$KИНВ^*$	$KСЧ^*$				
$X \rightarrow \Pi 7$	$X \rightarrow \Pi 8$	$X \rightarrow \Pi 9$	$X \rightarrow \Pi A$	$X \rightarrow \Pi B$	$X \rightarrow \Pi C$	$X \rightarrow \Pi D$		
$FX \neq 0$	$FL2$	$FX \geq 0$	$FL3$	$FL1$	$FX < 0$	$FL0$	$FX = 0$	
$\Pi \rightarrow X 7$	$\Pi \rightarrow X 8$	$\Pi \rightarrow X 9$	$\Pi \rightarrow X A$	$\Pi \rightarrow X B$	$\Pi \rightarrow X C$	$\Pi \rightarrow X D$		
$KX \neq 07$	$KX \neq 08$	$KX \neq 09$	$KX \neq 0A$	$KX \neq 0B$	$KX \neq 0C$	$KX \neq 0D$		
$KБП7$	$KБП8$	$KБП9$	$KБПA$	$KБПB$	$KБПC$	$KБПD$		
$KX \geq 07$	$KX \geq 08$	$KX \geq 09$	$KX \geq 0A$	$KX \geq 0B$	$KX \geq 0C$	$KX \geq 0D$		
$KПП7$	$KПП8$	$KПП9$	$KППA$	$KППB$	$KППC$	$KППD$		
$KX \rightarrow \Pi 7$	$KX \rightarrow \Pi 8$	$KX \rightarrow \Pi 9$	$KX \rightarrow \Pi A$	$KX \rightarrow \Pi B$	$KX \rightarrow \Pi C$	$KX \rightarrow \Pi D$		
$KX < 07$	$KX < 08$	$KX < 09$	$KX < 0A$	$KX < 0B$	$KX < 0C$	$KX < 0D$		
$KП \rightarrow X 7$	$KП \rightarrow X 8$	$KП \rightarrow X 9$	$KП \rightarrow X A$	$KП \rightarrow X B$	$KП \rightarrow X C$	$KП \rightarrow X D$		
$KX = 07$	$KX = 08$	$KX = 09$	$KX = 0A$	$KX = 0B$	$KX = 0C$	$KX = 0D$		

жав клавишу  $\overleftarrow{ШГ}$ , возвращаемся на шаг назад (адрес команды уменьшился на единицу) и вводим команду повторно. Эти действия называют *редактированием программы*.

Если ошибка обнаружена в команде, входящей в состав двухшагового оператора, то необходимо заново ввести оператор, возвратившись на два шага назад, дважды нажав клавишу  $\overleftarrow{ШГ}$ .

Может оказаться, что ошибка ввода обнаружена не сразу, например после набора всей программы. В этом случае установка программы на необходимый адрес производится также с помощью клавиш  $\overleftarrow{ШГ}$ ,  $\overleftarrow{\overline{ШГ}}$ , однако проще вернуться в режим вычислений ( $F$  АВТ) и, нажав клавиши БП  $N$  F ПРГ, установить счетчик адресов на адрес ( $N$ ) той команды, которую необходимо ввести повторно или заменить. На место лишнего оператора помещают специальный оператор КНОП (Нет операции). Эта команда играет роль «заплаты» и практически не влияет на ход выполнения готовой программы.

В целом процедура ввода и отладки программы выглядит так. Если программа начинается с адреса  $N$ , то, чтобы перейти в режим программирования, следует нажать клавиши БП  $N$  F ПРГ. При этом счетчик адресов устанавливается на адрес  $N$ . Если счетчик нужно установить на начало программы  $N=00$ , то вместо команд БП 00 F ПРГ удобно воспользоваться командами В/О F ПРГ. После того как набрана очередная команда, содержимое счетчика адресов автоматически увеличивается на единицу.

Когда программа введена полностью, ее можно проконтролировать, нажимая клавишу  $\overleftarrow{ШГ}$ . Шаг за шагом нужно проверить коды всех команд и, убедившись в отсутствии ошибок, перейти в режим автоматических вычислений.

Следует отметить, что все вычисления в ручном исполнении (как мы это делали раньше) или по введенной программе ПМК выполняет в режиме вычислений. В этом же режиме в регистры адресуемой памяти вносят данные, которые необходимы для работы программы. Оба режима независимы, поэтому не имеет значения, когда вводятся данные, до или после набора программы.

Расчет по программе запускают, нажав клавиши В/О С/П. Так можно делать всегда, когда она начинается с адреса 00. Если программа записана с адреса  $N$ , то вместо команд В/О С/П используют команды БП  $N$  С/П.

Когда программа составлена, введена и отредактирована, необходимо проверить ее работу на одном или нескольких контрольных примерах (тестах). Желательно, чтобы верный ответ был известен. Иногда необходимо проконтролировать вычисления по шагам. Особый интерес это представляет, если использовать ПМК в качестве устройства для исполнения алгоритмов на УАЯ. Для этого в режиме автоматических вычислений нажимают клавишу ПП,

которая в этом режиме выполняет лишь одну очередную команду программы. При этом на индикатор выводится результат исполнения этой команды, т. е. содержимое РХ. Начинать проверку можно с любого адреса.

Останов вычислений необходим для вывода на индикатор результатов расчета. Для этого в программу записывают оператор «Стоп», нажав в режиме программирования клавишу С/П. Нажатие этой же клавиши в режиме вычислений соответствует команде «Пуск».

Если при вычислениях обнаружится недопустимая операция, то ПМК остановится. На индикатор будет выведено сообщение ЕГГОГ. В ряде случаев останов может произойти не на той команде, которая привела к ошибке, а на следующей за ней, тогда после запуска вычислений одна команда окажется пропущенной.

## 1.6. АДРЕСАЦИЯ В ПМК

Мы уже познакомились с операторами присваивания  $X \rightarrow PA$ ,  $P \rightarrow XA$ . Такой способ обращения к памяти называется *прямой адресацией*. Однако есть иной способ обращения, когда адрес перехода записан в адресуемом регистре, — *косвенный*. Чтобы отличить косвенную адресацию от прямой, используют клавишу К. Косвенная адресация, например  $KX \rightarrow PA$ , означает передачу содержимого РХ регистру памяти, адрес которого записан в РА.

Запишем число 25 в Р6 с помощью команды косвенной адресации. Пусть адрес Р6 — число 6 хранится в РА: 6  $X \rightarrow PA$ . Теперь можно записать 25: 25  $KX \rightarrow PA$ . Очистим регистр X (СХ) и проверим содержимое Р6 ( $P \rightarrow X6$  «25»). Число 25 на индикаторе — признак того, что нам удалось записать число из РХ в Р6, передав его через регистр А.

При косвенном обращении модифицируется содержимое регистра, через который идет адресация. Если это регистры 7—9, А, В, С, Д, то модификация заключается в том, что у дробных чисел, больших нуля, отбрасывается дробная часть. Если в качестве адреса использовано целое число, то его значение не меняется. Модификация через регистры 4—6 сопровождается увеличением на единицу целой части, а через регистры 0—3, напротив, уменьшением на единицу целой части. В программах команды косвенного обращения через регистры 0—6 используют чаще всего тогда, когда нужно записать переменное содержимое РХ в разные регистры памяти или вызывать в РХ содержимое разных регистров.

Теперь мы готовы к составлению первой программы. Пусть требуется вычислять по формуле

$$(\sqrt[3]{a} - b(a+c^2)) / \sqrt[6]{b-c}.$$

Составим сначала программу на бумаге. В программу включим те же операции и в той же последовательности, как это дела-

ется в режиме непосредственного счета, указывая адреса и команды. Имеются только два существенных отличия от прежней последовательности — это команды с адресами 00 и 21. Первая из них появилась потому, что вычисление начинается с набора числа 3. Если к моменту начала вычислений на индикаторе будет ненулевое число, например 5, то число 3 будет воспринято ПМК как продолжение ввода, как если бы мы вводили число 53. Чтобы это исключить, мы сделали первой командой программы CX. Второе отличие связано с необходимостью закончить вычисления после того, как будет получен результат. Эту роль выполняет оператор C/P.

### Программа КОМБИНИРОВАННЫЕ ДЕЙСТВИЯ

<i>a</i>	X→ПА	<i>b</i>	X→ПВ	<i>c</i>	X→ПС	B/O	C/P
00	CX	0Г	07	Π→XC	6C	14	Π→XB 6L
01	3	03	08	FX <sup>2</sup>	22	15	Π→XC 6C
02	F1/X	23	09	+	10	16	— 11
03	Π→XA 6—	10	×	12	17	FX <sup>y</sup>	24
04	FX <sup>y</sup> 24	11	—	11	18	↔	14
05	Π→XB 6L	12	6	06	19	FO	25
06	Π→XA 6—	13	F1/X	23	20	:	13
					21	C/P	50

Следующий вариант программы позволяет уменьшить число команд на одну:

00	CX	0Г	07	Π→XC	6C	14	—	11
01	3	03	08	FX <sup>2</sup>	22	15	F In	18
02	F1/X	23	09	+	10	16	6	06
03	Π→XA 6—	10	×	12	17	:	13	
04	FX <sup>y</sup> 24	11	—	11	18	Fe <sup>x</sup>	16	
05	Π→XB 6L	12	Π→XB	6L	19	:	13	
06	Π→XA 6—	13	Π→XC	6C	20	C/P	50	

### Инструкция

1. B/O F ПРГ — переход в режим программирования, справа на индикаторе высвечиваются показания счетчика адресов — 00.

2. Ввод программы. Нажать клавишу CX — на индикаторе слева код 0Г, справа — адрес следующей команды 01. Нажать 3 — на индикаторе два кода 03 0Г, на счетчике адресов — 02 и т. д. После ввода последней команды счетчик адресов показывает 22, слева на индикаторе 50 — код команды C/P.

3. Редактирование программы. Чтобы возвратиться в режим вычислений, необходимо нажать клавиши F АВТ.

4. Ввод числовых данных в адресуемую память: 2.38 X→ПА, 3.7 X→ПВ, 1.96 X→ПС. Данные можно было бы ввести еще до набора программы, однако удобнее постоянно пользоваться каким-либо одним способом.

5. B/O C/P — запуск программы на счет. Через 12 с на индикаторе появляется результат — 19.77259. (Сокращенно это можно записать так: C/P (12 с) «—19.77259».)

Повторим вычисления с другими данными:  $a=64, b=65, c=1$ . Программа хранится в ПМК до его выключения.

64 X→ПА, 65 X→ПВ, 1 X→ПС B/O C/P «—2110.5».

### Структура программы

00—11: вычисление числителя дроби;  
 12—17: вычисление знаменателя;  
 18, 19: вывод из РY в РT числа 1/6, которое мешает разделить числитель на знаменатель;  
 20 : вычисление частного;  
 21 : завершение вычисления — прерывается выполнение программы и выводится результат на индикатор.

Может показаться, что составление и отладка программы, запись ее в ПМК сводит на нет выигрыш во времени решения по сравнению с непосредственным счетом в режиме вычислений. В действительности это не так. На ввод программы нужно затратить время и усилия лишь раз, при решении каждого последующего примера получается выигрыш во времени. Чем больше примеров решается, тем ощутимее преимущества, которые дает программа.

### 1.7. СОСТАВНЫЕ ОПЕРАТОРЫ

Многие программы, которые приходится составлять для решения практических задач, не удается строить в виде линейной последовательности команд. Часто необходимо, чтобы в зависимости от промежуточных результатов расчет шел по тем или иным формулам. Для составления подобных программ используются составные операторы: безусловных и условных переходов, переходов на подпрограммы, организации циклов.

**Операторы безусловных переходов.** С помощью этих операторов управляют содержимым счетчика адресов. Безусловный переход на шаг с адресом  $N$  в программе вводится оператором БП  $N$ . Оператор выполняется командой, которая занимает два адреса в программной памяти. Оператор прерывает линейную последовательность выполнения команд.

Пример. Адрес Команда

25	БП
26	47
...	...
47	...

После выполнения команды 24 осуществляется безусловный переход на адрес 47. Команды 27—46 будут пропущены. Команда КБПМ занимает один шаг. При ее выполнении происходит безусловный косвенный переход по адресу, записанному в регистр  $M$ .

В УАЯ оператор БП отсутствует. Дело в том, что использование этого оператора нарушает естественный порядок исполнения алгоритма делает его труднообозримым. Исключение БП избавляет в УАЯ от необходимости нумерации команд. Такой подход характерен для получивших большое распространение языков структурного программирования, например Паскаль, Ада.

**Операторы условных переходов.** Это операторы, осуществляющие переход к адресу  $M$  в зависимости от результата сравнения содержимого РХ с нулем. Проверяется выполнение одного из условий:  $x < 0$ ,  $x = 0$ ,  $x \neq 0$ ,  $x \geq 0$ . Переходы вводятся операторами  $\text{FX} < 0$ ,  $\text{FX} = 0$ ,  $\text{FX} \neq 0$ ,  $\text{FX} \geq 0$ , которые требуют также ввода адреса перехода  $M$ . Если условие выполнено, то адрес перехода  $M$  игнорируется и выполняется последующая часть программы, в противном случае выполняется переход на шаг с адресом  $M$ .

Косвенный условный переход — это одношаговая команда  $\text{KX} < 0M$ ,  $\text{KX} = 0M$ ,  $\text{KX} \neq 0M$ ,  $\text{KX} \geq 0M$ , где  $M$  — адрес регистра, вызывает условный переход по адресу, который хранится в этом регистре.

В УАЯ соответствующая команда ветвления имеет две формы: полную и короткую

Полная форма	Короткая форма
если $p$	если $p$
то $s_1$	то $s$
иначе $s_1$	
все	все

$p$  — это произвольное условие, в ПМК оно ограничено соотношениями равенства-неравенства;  $s_1$ ,  $s_2$ ,  $s$  — серии команд. В полной форме, если  $p$  верно, то выполняется  $s_1$ , пропускается  $s_2$ ; иначе выполняется  $s_2$ , пропускается  $s_1$ . В короткой форме, если  $p$  верно, то выполняется  $s$ , в противном случае пропускается.

Очевидно, что условный оператор ПМК срабатывает как команда ветвления в УАЯ в короткой форме. Действительно, если условие верно, то пропускается следующая за условием команда. В противном случае совершается переход к адресуемой команде.

**Оператор обращения к подпрограмме.** Повторяющиеся серии команд целесообразно организовывать в виде подпрограмм (ПП). Подпрограмма начинается с некоторого адреса  $N$  и содержит оператор возврата из подпрограммы В/О. Обращение к подпрограмме задается двухшаговой командой ПП  $N$  (скобка, как и ранее, обозначает единую команду). Ту же задачу решает

одношаговая команда косвенного обращения к подпрограмме КППМ, где адрес  $N$  первой команды подпрограммы является содержимым регистра  $M$  адресуемой памяти.

**Организация циклов.** Алгоритмы бывают линейными, с ветвлением и циклические. Для описания в программах алгоритмов первых двух типов используют операторы присваивания, безусловного и условного переходов. Операторы безусловного и условного перехода можно применять и для организации циклов.

Существует, однако, специальный способ построения циклов в программе с помощью счетчиков — аналогов команд повторения с параметром в УАЯ. Для построения циклов используются операторы FL0, FL1, FL2, FL3. С их помощью организуются счетчики в регистрах P0—P3. В программе после этих операторов указывается адрес перехода  $N$ , т. е. речь идет о двухшаговых командах. Рассмотрим для примера FL0 или еще конкретнее FL0  $N$ . При обращении к оператору FL0 (к остальным аналогично) из содержимого нулевого регистра-счетчика вычитается единица, оно модифицируется как при косвенном обращении к регистру P0.

Если разность не равна нулю, то переход осуществляется по адресу  $N$ , в противном случае оператор  $N$  игнорируется, работа по программе продолжается.

В УАЯ команда цикла имеет вид

**пока** (некоторое условие)

**иц**

*s*

**кц**

либо в случае цикла с параметром  $i$

**для**  $i$  **от**  $A$  **до**  $B$  **шаг**  $C$

**иц**

*s*

**кц**

Серия  $s$  (тело цикла) повторяется, пока выполняется условие, а в случае цикла с параметром, пока  $i$  меньше или равно  $B$ , при  $C > 0$ . Известно, что в структурных языках, на которые ориентирован УАЯ, обычно рассматриваются два типа циклов: цикл типа **пока** и **до**. В первом случае условие проверяется перед каждым входом в тело цикла. Только если оно выполнено, управление перейдет к телу цикла, иначе оно будет передано на следующий блок. После выполнения тела цикла управление автоматически переходит к началу цикла. Во втором случае условие проверяется на выходе из цикла, возврат к началу обусловлен выполнением условия.

В ПМК циклом управляет оператор  $\text{FL } R$  ( $R=0, 1, 2, 3$ ). После каждого исполнения цикла проверяется содержимое соответствующего регистра-счетчика. Управление выходит за пределы цикла, если в момент очередной проверки в счетчике записана 1.

В противном случае управление переходит к началу цикла, а содержимое Р0 заменяется целым числом, на единицу меньшим текущего, — модифицируется.

## **1.8. ПРОГРАММИРУЕМЫЙ МИКРОКАЛЬКУЛЯТОР И УЧЕБНО-АЛГОРИТМИЧЕСКИЙ ЯЗЫК**

Соответствие между командами в алгоритме для УАЯ и в программе для ПМК удобнее всего рассматривать на конкретном примере. Рассмотрим решение квадратного уравнения.

Решим квадратное уравнение  $x^2 - 3x + 2 = 0$  по программе  
 $1 \rightarrow \text{ПА} 3 / - / \quad x \rightarrow \text{ПВ} 2 \quad X \rightarrow \text{ПС} \quad B/O \quad C/P$

После останова считываем значения корней из Р1 и Р2:

Для решения другого квадратного уравнения  $x^2 - x - 6 = 0$  достаточно, сохранив программу, внести новые коэффициенты:

стачто, сохранив программу, внести новые коэффициенты.

Установим соответствия между УАЯ и программами. Заданию в УАЯ аргументов (*арг*) в программе для ПМК соответствует ввод коэффициентов квадратного уравнения в регистры А, В, С. Перечислению в рез переменных  $x_1$  и  $x_2$ , которым будут присвоены вычисленные значения корней, — резервирование регистров 1 и 2 для хранения этих значений.

Литерной переменной А в языке ПМК эквивалента нет. Ее значению «нет решения» на ПМК соответствует индикация сигнала ЕГГОГ. Отсутствие этого сигнала отвечает значению А = «есть решение». Это позволяет исключить в программе условный переход при ветвлении алгоритма: сигнал ЕГГОГ прекращает решение в случае отсутствия корней, как оператор С/П при их наличии. По этой же причине нецелесообразно запоминать дискриминант  $D$ . Удобнее хранить значение  $\sqrt{D}$ , так можно избавиться от лишнего вычисления корня при расчете  $x_2$ .

В целом три серии команд программы 00—08, 09—16, 17—26 соответствуют командам УАЯ:

$$D := b \wedge 2 - 4 * a * c; \quad x_1 := (-b + \sqrt{D}) / 2a; \quad x_2 := (-b - \sqrt{D}) / 2a.$$

Алгоритм на УАЯ	Программа для ПМК	Пошаговая проверка (ПП)
<p><b>алг КВАДРАТНОЕ УРАВНЕНИЕ</b>      веш <math>a, b, c, x_1, x_2</math>, лит А      арг <math>a, b, c</math>      рез <math>x_1, x_2, A</math></p>	<p>Распределение памяти  <math>a \rightarrow \text{ПА}</math> <math>b \rightarrow \text{ПВ}</math>  <math>c \rightarrow \text{ПС}</math> <math>x_1</math> и <math>x_2</math> в Р1      и Р2; ЕГГОГ соответ-</p>	$x - 3x + 2 = 0$ $a = 1; b = -3;$ $c = 2$ Ввод программы и данных

Алгоритм на УАЯ	Программа для ПМК	Пошаговая проверка (ПП)																																																																																																																																							
<p><b>нач вещ D</b>  <math>D := b \wedge 2 - 4 * a * c</math></p> <p><b>если</b> <math>D &lt; 0</math>  <b>то</b> <math>A := \text{«нет решения»}</math>  <b>иначе</b> <math>A := \text{«есть решение»}</math></p> $x_1 := (-b + \sqrt{D}) / 2/a$ $x_2 := (-b - \sqrt{D}) / 2/a$ <p><b>всё</b>  <b>кон</b></p>	<p>стывает значению A, отсутствие ЕГГОГ — решение есть</p> <p>Выделение РД для хранения <math>\sqrt{D}</math></p> <table> <tbody> <tr><td>00</td><td><math>\Pi \rightarrow XB</math></td><td>6L</td><td>-3</td><td>ПП</td></tr> <tr><td>01</td><td><math>FX^2</math></td><td>22</td><td>9</td><td>ПП</td></tr> <tr><td>02</td><td>4</td><td>04</td><td>4</td><td>ПП</td></tr> <tr><td>03</td><td><math>\Pi \rightarrow XA</math></td><td>6—</td><td>1</td><td>ПП</td></tr> <tr><td>04</td><td>X</td><td>12</td><td>4</td><td>ПП</td></tr> <tr><td>05</td><td><math>\Pi \rightarrow XC</math></td><td>6C</td><td>2</td><td>ПП</td></tr> <tr><td>06</td><td>X</td><td>12</td><td>8</td><td>ПП</td></tr> <tr><td>07</td><td>—</td><td>11</td><td>1</td><td>ПП</td></tr> <tr><td>08</td><td>F V</td><td>21</td><td>1</td><td>ПП</td></tr> </tbody> </table> <p>Индикация ЕГГОГ</p> <table> <tbody> <tr><td>09</td><td><math>X \rightarrow \Pi D</math></td><td>4Г</td><td>1</td><td>ПП</td></tr> <tr><td>10</td><td><math>\Pi \rightarrow XB</math></td><td>6L</td><td>-3</td><td>ПП</td></tr> <tr><td>11</td><td>—</td><td>11</td><td>4</td><td>ПП</td></tr> <tr><td>12</td><td>2</td><td>02</td><td>2</td><td>ПП</td></tr> <tr><td>13</td><td>:</td><td>13</td><td>2</td><td>ПП</td></tr> <tr><td>14</td><td><math>\Pi \rightarrow XA</math></td><td>6—</td><td>1</td><td>ПП</td></tr> <tr><td>15</td><td>:</td><td>13</td><td>2</td><td>ПП</td></tr> <tr><td>16</td><td><math>X \rightarrow \Pi 1</math></td><td>41</td><td>2</td><td>ПП</td></tr> <tr><td>17</td><td><math>\Pi \rightarrow XD</math></td><td>6Г</td><td>1</td><td>ПП</td></tr> <tr><td>18</td><td><math>\Pi \rightarrow XB</math></td><td>6L</td><td>-3</td><td>ПП</td></tr> <tr><td>19</td><td>+</td><td>10</td><td>-2</td><td>ПП</td></tr> <tr><td>20</td><td>/—/</td><td>0L</td><td>2</td><td>ПП</td></tr> <tr><td>21</td><td>2</td><td>02</td><td>2</td><td>ПП</td></tr> <tr><td>22</td><td>:</td><td>13</td><td>1</td><td>ПП</td></tr> <tr><td>23</td><td><math>\Pi \rightarrow XA</math></td><td>6—</td><td>1</td><td>ПП</td></tr> <tr><td>24</td><td>:</td><td>13</td><td>1</td><td>ПП</td></tr> <tr><td>25</td><td><math>X \rightarrow \Pi 2</math></td><td>42</td><td>1</td><td>ПП</td></tr> <tr><td>26</td><td>C/P</td><td>50</td><td>«1»</td><td></td></tr> </tbody> </table>	00	$\Pi \rightarrow XB$	6L	-3	ПП	01	$FX^2$	22	9	ПП	02	4	04	4	ПП	03	$\Pi \rightarrow XA$	6—	1	ПП	04	X	12	4	ПП	05	$\Pi \rightarrow XC$	6C	2	ПП	06	X	12	8	ПП	07	—	11	1	ПП	08	F V	21	1	ПП	09	$X \rightarrow \Pi D$	4Г	1	ПП	10	$\Pi \rightarrow XB$	6L	-3	ПП	11	—	11	4	ПП	12	2	02	2	ПП	13	:	13	2	ПП	14	$\Pi \rightarrow XA$	6—	1	ПП	15	:	13	2	ПП	16	$X \rightarrow \Pi 1$	41	2	ПП	17	$\Pi \rightarrow XD$	6Г	1	ПП	18	$\Pi \rightarrow XB$	6L	-3	ПП	19	+	10	-2	ПП	20	/—/	0L	2	ПП	21	2	02	2	ПП	22	:	13	1	ПП	23	$\Pi \rightarrow XA$	6—	1	ПП	24	:	13	1	ПП	25	$X \rightarrow \Pi 2$	42	1	ПП	26	C/P	50	«1»		
00	$\Pi \rightarrow XB$	6L	-3	ПП																																																																																																																																					
01	$FX^2$	22	9	ПП																																																																																																																																					
02	4	04	4	ПП																																																																																																																																					
03	$\Pi \rightarrow XA$	6—	1	ПП																																																																																																																																					
04	X	12	4	ПП																																																																																																																																					
05	$\Pi \rightarrow XC$	6C	2	ПП																																																																																																																																					
06	X	12	8	ПП																																																																																																																																					
07	—	11	1	ПП																																																																																																																																					
08	F V	21	1	ПП																																																																																																																																					
09	$X \rightarrow \Pi D$	4Г	1	ПП																																																																																																																																					
10	$\Pi \rightarrow XB$	6L	-3	ПП																																																																																																																																					
11	—	11	4	ПП																																																																																																																																					
12	2	02	2	ПП																																																																																																																																					
13	:	13	2	ПП																																																																																																																																					
14	$\Pi \rightarrow XA$	6—	1	ПП																																																																																																																																					
15	:	13	2	ПП																																																																																																																																					
16	$X \rightarrow \Pi 1$	41	2	ПП																																																																																																																																					
17	$\Pi \rightarrow XD$	6Г	1	ПП																																																																																																																																					
18	$\Pi \rightarrow XB$	6L	-3	ПП																																																																																																																																					
19	+	10	-2	ПП																																																																																																																																					
20	/—/	0L	2	ПП																																																																																																																																					
21	2	02	2	ПП																																																																																																																																					
22	:	13	1	ПП																																																																																																																																					
23	$\Pi \rightarrow XA$	6—	1	ПП																																																																																																																																					
24	:	13	1	ПП																																																																																																																																					
25	$X \rightarrow \Pi 2$	42	1	ПП																																																																																																																																					
26	C/P	50	«1»																																																																																																																																						

алг СУММА КВАДРАТОВ НАТУРАЛЬНЫХ ЧИСЕЛ (нат  $n$ , цел  $s$ )

## Программа с пока-циклом

арг  
рез  
нач нач

*i* :=

$s :=$

пока  $i \leq$

$$n = P0, \quad i = P1, \\ s = PC, \quad n \text{ B/O} \quad C/\Pi$$

00	$X \rightarrow \Pi 0$	40
01	I	01
02	$X \rightarrow \Pi 1$	41
03	CX	0Γ
04	$X \rightarrow \Pi C$	4C
05	$\Pi \rightarrow X 0$	60

нц

s := s + i  $\wedge$  2

i := i + 1

кц

кон

Вариант, адаптированный к ПМК:

<i>n</i>	В/О	С/П	00	X → P0	40	03	FX <sup>2</sup>	22	06	02	02
01	CX		01	0G	04		+	10	07	C/P	50
02	P → X0		02	60	05		FL0	5Г			

На ПМК можно также реализовать цикл «пока», характерный для УАЯ. Его заголовок строится с помощью оператора условного перехода, роль команды «Конец цикла» выполняет оператор безусловного перехода. Рассмотрим алгоритм на УАЯ и соответствующую ему программу «Сумма квадратов натуральных чисел».

Программа с несвойственным ПМК пока-циклом далека от оптимальной. Укрупненные действия УАЯ приходится заменять сериями команд ПМК. Чтобы соединить в единое целое эквиваленты структур развитого языка, понадобились дополнительные команды. Переменным алгоритма в программе соответствуют адресуемые регистры памяти и стека: *n* — Р0, *i* — Р1, *s* — РС. Команда присваивания переводится как запись числа в соответствующий регистр (команды 01—04, 14, 18) или вызов в РХ некоторого числа (05, 06, 10, 11, 15, 21). Команде ветвления соответствует сравнение содержимого РХ с нулем (08), команде кц (Конец цикла) — оператор БП с адресом перехода (19, 20).

Есть и еще одно отличие. В УАЯ значения аргумента считаются заведомо введенными в вычислительное устройство. Остается только выполнить расчет. Для ПМК необходимо дополнить алгоритм и программу специальными средствами ввода данных и вывода результата. Такие средства в программе — это особенность ввода: *n* В/О С/П и команда X → P0, а также команда по адресу 21 для индикации результата.

Стандартный набор команд УАЯ оказывается недостаточным для того, чтобы организовать диалог с человеком, особенно в игровых задачах, где действия противника зависят от промежу-

06	P → XI	61
07	—	11
08	FX ≥ 0	59
09	21	21
10	P → XC	6C
11	P → XI	61
12	FX <sup>2</sup>	22
13	+	10
14	X → PC	4C
15	P → XI	61
16	1	01
17	+	10
18	X → PI	41
19	БП	51
20	05	05
21	P → XC	66
22	C/P	50

точных ходов ЭВМ и наоборот. В связи с этим в дальнейшем используются две дополнительные команды: **ВВОД** и **ВЫВ** (вывод).

## ЗАДАНИЯ

Составьте программы для решения следующих задач.

1. Решение квадратного уравнения  $ax^2+bx+c=0$ . Для  $D < 0$  ПМК должен выдавать сигнал ЕГГОГ, случаи  $D=0$  и  $D > 0$  целесообразно рассмотреть совместно, вычисляя корни по общей формуле.

2. Вычисление суммы кубов *n* чисел в двух вариантах:

2.1. Числа вводятся в ПМК одно за другим с помощью С/П и по мере ввода возводятся в куб.

2.2. Все числа предварительно вводятся в регистры адресуемой памяти (*n* < 12, Р0—Р2 используются для накопления суммы и организации цикла).

Есть несколько способов возвведения в куб числа, находящегося в РХ:

а) с помощью адресуемой памяти, например X → РА FX<sup>2</sup> P → RA X

б) только с помощью стека: B↑ FX<sup>2</sup> X

в) с помощью оператора BX (Восстановление X): FX<sup>2</sup> FBX X. Не так удобно использовать функцию XУ: B↑ 3 ↔ FXУ. Объясните, почему.

3. Суммирование четвертых степеней.

4. Суммирование обратных величин чисел.

5. Вычисление сопротивления электрической цепи при параллельном соединении, если известны  $R_1, R_2, \dots, R_n$ . Расчетная формула:  $R = 1 / \sum_{i=1}^n 1/R_i$ .

6. Расчет среднего арифметического  $(a+b)/2$ , геометрического  $\sqrt{ab}$  и гармонического  $1/(1/a+1/b)$ . Сравните значения для различных числовых данных.

7. Вычисление суммы членов арифметической прогрессии  $s = (2a_1 + (n-1)d)n/2$ , суммы членов геометрической прогрессии  $s = a_1(q^n - 1)/(q - 1)$ , площади треугольника, заданного двумя сторонами и углом между ними  $S = ab \sin c/2$ .

8. Вычисление диагонали прямоугольного параллелепипеда  $d = \sqrt{a^2 + b^2 + c^2}$ . Считая *a*, *b*, *c* приближенными числами (абсолютные погрешности соответственно  $\Delta_a, \Delta_b, \Delta_c$ ), вычислите в программе также  $\Delta_d$ . Для конкретных значений *a*, *b*, *c*,  $\Delta_a, \Delta_b, \Delta_c$  найдите *d* и  $\Delta_d$ . С помощью  $\Delta_d$  округлите результат. Расчетная формула:  $\Delta_d = (a\Delta_a + b\Delta_b + c\Delta_c)/\sqrt{a^2 + b^2 + c^2}$ .

9. Нахождение наибольшего из *n* чисел, вводимых по очереди с помощью С/П. Проверьте программу на числах:  $22\pi/15, 5e/3, \sqrt{31}/14, 25 \sin^2 25^\circ, 3 \lg 24 + \ln 14, 88/19$ . Ввод данных:

22 B↑ 15 : Fπ X  
1 Fe<sup>x</sup> 5 X 3 : FV  
31 B↑ 1 . 4 : FV

25 F sin FX<sup>2</sup> 25 X  
Переключатель углов в положении Г.  
24 Flg 3 X 1 . 4 Fln +  
88 B↑ 19 :

10. Для составления таблиц значений функций:

10.1.  $y = 8\sqrt{x} + 3 \cos^2 x$ ;

10.2.  $y = x^3 + 3x^2 + 7$  на отрезке  $[0, 2]$  с длиной шага  $h = 0,2$  — всего 11 значений. По полученным данным построить на миллиметровой бумаге графики функций.

Организуйте подпрограмму вычисления функции для фиксированного значения *x*. При планировании памяти нужно предусмотреть регистры для хранения *h* и вычисленных текущих значений *x* и *y*, которые индицируются.

11. Вычисление площади боковой поверхности усеченного конуса, если диагональ осевого сечения *d* и образующая *a* составляют с плоскостью основания угол *a*. Расчетная формула:  $S = \pi a \sqrt{d^2 - a^2 \sin^2 a}$ ; контрольный пример:  $d = 3, a = 2, a = 30^\circ$ , переключатель в положении Г. Результат: 17,771531. Вычислите площадь поверхности при различных значениях аргументов.

## 12. Вычисление функции

$$y = \begin{cases} \pi/(3x), & \text{если } x < -1, \\ \arcsin x, & \text{если } -1 \leq x \leq 1, \\ \pi x/3, & \text{если } x > 1 \end{cases}$$

Контрольный пример:  $x=3$ , переключатель в положении Р. Результат: 3,1415926.  
Найдите по программе значения  $y$  для  $x=0.5$  ( $y=0,52359878$ ),  $x=-1$  ( $y=-1,5707963$ ). Постройте таблицу значений функции, рассчитанных с постоянным шагом.

## 13. Вычисление функции

$$z = \begin{cases} 3t-2, & \text{если } t < 0, \\ 2 \sin t, & \text{если } t \geq 0; \end{cases} \quad t = \begin{cases} \lg(-x) & \text{при } x < 0, \\ \sqrt[4]{x} & \text{при } x \geq 0. \end{cases}$$

Контрольные примеры: для  $x=-1$  получается 0, для  $x=16$   $y=1,8185949$ .

14. Вычисление площади равностороннего треугольника. Стороны треугольника  $a, b, c$ . Если треугольник не равносторонний, ПМК будет выдавать нуль. Контрольный пример: при  $a=b=c=6$   $S=15,588457$ , при  $a=3, b=4, c=3$   $S=0$ .

15. Вычисление суммы положительных и отрицательных чисел, которые в произвольном порядке вводятся в ПМК. Суммы положительных чисел хранятся в РС, отрицательных в РД. Контрольный пример: ввод  $-1, -0,5, 6, 7, -4, 2, 15$ ; результаты: П→ХС «30»; П→ХД «-5,5». Испытайте программу на различных последовательностях данных.

16. Решение задачи на распознавание: индицируется 1, если выполнены требуемые условия, и 0 в противном случае.

16.1.  $A(x, y)$  принадлежит биссектрисе II, IV координатных углов и находится внутри круга радиуса  $R$  с центром в начале координат. Контрольные примеры: при  $x=-2, y=2, R=3$  индицируется 1, при  $x=3, y=4, R=4,5$  — «0».

16.2. Прямая  $y=mx+n$  пересекает параболу  $y=ax^2+bx+c$  в двух различных точках. Контрольные примеры: при  $a=1, b=2, c=3, m=4, n=2$  получаем 0, при  $a=1, b=2, c=2, m=5, n=1$  — «1».

16.3. Натуральное число  $N$  кратно трем и трехзначно. Контрольные примеры: при  $N=243$  «1»,  $N=353$  «0»,  $N=27$  «0». Условие кратности используемого числа тройке:  $\sin(N \cdot 60^\circ) = 0$ . Переключатель углов установлен в положение Г. В общем виде условие кратности числа  $N$  числу  $M$ :  $\sin(N \cdot 180^\circ/M) = 0$ . Отметим, что в данном случае нельзя пользоваться радианной мерой, т. е. вычислять  $\sin(\pi N/M)$ : ответ будет неверным из-за приближенного значения  $\pi$ .

16.4.  $a$  и  $b$  — натуральные числа. Остаток от деления  $a$  на  $b$  равен  $c$  или  $d$ . Контрольные примеры: при  $a=28, b=6, c=5, d=4$  «1»; при  $a=37, b=12, c=2, d=3$  «0». Остаток от деления  $a$  на  $b$  подсчитывается по формуле  $r = a - b[a/b]$ , где  $[a/b]$  — целая часть от деления  $a$  на  $b$ . Для вычисления  $[x]$  в языках высокого уровня, например Бейсике, используют функцию INT(X). В ПМК «Электроника МК-61» и «Электроника МК-52» имеется аналогичная функция  $y=[x]$ . В «Электронике МК-56» при вычислении  $[a]$  для  $a \geq 0$  можно воспользоваться программой:

00. В↑ 01.1 02+ 03.Х→П3 04.КП→Х3 05.П→Х3 06.С/П

Ввод:  $a$  В/О С/П "[a]".

Команда КП→Х3 выделяет из содержимого РЗ целую часть, если это содержимое больше или равно единице. Для исключения нуля нам пришлось заранее увеличить число на единицу и воспользоваться косвенным обращением к регистру, в котором при модификации вычитается единица. Этой процедурой часто пользуются в программах (см. гл. 2).

17. Вычисление периода  $T$  обращения спутника вокруг Земли, если радиус Земли  $R=6370 \cdot 10^3$  м, высота полета  $h=276 \cdot 10^3$  м,  $g=9,8$  м/с<sup>2</sup>. Расчетная формула:  $T=2\pi(R+h)\sqrt{R+h}/(R\sqrt{g})$ . Ответ:  $5,4 \cdot 10^3$  с. Постройте с помощью программы зависимость периода обращения спутника от высоты полета в виде таблицы с постоянным шагом.

18. Спутник движется вокруг Земли по круговой орбите радиуса  $r$ . После срабатывания тормозного двигателя скорость спутника уменьшается и он переходит на эллиптическую орбиту, касающуюся Земли. Составьте программу, вычисляющую, через какое время после торможения спутник приземлится, если радиус Земли  $R$ . Если пренебречь сопротивлением атмосферы, то расчетная формула  $t=\pi((R+r)/2)^{3/2}/(R\sqrt{g})$ . Контрольный пример:  $R=6370 \cdot 10^3$  м,  $r=6500 \cdot 10^3$  м,  $g=9,81$  м/с<sup>2</sup>. Ответ: 43 мин.

19.1. Автомобиль массой  $m$  (кг) движется в гору с ускорением  $a$  (м/с<sup>2</sup>). Вычислите силу тяги, если угол равен  $\sin a$  и коэффициент сопротивления  $\mu$ . Расчетная формула:  $F=m(g(\sin a + \mu \cos a) + a)$ . Контрольный пример:  $m=4000$  кг,  $a=0,2$  м/с<sup>2</sup>,  $\sin a=0,04$ ,  $\mu=0,04$ ,  $g=9,8$  м/с<sup>2</sup>. Ответ:  $F=3200$  Н.

Рассчитайте силу тяги разных типов автомобилей. Масса автомобилей при полной загрузке: «Запорожец ЗАЗ-965» — 980 кг, «Москвич-407» — 1280 кг, «Волга» — 1885 кг, «Чайка» — 2525 кг.

19.2. Бруск массой  $m$  движется по горизонтальной плоскости с ускорением  $a$  под действием силы, приложенной к нему под углом  $a$ . Коэффициент трения  $k$ . Найдите силу натяжения нити  $T$  и силу давления  $N$  бруска на поверхность.

Решите по программе задачу при следующих данных:  $m=350$  г,  $a=35^\circ$ ,  $a=0,4$  м/с<sup>2</sup>,  $k=0,12$ ,  $g=9,8$  м/с<sup>2</sup>. Расчетные формулы:  $T=m(a+kg)/(\cos a+k \sin a)$ ,  $N=m(g \cos a - a \sin a)/(\cos a+k \sin a)$ . Ответ:  $T=0,621$  Н,  $N=3,07$  Н.

20. Вычисление наибольшего общего делителя (НОД) двух натуральных чисел. Используйте при этом только вычитание и сравнение чисел на большем-меньше-равно. Алгоритм решения задачи следующий:

1. Если числа равны, то любое из них есть искомое. Конец.

2. Вычитаем из большего числа меньшее.

3. Заменяем большее число полученной разностью, сохраняем меньшее число.

4. Возвращаемся к указанию 1.

21. Составьте программу, позволяющую сравнивать два натуральных числа (больше, равно, меньше), если допустимыми являются действия вычитания двух чисел  $a$  и  $b$  и проверки их равенства. Алгоритм решения задачи следующий:

1. Вычитаем из  $a$  единицу.

2. Если  $a-1=b$ , то  $a>b$ . Конец.

3. Если  $a-1=0$ , то  $a \leq b$ . Конец.

4. Заменяем значение  $a$  на  $a-1$ .

5. Возврат к указанию 1.

Программу для задачи 21 нужно использовать как подпрограмму для выполнения указания 2 алгоритма.

22. Построить программу вычисления остатка от деления  $m$  на  $n$ . Допустимы действия вычитания и проверки равенства двух чисел. Алгоритм решения задачи следующий:

1. Если  $m=n$ , то остаток равен 0. Конец.

2. Если  $m < n$ , то  $m$  есть остаток. Конец.

3. Заменяем  $m$  разностью  $m-n$ .

4. Возврат к указанию 1.

Задачи 20—22 с ограничением помогают выявить максимальные возможности отдельных операторов ПМК. При этом не требуется использовать оптимальный алгоритм решения. Так, вычисление наибольшего общего делителя на ПМК целесообразней программировать последовательным делением.

23. Забытое тождество. Забыть вовсе не означает ничего не помнить, чаще мы просто сомневаемся: один или два, плюс или минус, квадрат или куб и т. д. В таких случаях мы обычно подставляем допустимые значения переменной в обе части предполагаемого равенства. Если результаты не совпадут, то сомнений нет — тождество восстановлено неверно. Этот экспериментальный метод неполной индукции возможен при одном условии: вычисления доступны человеку и не требуют большого труда и времени. Большую услугу здесь мо-

жет оказать микрокалькулятор. В следующих примерах проверьте тождества, запрограммировав их. Для проверки подставляйте несколько различных значений аргументов.

- 23.1.  $(a+b)^2 = a^2 + 2ab + b^2$ .
- 23.2.  $(a-b)^3 = a^3 - 3a^2b + 3ab^2 - b^3$ .
- 23.3.  $(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$ .
- 23.4.  $a^5 - b^5 = (a-b)(a^4 + a^3b + a^2b^2 + ab^3 + b^4)$ .
- 23.5.  $\lg a / \lg b = \ln a / \ln b (a, b > 0, b \neq 1)$ .
- 23.6.  $a = 10^{\lg a} (a > 0)$ .
- 23.7.  $\lg(ab) = \lg a + \lg b (a, b > 0)$ .
- 23.8.  $\cos^2 a = 1 - \sin^2 a$  (переключатель Р—Г в любой позиции).
- 23.9.  $\operatorname{tg} a = \sin a / \cos a (a \neq \pi/2 + k\pi)$ .
- 23.10.  $\sqrt{1-x^2} = \sin(\arccos x) (|x| \leq 1)$ .
- 23.11.  $1^2 + 2^2 + \dots + n^2 = n(n+1)(2n+1)/6$ .
- 23.12.  $1^3 + 2^3 + \dots + n^3 = (n(n+1)/2)^2$ .

Расхождения в результатах, полученные в левых и правых частях равенств, должны относиться только к последнему десятичному знаку и возникают из-за приближенных вычислений.

При возведении в степень, чтобы исключить накопление ошибок, лучше не пользоваться оператором  $X^y$ . Так, если аргумент находится в РХ, то процедура возведения в куб выглядит так:  $B \uparrow FX^2 X$ , в четвертую степень:  $FX^2 FX^2$ , в пятую степень:  $B \uparrow FX^2 FX^2 X$  и т. д.

## ОТВЕТЫ

9. Программа для ПМК и алгоритм выбора большего из  $n$  чисел.

алг БИН (нат $n$ , веществ MAX, веществ	00	КНОП	54
таб A[1 : n])	01	$B \uparrow$	0E
арг $n$ , А	02	C/P	50
рез MAX	03	—	11
нач нат $i$	04	FX < 0	5C
$i = 2$ ; MAX := A[1]	05	08	08
	06	FBX	0
пока $i \leq n$	07	B/O	52
нц если MAX < A[i]	08	↔	14
то MAX := A[i]	09	B/O	52
все			
$i := i + 1$			
кц			
кон			

Инструкция. Ввести первое число, В/О С/П, далее очередное число, С/П, ...

Контрольные примеры. 22 Fπ × 15 : В/О С/П «4.6076691»; 1 Fex 5 × 3 : С/П «4.6076691»; 31 B↑ 1.4 : F π С/П «4.7056197» и т. д. После последнего ввода индицируется 4.7056197, поскольку это наибольшее число.

Поясним решение задачи. Наибольшее из рассмотренных чисел находится в РУ, очередное число — в РХ. Вычисляем их разность: если она отрицательна, то очередное число — большее, оно пересыпается в РУ и индицируется, в противном случае индицируется прежнее содержимое РУ.

Рассмотрим, как этот алгоритм реализован в программе. Прежде всего отметим необычную роль оператора В/О. Поскольку здесь нет подпрограммы, В/О заменяет двухшаговую команду БП 01. Так организован цикл, начинающийся с команды 01. Вместо команды по адресу 00 записан оператор КНОП, который не влияет на работу программы. В качестве исходного наибольшего числа программа выбирает первое число, оно присваивается РУ и индицируется (команды 01, 02). После ввода в РХ очередного числа программа запускается на продолжение счета нажатием клавиши С/П; полученная разность сравни-

вается с нулем (команды 03, 04). Если условие выполнено, то вычитаемое больше, вызывается из РХ1 (FBX) и индицируется (команды 06, 07, 01, 02). В противном случае на индикатор выводится предыдущее содержимое РУ и т. д. (команды 05, 08, 09, 01, 02). В любом случае цикл ввода и обработки данных можно повторить, пока не исчерпан список чисел.

Нетрудно заметить насыщенность команд УАЯ по сравнению с командами программы. Алгоритм на УАЯ близок к естественному человеческому языку: если ... то ... иначе ... В алгоритме таблица данных предполагается заранее введенной в память исполнительного устройства. В программе числа вводятся поэлементно, в процессе решения задачи. Это результат ограниченной памяти ПМК. В первом случае человек не участвует в процессе решения, во втором его присутствие необходимо. В режиме диалога человек как бы компенсирует малые возможности ПМК. Роль человека в диалоге пассивна, за ним остается только ввод данных. Только в случае небольших массивов (например, шесть элементов, как в примере) удается построить для ПМК программу, когда все числа заранее вводятся в память. В этом случае участие человека в решении не требуется, программа приближается к алгоритму на УАЯ.

### 10. Распределим адресуемую память:

P0 — счетчик, фиксирующий, сколько раз вычислялись пары  $x$  и  $y$ , начальное значение 11;

PX — текущее значение  $x$ , начальное 0;

PY — текущее значение  $y$ ;

RD — константа  $h=0.2$ .

### 10.2. Программа ТАБЛИЦА

00	CX	0Г	12	X → ГС	4C	24	C/P	50
01	0	00	13	Π → ХВ	6L	25	B↑	0E
02	.	0 —	14	C/P	50	26	FX <sup>2</sup>	22
03	2	02	15	Π → ХС	6C	27	×	12
04	X → ПД	4Г	16	C/P	50	28	FBX	0
05	1	01	17	Π → ХВ	6L	29	3	03
06	1	01	18	Π → ХД	6Г	30	×	12
07	X → ПО	40	19	+	10	31	+	10
08	CX	0Г	20	X → ПВ	4L	32	7	07
09	X → ПВ	4L	21	FL0	5Г	33	+	10
10	ПП	53	22	10	10	34	B/O	52
11	25	25	23	Fπ	20			

Инструкция. Ввод: В/О С/П. При останове на индикаторе высвечивается значение  $x_i$  ( $i=1, 2, \dots, n$ ). Еще одно нажатие С/П выводит на индикатор соответствующее значение  $y_i$ . Нажатие С/П, и цикл вычислений очередной пары значений  $x$  и  $y$  повторяется. После  $n$  повторений цикла ( $n=11$ ) индицируется π, что означает окончание вычислений.

Рассчитанные по программе результаты по мере индикации сводим в таблицу:

x	0	0,2	0,4	0,6	0,8	1,0	1,2	1,4	1,6...
y	7	7,128	7,544	8,296	9,432	11	13,048	15,624	18,776...

Команды 00—09 — формирование начальных данных. Команды, начиная с 25, образуют подпрограмму вычислений функции. Вместо команд 01—03 можно использовать команды 2 ВП 1 /—/, однако в этом случае потребуется лишний шаг программы.

## 16.1. Алгоритм и программа РАСПОЗНАВАНИЕ.

```

алг РАСПОЗНАВАНИЕ (вещ, x, y, R, лит, z).
арг x, y, R
рез z
нач если  $(y = -x) \wedge (x^2 + y^2 < R^2)$ 
    то z := «условие выполнено»
    иначе z := «условие не выполнено»
все
кон

```

В развитых языках программирования допустимо объединение нескольких условий под ключевым словом **если** ( $\wedge$  — знак конъюнкции или логического умножения). Отсутствие этих возможностей в языке ПМК делает программу громоздкой. Распределим память  $x \rightarrow \text{РА}$ ,  $y \rightarrow \text{РВ}$ ,  $R \rightarrow \text{РС}$ .

00	$X \rightarrow \text{ПС}$	4C	08	21	21	16	$\bar{}$	11
01	$F_O$	25	09	$\Pi \rightarrow \text{ХА}$	6	—	17	$FX < 0$
02	$X \rightarrow \text{ПВ}$	4L	10	$FX^2$	22	18	21	21
03	$F_O$	25	11	$\Pi \rightarrow \text{XB}$	2L	19	1	01
04	$X \rightarrow \text{ПА}$	4	—	$FX^2$	22	20	C/P	50
05	$\Pi \rightarrow \text{XB}$	6L	13	+	10	21	0	00
06	+	10	14	$\Pi \rightarrow \text{ХС}$	6C	22	C/P	50
07	$FX = 0$	5E	15	$FX^2$	22			

Инструкция. Данные вводятся в стек:  $x \uparrow B \uparrow y \uparrow R \uparrow B/O \uparrow C/P$ .

Контрольный пример. 2  $/-$   $B \uparrow$  2  $B \uparrow$  3  $B/O \uparrow C/P$  «1»

23.11. Ввод n B/O C/P

00	$X \rightarrow \text{П0}$	40	08	1	01	17	$FL_0$	5Г
01	I	01	09	+	10	18	14	14
02	+	10	10	$\times$	12	19	—	11
03	$\Pi \rightarrow \text{Х0}$	60	11	6	06	20	$FX = 0$	5E
04	$\times$	12	12	:	13	21	24	24
05	$FBX$	0	13	0	00	22	1	01
06	2	02	14	$\Pi \rightarrow \text{Х0}$	60	23	C/P	50
07	$\times$	12	15	$FX^2$	22	24	0	00
			16	+	10	25	C/P	50

По командам 00—12 вычисляется правая часть соотношения, по командам 13—18 — левая часть. Если они равны, индицируется 1, в противном случае 0.

## Глава 2

### МЕТОДЫ ПРОГРАММИРОВАНИЯ

#### 2.1. БАЗОВЫЕ ПРОГРАММЫ

Базовыми будем называть несложные программы, которые часто используются в составе других программ.

#### Б1. Модуль (абсолютная величина числа).

##### Программа МОДУЛЬ-1

00	$FX^2$	22
01	$F\sqrt{-}$	21
02	C/P	50

##### Программа МОДУЛЬ-2

00	$FX < 0$	5C
01	03	03
02	/—/	0L
03	C/P	50

Инструкция. Ввод:  $x \uparrow B/O \uparrow C/P$ .

Примеры. 25  $/-$   $B/O \uparrow C/P$  «25»; 7  $B/O \uparrow C/P$  «7»; 0  $B/O \uparrow C/P$  «0».

Б2. Распознавание целого четного числа, проверка делимости чисел. Программы основаны на том, что  $\sin x = 0$  тогда и только тогда, когда  $x = 180^\circ n$  ( $n = 0, \pm 1, \pm 2, \dots$ ). Нельзя, однако, использовать  $x = \pi n$ , так как ПМК оперирует округленным значением  $\pi$ . Может оказаться, что  $\sin \pi n \neq 0$ , например  $\sin 7\pi = 2 \cdot 10^{-7}$  (переключатель в положении Р, 7 F<sub>л</sub> × F<sub>sin</sub>). Для индикации результатов используют коды 1 и 0. Индикация 1 означает, что испытуемое число целое, или четное, или кратно другому, индикация 0 — свойство не выполняется.

##### Б2.1. Целое число x.

00	$B \uparrow$	0E	04	$\times$	12	08	1	01
01	1	01	05	$F\sin$	1C	09	C/P	50
02	8	08	06	$FX = 0$	5E	10	0	00
03	0	00	07	10	10	11	C/P	50

Инструкция. Ввод  $x \uparrow B/O \uparrow C/P$ . Переключатель в положении Г.

Примеры. 37.2  $B/O \uparrow C/P$ , индицируется 0, т. е. число не целое; 15  $/-$   $B/O \uparrow C/P$  «1» — целое число; 0  $B/O \uparrow C/P$  «1» — целое число.

Б2.2. Четное число x. Программа строится на основе программы Б2.1 — испытуемое число делится на два, далее проверяется, целое ли оно.

00	$B \uparrow$	0E	05	0	00	10	1	01
01	2	02	06	0	12	11	C/P	50
02	:	13	07	$F\sin$	1C	12	0	00
03	1	01	08	$FX = 0$	5E	13	C/P	50
04	8	08	09	12	12			

Инструкция. Ввод:  $x \uparrow B/O \uparrow C/P$ .

Примеры. 25  $B/O \uparrow C/P$  «0» — число нечетное; 8  $/-$   $B/O \uparrow C/P$  «1» — число четное; 0  $B/O \uparrow C/P$  «1» — число четное.

На примере этой программы нетрудно заметить, как, объединяя известные фрагменты программ, можно ускорить процесс составления необходимой программы. Заметны также и недостатки этого метода — программа, составленная механическим объединением, получалась неэффективной, ее нетрудно сократить, если команды 01—05 заменить двумя другими: 01. 9 09 02. 0 00.

Б2.3. Делимость x на y. Программа строится на использовании программы Б2.1 — число x делится на y, далее проверяется, целый ли результат.

00	13	04	$\times$	12	08	1	01
01	01	05	F sin	1C	09	C/P	50
02	8	08	06	FX = 0	5E	10	0 00
03	0	00	07		10	11	C/P 50

Инструкция. Ввод:  $x$  В $\uparrow$   $y$  В/O С/P. Переключатель в положении Г.

Примеры. 13 В $\uparrow$  2 В/O С/P «0» — значит 13 не кратно 2; 27 /—/ В $\uparrow$  9 В/O С/P «1» — число — 27 кратно 9.

**Б.3.** Отделение целой части  $[x]$ , дробной части  $\{x\}$ , остатка от деления. Программы основаны на использовании операторов косвенной адресации, которые отделяют целую часть содержимого соответствующего регистра. Содержимое РХ должно удовлетворять условию  $1 \leq x < 10^8$ . Если косвенное обращение осуществляется к регистрам 0—3, то их содержимое модифицируется — дробная часть отбрасывается, а из целой вычитается единица; в регистрах 4—6 единица прибавляется, в остальных регистрах целая часть не изменяется, а дробная также отбрасывается. Например, в Р7 хранится число 12,36. Оператор КП→X7 оставляет в Р7 число 12. Чтобы в этом убедиться, вызовите содержимое регистра на индикатор: П→X7 «00000012» (как известно, нули слева от числа являются незначащими). Несколько сложнее обстоит дело, если  $0 \leq x < 1$ . В этом случае к числу  $x$  прибавляется единица (так число попадает в нужный нам диапазон  $1 \leq x < 10^8$ ) и сумма записывается в любой регистр, где при косвенном обращении происходит модификация с уменьшением на единицу (это необходимо, чтобы вернуться к прежнему диапазону). Это может быть любой из регистров от 0 до 3. Пусть  $x=0,65$ . Прибавим к числу единицу и запишем сумму в Р3. Выполним команды КП→X3 П→X3 «0». Убедившись на практике в работоспособности метода, перейдем к программам.

**Б3.1.** Целая часть  $[x]$ ,  $1 \leq x < 10^8$  (два варианта). Ввод:  $x$  В/O С/P (общий для двух вариантов).

00	X→P7 47	00	B $\uparrow$ 0E 04	KП→X0 G0
01	KП→X7 77	01	1 01 05	P→X0 60
02	P→X7 67	02	+ 10 06	C/P 50
03	C/P 50	03	X→P0 40	

Пример. 3,57 В/O С/P «3».

Пример. 0,28 В/O С/P «0».

Сравнивая программы, можно убедиться, что вторая программа является обобщением первой программы, включая ее как частный случай.

**Б3.2.** Дробная часть  $\{x\}$ ,  $0 \leq x < 10^8$ . Дробную часть числа удобно находить как разность данного числа и его целой части:  $\{x\} = x - [x]$ , а значит, и в этот раз нам придется иметь дело с двумя вариантами программы. Ввод:  $x$  В/O С/P (общий для двух вариантов).

00	X→P7 47	00	B $\uparrow$ 0E 05	F <sub>O</sub> 25
01	KП→X7 77	01	1 01 06	P→X7 67
02	F <sub>O</sub> 25	02	+ 10 07	— 11
03	P→X7 67	03	X→P7 47	08 C/P 50
04	— 11	04	KП→X7 77	
05	C/P 50			

Пример. 2,56 В/O С/P «0.56».

Примеры. 0,39 В/O С/P «0.39»; 2,56 В/O С/P «0.56».

**Б3.3.** Остаток от деления  $x$  на  $y$ ,  $0 < x, y < 10^8$ . Если  $q$  — частное и  $r$  — остаток от деления  $x$  на  $y$ , то  $r=x-qy$ , а  $q=[x/y]$ . На этом соотношении построена программа «Остаток».

00	X→PВ 4L	05	1 01 10	P→XB 6L
01	↔ 14	06	+ 10 11	P→X0 60
02	X→PA 4—	07	X→P0 40 12	× 12
03	P→XB 6L	08	KП→X0 G0 13	— 11
04	:	09	P→XA 6— 14	C/P 50

Инструкция. Ввод:  $x$  В $\uparrow$   $y$  В/O С/P.

Примеры. 128 В $\uparrow$  15 В/O С/P «8» — остаток от деления 128 на 15; 13 В $\uparrow$  25 В/O С/P «13».

**Б4.** Округление числа. Производится с точностью до половины единицы последнего значащего разряда, сохранившегося после округления. Абсолютная погрешность округления  $\varepsilon=0,5 \cdot 10^n$ , где  $n > 0$ , если округление должно производиться слева от разряда единиц;  $n < 0$ , если справа;  $n=0$ , если в разряде единиц.

00	X→PA 4—	08	F10 <sup>x</sup> 15	16	FX < 0 5C
01	FX <sup>2</sup> 22	09	+ 10 17	21	
02	F V— 21	10	FBX 0 18	FO 25	
03	↔ 14	11	— 11 19	/—/ 0L	
04	X→PВ 4L	12	P→XB 6L 20	C/P 50	
05	F10 <sup>x</sup> 15	13	F10 <sup>x</sup> 15 21	FO 25	
06	:	14	× 12 22	C/P 50	
07	7	07	P→XA 6—		

Инструкция. Ввод:  $n$  В $\uparrow$   $x$  В/O С/P.

Примеры. Округлить 3,6567 до 0,005, т. е.  $n=-2$ . Вводим 2 /—/ В $\uparrow$  3,6567 В/O С/P «3,66»; 125,4 до 5: 1 В $\uparrow$  125,4 В/O С/P «130»; 125,4 до 0,5; 0 В $\uparrow$  125,4 В/O С/P «125»; —38,76 до 0,05; 1 /—/ В $\uparrow$  38,76 /—/ В/O С/P «—38,8».

#### Структура программы

01, 02 : вычисление абсолютной величины числа.

05, 06 : деление результата на 10, чтобы установить запятую за последним верным разрядом.

07—11 : прибавление к числу константы  $10^7$ , разрядная сетка переполняется и тем самым отсекается дробная часть суммы, т. е. получается округленный результат.

Это основная часть программы. Чтобы получить округленное число, осталось вычесть  $10^7$ .

12—14 : возвращение запятой в положение, которое оно занимало до переноса.

15—17 : если данное число  $x \geq 0$ , результат вычисления индицируется, в противном случае продолжение вычислений.

18—20 : индикация результата с противоположным знаком.

## Б5. Обмен содержимым регистров А и В.

00	$P \rightarrow XA$	6—	02	$X \rightarrow PA$	4—	04	$X \rightarrow PB$	4L
01	$P \rightarrow XB$	6L	03	$\leftrightarrow$	14	05	C/P	50

Инструкция.  $a \ X \rightarrow PA \ b \ X \rightarrow PB \ B/O \ C/P \ P \rightarrow XA \ «b» \ P \rightarrow \rightarrow XB \ «a»$ .

Пример. 12  $X \rightarrow PA \ 7 \ X \rightarrow PB \ B/O \ C/P \ P \rightarrow XA \ «7» \ P \rightarrow XB \ «12»$ .

**Б6. Умножение числа на  $10^n$ .** В различных задачах, в том числе логических, нередко требуется умножить данное число на степень десятки. Для этого удобно использовать оператор ВП. Например, число 2, находящееся в РХ, умножить на 1000 — ВП 3 — на индикаторе  $2 \cdot 10^3$ , т. е. 2000. Число, хранящееся в РА, разделить на 10000  $P \rightarrow XA$  ВП 4 /—/ — на индикаторе частное. Предположим, в РА находится число 25,3; после выполнения команд ВП 4 /—/ читаем  $25,3 \cdot 10^{-4}$ . Чтобы нормализовать число, достаточно выполнить команду, например, В↑ — на экране  $2,53 \cdot 10^{-3}$ .

Если множимое равно нулю, оператор ВП использовать нельзя. Пусть содержимое РХ равно нулю, нажатие клавиши ВП сформирует в РХ единицу, поэтому, когда множимое вычисляется в процессе работы программы и не известно заранее, будет ли оно неравным нулю, следует предусмотреть возможность появления нуля, оговорив этот случай в программе, или отказаться от использования оператора ВП. В любом случае можно использовать операторы F10<sup>x</sup> ×. Например, в РХ находится число 2.3. Чтобы умножить его на 100, набираем 2 F10<sup>x</sup> × читаем «230». В РХ — нуль, умножаем его на 10: 1 F10<sup>x</sup> × «0».

## ЗАДАНИЯ

1. Постройте алгоритм ОСТАТОК на УАЯ.

2. Составьте модификацию программы ОСТАТОК для случая, когда  $x$  задано не меньше  $y$ . Как основу удобно использовать программу Б2.1.

3. Составьте программы для вычисления наибольшего общего делителя (НОД) и наименьшего общего кратного (НОК) двух натуральных чисел. Запишите соответствующий алгоритм на УАЯ: НОД ( $c, d$ ); НОК ( $c, d$ ). Задачу поможет решить следующий алгоритм НОД:

1. Большее из двух чисел разделить на другое число.

2. Если остаток равен нулю, то меньшее число есть НОД данных чисел, в противном случае большее число заменить остатком.

3. Возвратиться к указанию 1.

Нетрудно заметить, что в рассматриваемом случае частное от деления не меньше единицы, а значит, остаток можно найти по программе, составленной в задании 2. Наименьшее общее кратное связано с наибольшим общим делителем несложным соотношением:  $\text{НОК}(x, y) = x \cdot y / \text{НОД}(x, y)$ .

4. Составьте программу для вычисления НОД, основанную на последовательном вычитании чисел, по следующему алгоритму:

1. Если числа равны, то каждое из них есть НОД. Конец.

2. Из большего числа вычесть меньшее.

3. Большее число заменить полученной разностью.

4. Возвратиться к указанию 1.

Запишите также соответствующий алгоритм поиска НОД на УАЯ.

Примечание. В заданиях 3 и 4 удобно использовать программу Б5.

## ОТВЕТЫ

### 2. Модифицированная программа ОСТАТОК

00	$X \rightarrow PB$	4L	04	:	13	08	$P \rightarrow XB$	6L
01	$\leftrightarrow$	14	05	$X \rightarrow P7$	47	09	$P \rightarrow X7$	67
02	$X \rightarrow PA$	4—	06	$KP \rightarrow X7$	Г7	10	$\times$	12
03	$P \rightarrow XB$	6L	07	$P \rightarrow XA$	6—	11	$\overline{-}$	11
						12	C/P	50

Инструкция. Ввод:  $x \ B \uparrow y \ B/O \ C/P$ .

3. Программа НОД ( $c, d$ ) и алгоритм УАЯ.

00	$X \rightarrow PD$	4Г	09	$X \rightarrow PC$	4C	18	$P \rightarrow XD$	6Г
01	$\leftrightarrow$	14	10	$\leftrightarrow$	14	19	$P \rightarrow X7$	67
02	$X \rightarrow PC$	4C	11	$X \rightarrow PD$	4Г	20	$\times$	12
03	$P \rightarrow XD$	6Г	12	$P \rightarrow XC$	6C	21	$\overline{-}$	11
04	$\overline{-}$	11	13	$P \rightarrow XD$	6Г	22	FX=0	5E
05	FX<0	5C	14	$P \rightarrow XD$	6Г	23	02	02
06	12	12	15	$X \rightarrow P7$	47	24	$P \rightarrow XD$	6Г
07	$P \rightarrow XC$	6C	16	$KP \rightarrow X7$	Г7	25	C/P	50
08	$P \rightarrow XD$	6Г	17	$P \rightarrow XC$	6C			

Инструкция. Ввод:  $c \ B \uparrow d \ B/O \ C/P$ .

алг НОД и НОК (нат  $c, d$ , НОД, НОК)

арг  $c, d$   
рез НОД, НОК

нач цел  $a, \ nat \ b, s$

$a := c; \ b := d$

пока  $a > 0$

ни

если  $a < b$

то  $s := a; \ a := b; \ b := s$

все

$a := a - [a/b] * b$

кц

НОД :=  $b$ ; НОК :=  $c * d / \text{НОД}$

кон

### 4. Модифицированная программа и алгоритм поиска НОД ( $c, d$ ).

00	$X \rightarrow PD$	4Г	07	$P \rightarrow XD$	6Г	14	$\leftrightarrow$	14
01	$\leftrightarrow$	14	08	C/P	50	15	$X \rightarrow PD$	4Г
02	$X \rightarrow PC$	4C	09	FX<0	5C	16	$P \rightarrow XC$	6C
03	$P \rightarrow XD$	6Г	10	$\overline{-}$	16	17	$P \rightarrow XD$	6Г
04	$\overline{-}$	11	11	$P \rightarrow XC$	6C	18	$\overline{-}$	11
05	FX=0	5E	12	$P \rightarrow XD$	6Г	19	X→PC	4C
06	09	09	13	$X \rightarrow PC$	4C	20	БП	51
						21	03	03

Инструкция. Ввод: с В↑ d В/О С/П  
алг НОД-МОДИФИКАЦИЯ (нат с, d, НОД)  
арг с, d  
рез НОД  
нач нат s  
пока с ≠ d  
нц  
если с < d  
то с := c; c := d; d := s  
все  
с := с - d  
кц  
НОД := d  
кон

## 2.2. ОРГАНИЗАЦИЯ ЦИКЛОВ С ПОМОЩЬЮ СЧЕТЧИКОВ И ОПЕРАТОРОВ УСЛОВНОГО ПЕРЕХОДА

### 1. Вычисление факториала $k! = k(k-1)(k-2) \dots 2 \cdot 1$ .

00	X→П0	40	03	X	12	05	02	02
01	1	01	04	FL0	5Г	06	C/П	50
02	П→Х0	60						

Инструкция. Ввод: k В/О С/П.

алг ФАКТОРИАЛ (нат k; Ф)  
арг k  
рез Ф  
нач нат i  
i := 1; Ф := 1  
пока i < k  
нц  
i := i + 1; Ф := Ф \* i  
кц  
кон

Счетчиком будем называть один из регистров памяти Р0—Р3, начальное содержимое которого равно числу необходимых повторений цикла. Для решения этой задачи воспользуемся регистром Р0.

Пусть Р0 — счетчик  $k$  последовательных повторений. Будем вычислять факториал как произведение, равное предыдущему результату, умноженному на содержимое Р0. При каждом обращении к оператору FL0 содержимое регистра Р0 уменьшается на единицу, единице равно и начальное произведение (команды 01—04). В итоге вычисляется произведение  $k(k-1)(k-2)\dots 2 \cdot 1 = k!$ , т. е. искомый факториал. Управление будет передаваться к началу программы до тех пор, пока содержимое Р0 будет больше нуля. Управляя в программе организацией цикла, регистр Р0 одновременно «поставляет» множители для вычисления  $k!$ .

2. Вычисление числа сочетаний  $C^n_m = m!/n!(m-n)!$ . Рассмотрим программу, в которой в качестве подпрограммы использована программа вычисления факториала, и алгоритм.

00	ПП	53	07	ПП	53	13	1	01
01	11	11	08	11	11	14	П→Х0	60
02	↔	14	09	:	13	15	×	12
03	ПП	53	10	С/П	50	16	FL0	5Г
04	11	11	11	X→П0	40	17	14	14
05	:	13	12	F0	25	18	B/O	52
06	↔	14						

Инструкция. (m - n) В↑ n В↑ m В/О С/П.

Контрольный пример. Вычислить по программе  $C_5^2 = 3$  В↑ 2 В↑ 5 В/О С/П «10».

### Структура программы

00,01 : вычисление  $m!$ ,  
02 : результат передается в РY, значение  $n$  — в РX,  
03,04 : вычисление  $n!$ ,  
05 : расчет  $m!/n!$ ,  
06 : частное сохраняется в РY, значение  $m - n$  направляется в РX,  
07,08 : расчет  $(m-n)!$ ,  
09,10 : вычисление  $C^n_m$  как частного от деления числителя расчетной формулы, т. е. содержимого РY, на знаменатель — содержимое РX и останов для индикации.

алг СОЧЕТАНИЯ (нат m, n, C)  
арг m, n  
рез C  
нач нат D, цел p  
ФАКТОРИАЛ (m, C)  
ФАКТОРИАЛ (n, D)  
C := C/D  
p := m - n  
ФАКТОРИАЛ (p, D)  
C := C/D  
кон

3. Биномиальные коэффициенты. Расчетная формула:  $(a+b)^m = C^0_m a^m + C^1_m a^{m-1} b + \dots + C^k_m a^{m-k} b^k + \dots + C^m_m b^m$  (называется формулой бинома Ньютона),  $C^0_m, C^1_m, \dots, C^k_m, \dots, C^m_m$  — биномиальные коэффициенты, часто встречаются в алгебре, математическом анализе, теории вероятностей. Вычислять коэффициенты может программируемый микрокалькулятор. Рассмотрим программу «Биномиальные коэффициенты-1», которая каждый следующий коэффициент, начиная со второго, рассчитывает по формуле  $C^{n-1}_m = C^n_m (m-n)/(n+1)$ . Здесь  $C^0_m = 1$ , а  $n=0, 1, \dots, m-1$ .

## Программа БИНОМИАЛЬНЫЕ КОЭФФИЦИЕНТЫ-1

00	X → П0	40	05	KП → X4	G4	10	:	13
01	CX	0Г	06	F0	25	11	FLO	5Г
02	X → П4	44	07	P → X0	60	12	04	04
03	1	01	08	X	12	13	B/O	52
04	C/P	50	09	P → X4	64			

**Инструкция.** Ввод:  $m$  B/O C/P «1». После каждого следующего нажатия клавиши C/P на индикаторе будет значение рассчитанного биномиального коэффициента, пока вновь не появится 1.

**Контрольный пример.** Вычислить  $C_5^0 - C_5^5$ : 5 B/O C/P «1» C/P «5» C/P «10» C/P «10» C/P «5» C/P «1».

## Структура программы

00—02 : подготовка начальных данных,  $m$  — начальное значение, которое записывается в счетчик количества вычисленных коэффициентов P0. Этот же регистр используется для формирования очередного коэффициента. Далее очищается P4, в котором будет храниться переменное значение  $n+1$ .

03,04 : когда цикл завершается, индицируется значение  $C_m^0$ , равное единице.

05 : увеличение на единицу содержимого P4.

06—10 : вызванное при косвенном обращении число не нужно для расчетов и вращением стека выводится в РТ, одновременно в РХ поступает  $C_m^n$ , чтобы участвовать в вычислении  $C_m^{n+1}$ . Последующие умножение и деление вычисляют коэффициент  $C_m^{n+1}$ .

11—13 : пока содержимое счетчика положительно, коэффициенты индицируются. Когда это условие не выполняется, управление по команде B/O передается команде с адресом 01. В итоге индицируется 1, означающая, что процесс вычисления коэффициентов завершен.

Запишем на УAY соответствующий алгоритм:

алг БИНОМИАЛЬНЫЕ КОЭФФИЦИЕНТЫ (нат,  $m$  нат таб БК [1 : ( $m + 1$ )])

арг  $m$

рез БК

нач нат  $n, x$

$n := 1$ ; БК [1] := 1

пока  $n \leq m$

нц

$x := \text{БК}[n] * (m - n + 1)/n$ ;  $n := n + 1$ ; БК [n] := x

кц

кон

Программа, которую мы рассмотрели, выполняет избыточную работу. Известно, что биномиальные коэффициенты, возрастаю-

46

слева направо, после достижения максимума убывают так, что коэффициенты, равноотстоящие от концов, равны. Программа «Биномиальные коэффициенты-2» индицирует только первую половину коэффициентов, тем самым время вычислений уменьшается в два раза.

## Программа БИНОМИАЛЬНЫЕ КОЭФФИЦИЕНТЫ-2

00	X → П0	40	07	V↑	0E	14	18	18
01	CX	0Г	08	P → X0	60	15	FBX	0
02	X → П4	44	09	X	12	16	FL0	5Г
03	1	01	10	P → X4	64	17	04	04
04	C/P	50	11	:	13	18	K—	27
05	KП → X4	G4	12	—	11			
06	F0	25	13	FX < 0	50			

**Инструкция.** Ввод:  $m$  B/O C/P.

**Контрольный пример.** 8 B/O C/P «1» C/P «8» C/P «28» C/P «56» C/P «70» C/P «ЕГГОГ» — признак скончания вывода коэффициентов.

Не разбирая подробно структуру программы, поясним принцип ее действия. После вычисления очередного коэффициента сравниваем его с предыдущим. Если новый коэффициент оказывается больше, он выводится на индикатор, в противном случае на индикаторе появится ЕГГОГ — за это отвечает команда 18. Чтобы организовать сравнение, по команде 07 предыдущий результат направляется в РY. С помощью команды 12 вычисляется разность. Если она отрицательна, то по команде 15 последующий коэффициент выводится из РХ1 для индикации. В противном случае по команде 14 формируется сигнал ЕГГОГ. Выход из цикла произойдет только в том случае, если не выполняется условие  $x < 0$ .

Эту программу также можно модифицировать. Введем дополнительный счетчик на основе P1, который будет подсчитывать количество вычисленных коэффициентов (команды 01—05 программы «Биномиальные коэффициенты-3»). Несколько иной подходложен в основу программы «Биномиальные коэффициенты-4» — она выводит все коэффициенты, но оптимальна по числу использованных команд. Появление на экране нулей означает, что формирование и вывод коэффициентов завершены.

## Программа БИНОМИАЛЬНЫЕ КОЭФФИЦИЕНТЫ-3

00	X → П0	40	07	X → П4	44	14	P → X4	64
01	2	02	08	1	01	15	:	13
02	:	13	09	C/P	50	16	FL1	5L
03	1	01	10	KП → X4	G4	17	19	19
04	+	10	11	F0	25	18	K—	27
05	X → П1	41	12	P → X0	60	19	FL0	5Г
06	CX	0Г	13	X	12	20	09	09

**Инструкция.** Ввод:  $m$  B/O C/P.

# Программа БИНОМИАЛЬНЫЕ КОЭФФИЦИЕНТЫ-4

00	X → П0	40	06	П → X0	60	11	FO	25
01	1	01	07	П → X4	64	12	П → X4	64
02	X → П4	44	08	—	11	13	:	13
03	C/P	50	09	Х	12	14	БП	51
04	П → X0	60	10	КП → X4	Г4	15	05	05
05	C/P	50						

Инструкция. Ввод  $m$  В/О С/П.

## 2.3. ОБЪЕДИНЕНИЕ ПРОГРАММ. ОПТИМИЗАЦИЯ СОСТАВНОЙ ПРОГРАММЫ

1. Объединенная программа для вычисления факториала, числа размещений и сочетаний. В программе «Сочетание» предусмотрено вычисление  $m!$  и  $(m-n)!$ . Остается вычислить и сохранить  $A_m^n = m!/(m-n)!$ ,  $C_m^n = m!/(m-n)!/n!$ . Сохранив значение  $m!$ , мы решим поставленную задачу. Для этого в объединенной программе «Комплекс» служат три дополнительные команды 02, 07, 12.

### Программа КОМПЛЕКС

00	ПП	53	08	↔	14	16	1	01
01	14	14	09	ПП	53	17	П → X0	60
02	X → ПА	4—	10	14	14	18	Х	12
03	↔	14	11	:	13	19	FL0	5Г
04	ПП	53	12	X → ПС	4C	20	17	17
05	14	14	13	C/P	50	21	B/O	52
06	:	13	14	X → П0	40			
07	X → ПВ	4L	15	FO	25			

Инструкция.  $n$  В $\uparrow$   $(m-n)$  В $\uparrow$   $m$  В/О С/П. После того, как индикатор вновь засветится: П → XA « $m!$ », П → XB « $A_m^n$ », П → XC « $C_m^n$ ».

2. В программе «Сочетания» трижды используется подпрограмма «Факториал». Нетрудно заметить, что это означает лишние вычисления. Рассчитывая, например,  $C_{10}^6 = 10!/6!/4!$ , калькулятор независимо вычисляет каждую компоненту, а затем выполняет деление:  $C_{10}^6 = 3\ 628\ 800/720/24 = 210$ . Прямое вычисление  $C_{10}^6 = 10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 / 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2$  дает результат быстрее.

Попробуем оптимизировать нашу программу, воспользовавшись программой «Биномиальные коэффициенты-2». Предусмотрим в ней останов и индикацию  $C_m^n$  для заданного  $n$ . Назовем эту программу «Сочетания-2». Число  $n$  записывается в Р1 (команда 00). Вычисления идут до тех пор, пока содержимое Р4 не сравняется с  $n$ . Когда это произойдет, на индикатор будет выведено  $C_m^n$  (команды 06–12). В остальном структура программы осталась неизменной.

# Программа СОЧЕТАНИЯ-2

00	X → П1	41	08	—	11	16	П → X0	60
01	↔	14	09	FX = 0	5E	17	Х	12
02	X → П0	40	10	10	13	18	П → X4	64
03	CX	0Г	11	F0	25	19	:	13
04	X → П4	44	12	C/P	50	20	FL0	5Г
05	1	01	13	F0	25	21	06	06
06	П → X4	64	14	KП → X4	Г4	22	B/O	52
07	П → X1	61	15	FO	25			

Инструкция.  $t$  В $\uparrow$   $n$  В/О С/П.

Контрольный пример. Вычислим  $C_{10}^4 = 10$  В $\uparrow$  4 В/О С/П (17 с) «210». По сравнению с программой «Сочетание» новая программа длиннее на четыре команды. Время решения контрольного примера при этом сократилось на 7 с. Выигрыш во времени получен ценой удлинения программы.

В следующей программе «Сочетание-3» новые значения  $C_m^n$  вычисляются без использования программы «Факториал». Эта программа рассчитывает  $C_{10}^4$  за 12 с, т. е. быстрее всех предыдущих и короче их.

### Программа СОЧЕТАНИЯ-3

00	X → П0	40	06	Х	12	11	↔	14
01	↔	14	07	↔	14	12	FL0	5Г
02	X → П1	41	08	П → X1	61	13	05	05
03	1	01	09	Х	12	14	:	13
04	B $\uparrow$	0E	10	KХ → П1	L1	15	C/P	50
05	П → X0	60						

Инструкция. Ввод:  $t$  В $\uparrow$   $n$  В/О С/П.

Контрольный пример. 10 В $\uparrow$  4 В/О С/П (12 с) « $C_{10}^4 = 210$ ».

### ЗАДАНИЯ

1. Запишите на учебно-алгоритмическом языке алгоритм программы «Сочетания-3».

2. Составьте программу суммирования квадратов натуральных чисел от 1 до  $n$ , взяв за основу программу «Факториал». Для контроля запрограммируйте известную математическую формулу суммы квадратов  $n$  последовательных натуральных чисел:  $S_n^{(2)} = n(n+1)(2n+1)/6$ . Составьте единую программу с двумя остановами для индикации результатов последовательного сложения и полученных по формуле.

3. Составьте программу суммирования кубов  $n$  последовательных натуральных чисел:  $1^3 + 2^3 + \dots + n^3$ . На этот раз за основу удобно взять программу суммирования квадратов последовательным сложением. Придется после команды возвведения в квадрат вставить две команды: FBX и Х. Составьте контрольную программу, воспользовавшись формулой суммирования кубов:  $S_n^{(3)} = (n(n+1)/2)^2$ .

4. Модернизируйте программу вычисления  $S_n^{(3)}$ , используя для возведения в куб оператор Ху. В отличие от предыдущей программы, этот вариант легко приспособить для вычисления  $S_n^{(4)}$ .

5. Составьте программу суммирования квадратов  $n$  произвольных действительных чисел ( $n \leq 13$ ), используя возможности косвенной адресации. Данные разместите в регистрах начиная с Р1. В этом случае очередное слагаемое не трудно получить по команде КП → X В $\uparrow$ , которая не модифицирует содержи-

мого Р0. Подсчет циклов с одновременным формированием адреса нужного регистра, как обычно, выполняет FL0. Программа должна быть составлена так, чтобы диалог с ней протекал следующим образом:  $x_1 \rightarrow \text{П1}$ ,  $x_2 \rightarrow \text{П2}$ ,  $x_3 \rightarrow \text{П3}$ , ... В/О С/П « $S^2_n$ ».

## ОТВЕТЫ

### 2.1. Ввод: $n$ В/О С/П.

00	X → П0	40	03	FX <sup>2</sup>	22	06	02	02
01	CX	0Г	04	+ FL0	10	07	C/П	50
02	П → X0	60	05		5Г			

### 2.2. Ввод: $n$ В/О С/П.

00	B↑	0E	05	X	12	10	+	10
01	1	01	06	FBX	0	11	×	12
02	↔	14	07		2	02	12	6
03	+	10	08		X	12	:	13
04	FBX	0	09		1	01	14	C/П 50

### 3.1. Ввод: $n$ В/О С/П.

00	X → П0	40	04	FBX	0	08	02	02
01	CX	0Г	05	X	12	09	C/П	50
02	П → X0	60	06	+	10			
03	FX <sup>2</sup>	22	07	FL0	5Г			

### 3.2. Ввод: $n$ В/О С/П.

00	B↑	0E	03	+	10	06	:	13
01	B↑	0E	04	X	12	07	FX <sup>2</sup>	22
02	1	01	05	2	02	08	C/П	50

### 4. Ввод: $n$ В/О С/П.

00	X → П0	40	04	FX <sup>Y</sup>	24	08	FL0	5Г
01	CX	0Г	05	↔	14	09	02	02
02	3	03	06	FO	25	10	C/П	50
03	П → X0	60	07	+	10			

### 5. Ввод: $x_1 \rightarrow \text{П1}, x_2 \rightarrow \text{П2}, \dots, x_n \rightarrow \text{Пn}$ В/О С/П.

00	X → П0	40	03	FX <sup>2</sup>	22	06	02	02
01	CX	0Г	04	+	10	07	C/П	50
02	KП → XB↑	ГЕ	05	FL0	5Г			

## 2.4. ЦИКЛЫ И МАССИВЫ

Числовым массивом будем называть упорядоченное множество чисел. Для работы с массивами удобно использовать команды косвенной адресации, которые мы до сих пор применяли для отделения целой части числа.

Выполняя команду косвенной адресации, например КП → X $\alpha$ , где  $\alpha$  — адрес регистра памяти, ПМК помещает в RX число, извлеченное из регистра, адрес которого хранится в регистре  $\alpha$ .

Рассмотрим конкретный пример косвенной адресации. Пусть число 12,6 хранится в Р6, а адрес этого регистра, т. е. число 6, в свою очередь, записано в Р8 (выполните 12,6 X → П6, 6 X → П8), тогда команда косвенной адресации КП → X8 вызовет на экран число 12,6.

Для последовательной обработки элементов массива удобно использовать команды косвенной адресации к регистрам 0, 1, 2 и 3. При модификации из содержимого этих регистров вычитается единица.

Особое место среди команд косвенной адресации занимает КП → XB↑. Эта команда работает только на микрокалькуляторах «Электроника» МК-56, МК-54, Б3-34. Действие команды напоминает действие команды косвенной адресации к регистру 0: КП → X0, однако в отличие от нее не производится уменьшение на единицу содержимого регистра Р0. Командой КП → X B↑ удобно пользоваться, когда обращение к элементам массива происходит в цикле. В этом случае элементы массива размещаются последовательно, начиная с регистра Р1. Номер регистра, в котором размещен последний элемент массива, помещается в счетчик, который с помощью команды FL0 организован в Р0. В этом случае по команде КП → X B↑ в RX помещается очередной элемент массива, а обращение к команде FL0 уменьшает на единицу содержимое Р0. Так, элементы массивачитываются в порядке убывания номеров регистров, где они хранятся.

В новых моделях микрокалькуляторов «Электроника» МК-52 и МК-61 добавился 15-й регистр памяти PE, однако стало невозможным использование команды КП → X B↑. Впрочем, в некоторых случаях команду можно заменить. Есть несколько вариантов замены.

Фрагмент П → X0 X → ПЕ КП → XE может быть реализован на ПМК новых марок почти во всех случаях. Нередко можно сократить число нажимаемых клавиш: П → X0 X → ПЕ ↔ КП → XE или П → X0 X → ПЕ КП → XE. Вместо регистра E можно использовать любой другой от Р7 до Р9 и регистры А, В, С, Д. Какой из фрагментов использовать в конкретном случае, зависит от того, насколько важно сохранить текущее состояние стека. Так, после первого фрагмента теряется содержимое регистра предыдущего результата — RX1, второй фрагмент переставляется на новые позиции числа из RX и PY, а содержимое PZ безвозвратно теряется вместе с числом из RX1. Третий фрагмент также оставляет только числа из RX и PY, но сдвигает их в PZ и PT.

Вернемся, однако, к нашей базовой модели «Электроника МК-56». Пусть, например, массив состоит из трех чисел 4, 5, 14 и требуется вычислить сумму их квадратов. Разместим числа в регистрах с Р1 по Р3: 4 X → П1, 5 X → П2 14 X → П3. Всего в массиве три элемента, поэтому занесем в Р0 число 3. Задачу решим с помощью следующей программы.

# Программа СУММА КВАДРАТОВ

00	CX	0Г	03	+	10	05	01	01
01	KП→ХВ↑	ГЕ	04	FL0	5Г	06		
02	FX <sup>2</sup>	22					C/П	50

Инструкция. Ввод: В/О С/П. Если данные введены без ошибок, то после останова вычислений на индикаторе появится число 237.

Программа несложная, поэтому достаточно будет краткого комментария. При первом обращении к команде КП→ХВ↑ содержимое Р0 равно 3, а значит, из Р3 вызывается число 14, возведется в квадрат и складывается с суммой предыдущих квадратов. В первый раз эта сумма равна нулю. После выполнения команды FL0 содержимое Р0 уменьшается на единицу и становится равным 2. Обращение на этот раз происходит к Р2, т. е. в квадрат возводится число 5 и т. д., пока счетчик не зафиксирует, что задача решена.

Воспользуемся полученной информацией, чтобы составить программу, вычисляющую значения многочлена  $P_n(x)$  для заданного значения аргумента  $x$ . Учитывая ограниченную память ПМК, составим программу для вычисления многочлена степени  $n$  не больше 12. В этом случае все 13 (или меньшее число) коэффициентов размещаются в Р1—РД. Коэффициенты  $c_0, c_1, \dots, c_n$  многочлена  $P_n(x) = c_0 + c_1x + \dots + c_nx^n$  предварительно записываем в регистры:  $c_0$  X→П1,  $c_1$  X→П2, ...,  $c_{13}$  X→ПД. Вычисления удобно вести по схеме Горнера. Расчетная формула:

$$P_n(x) = (((((0 \cdot x + c_n)x + c_{n-1})x + c_{n-2})x + \dots + c_1)x + c_0).$$

При вводе  $x$  его нужно продублировать во всех ячейках стека, тогда он будет использоваться программой в качестве константы.

## Программа ГОРНЕР-1

00	B↑	0E	04	CX	0Г	08	FL0	5Г
01	1	01	05	X	12	09	05	05
02	+	10	06	KП→ХВ↑	ГЕ	10	C/П	50
03	X→П0	40	07	+	10			

Инструкция.  $x$  B↑ B↑ n B/O С/П.

Примеры. 1. Вычислить  $P_4(2)$ , если  $P_4(x) = 4 - x + 3x^2 + 2x^3 + x^4$ . Вводим коэффициенты: 4 X→П1, 1 /-/ X→П2, 3 X→П3, 2 X→П4, 1 X→П5. Вводим  $x=2$  и степень многочлена  $n=4$ : 2 B↑ B↑ 4 B/O С/П «46».

2. Вычислить  $P_5(1)$ , если  $P_5(x) = 5 + 2x - 4x^2 + x^3 - 2x^4 + x^5$ . Вводим коэффициенты: 5 X→П1, 2 X→П2, 4 /-/ X→П3, 1 X→П4, 2 /-/ X→П5, 1 X→П6. Вводим  $x=1$  и степень многочлена  $n=5$ : 1 B↑ B↑ 5 B/O С/П «3».

3. Вычислить  $P_5(-2)$ . Повторный ввод коэффициентов не требуется: 2 /-/

B↑ B↑ 5 B/O С/П «-87».

Не разбирая формально структуру программы, поясним назначение ее фрагментов и функционирование. По командам 00—03 вычисляется  $n+1$  и записывается в счетчик коэффициентов многочлена — регистр Р0. Команда B↑ играет роль разделителя числа, последним введенного с клавиатуры,

и единицы. Команды 05—09 — тело цикла, в котором вычисляется  $sx + c_i$  ( $s$  — ранее накопленное выражение). Команда 04 нужна для начального вычисления  $0 \cdot x + c$ . При первом обращении к команде 06 вызывается содержимое регистра  $n+1$ , т. е.  $c_n$ . Модификацию содержимого Р0 выполняет команда 08 (FL0), так что при повторном использовании команды косвенной адресации вызывается уже содержимое регистра  $n$ , т. е.  $c_{n-1}$  и т. д. Одновременно по команде 08 проверяется содержимое счетчика и по завершении цикла управление передается команде 10, необходимой для прекращения вычислений и индикации результата.

Обратите внимание на достаточно сложный ввод данных в программу. Можно упростить ввод, однако в этом случае придется вносить небольшие изменения в текст программы всякий раз, когда потребуется менять степень многочлена. Модернизированная программа «Горнер-2» особенно удобна в тех случаях, когда нужно вычислять много значений для одного и того же многочлена, например, составляя таблицу.

## Программа ГОРНЕР-2

00	B↑	0E	06	+	10	12	FL0	5Г
01	B↑	0E	07	X→П0	10	13	09	09
02	0	00	08	CX	0Г	14	C/П	50
03	4	04	09	×	12	15	БП	51
04	B↑	0E	10	KП→ХВ↑	ГЕ	16	00	00
05	1	01	11	+	10			

Инструкция. Ввод:  $x$  B/O С/П.

Примеры.  $P_4(x) = 4 - x + 3x^2 + 2x^3 + x^4$ . Поскольку  $n=4$ , то после ввода программы вносим в ее текст необходимые изменения: БП 02 F ПРГ 0 «00 0E 0E 04» 4 «04 00 0E 05» FABT. Вводим коэффициенты: 4 X→П1, 1 /-/ X→П2, 3 X→П3, 2 X→П4, 1 X→П5. Можно начинать расчеты: 2 B/O С/П, « $P_4(2)=46$ »; 3 C/П, « $P_4(3)=163$ »; 1 /-/ С/П, « $P_4(-1)=7$ » и т. д.

Без коррекции программы можно обойтись, если немного усложнить ввод.

## Программа ГОРНЕР-3

00	X→П0	40	04	0	00	08	FL0	5Г
01	FO	25	05	×	12	09	05	05
02	B↑	0E	06	KП→ХВ↑	ГЕ	10	C/П	50
03	B↑	0E	07	+	10			

Инструкция.  $x$  B↑ (n+1) B/O С/П.

## ЗАДАНИЯ

- Составьте программу записи чисел, вводимых с клавиатуры в регистры памяти с Р1 до РД. Для решения задачи удобно воспользоваться оператором косвенной адресации к Р0 без модификации, т. е. командой KХ→ПВ↑. Измен-

нение адреса текущего регистра выполняет команда FL0. В качестве образца можно воспользоваться программой «Сумма квадратов». Работа с программой должна начинаться с ввода адреса старшего регистра, т. е. числа 13. Числа вводятся в ПМК (регистр X) при очередных остановах.

2. Составьте программу, очищающую регистры с P1 по P9. Следует иметь в виду, что ввод данных не нужен: адрес старшего регистра и число 0, которое будет записано во все регистры, должны формироваться непосредственно в теле программы.

3. Составьте программу, очищающую регистры с P6 по РД. Эта задача существенно отличается от предыдущей. Нетрудно заметить, что наибольший номер регистра, в котором должен будет храниться элемент формируемого массива нулей, не совпадает с числом элементов этого массива: число элементов — 8, а номер регистра РД—13. Таким образом, содержимое Р0 нельзя одновременно использовать в качестве счетчика и указателя адреса очередного регистра.

4. Программа «Горнер» может быть использована для перевода чисел из произвольной  $p$ -ичной системы счисления в десятичную, если  $p \leq 9$ . В этом случае  $n$  — это число цифр  $p$ -ичного числа, а  $x=p$ . Ввод напоминает прежний:  $p$  B↑ B↑ (n—1) B/O C/P.

Небольшие отличия удобно иллюстрировать примером: перевести  $16\ 547_8$  в десятичную форму. Числа вводятся в регистры памяти в обратном порядке: 7 X→P1 4 X→P2 5 X→P3 6 X→P4 1 X→P5 8 B↑ B↑ 4 B/O C/P « $16547_8=7527_{10}$ ». Постройте для решения этой задачи свой вариант программы.

5. Запишите алгоритм программы «Горнер» на УАЯ.

### ОТВЕТЫ

#### 1. 13 B/O C/P «л»

00	X→P0	40	03	FL0	5Г	05	Fπ	20
01	C/P	50	04	01	01	06	C/P	50
02	KX→ПВ↑ LE							

#### 2. CX B/O C/P «0»

00	9	09	03	KX→ПВ↑ LE	05	03		
01	X→P0	40	04	FL0	5Г	06	C/P	50
02	CX	0Г						

#### 3. CX B/O C/P «0»

00	8	08	03	X→P4	44	06	FL0	5Г
01	X→P0	40	04	CX	0Г	07	04	04
02	5	05	05	KX→P4	L4	08	C/P	50

#### 5. Алгоритм на УАЯ.

алг ГОРНЕР (цел  $n$ , веш  $x$ ,  $p$ , веш таб  $C[0:n]$ )

арг  $n$ ,  $C$ ,  $x$

рез  $p$

нач цел  $i$

$p := 0$ ;  $i := n$

пока  $i \geq 0$

нц

$p := p * x + C$

$i := i - 1$

кц

кон

### 2.5. ПОЭТАПНАЯ ОБРАБОТКА ДАННЫХ

Работая с большими объемами данных на ПМК, часто приходится сталкиваться с отсутствием необходимой емкости памяти для их хранения. В этом случае удобно вводить информацию последовательно, небольшими порциями, обрабатывать ее по мере поступления, накапливая лишь результат. Процесс, организованный подобным образом, будем называть режимом поэтапного ввода данных и их обработки.

Поэтапную обработку данных удобно рассмотреть на примере составления программ статистической обработки материалов экспериментов и наблюдений.

Рассмотрим следующую задачу. Случайные величины  $X$  и  $Y$  могут описывать некоторую функциональную зависимость. Калькулятор находит ее характеристики: средние значения  $X$  и  $Y$  — соответственно  $\bar{x}$ ,  $\bar{y}$ ; среднеквадратические отклонения  $\sigma_x$  и  $\sigma_y$  (или их квадраты, т. е. дисперсии  $D_x=\sigma^2_x$ ,  $D_y=\sigma^2_y$ ); выборочный коэффициент корреляции  $\rho$ ; коэффициенты регрессии  $c_1$  и  $c_0$ . Расчетные формулы:

$$\bar{x} = \sum x_i / n, \quad \bar{y} = \sum y_i / n;$$

$$\sigma_x = \sqrt{\sum x_i^2 / n - \bar{x}^2}, \quad \sigma_y = \sqrt{\sum y_i^2 / n - \bar{y}^2};$$

$$\rho = (\sum x_i y_i / n - \bar{x} \bar{y}) / \sigma_x \sigma_y;$$

$$c_1 = \rho \sigma_y / \sigma_x, \quad c_0 = \bar{y} - c_1 \bar{x}.$$

При суммировании  $i$  изменяется от 1 до  $n$ .

Коэффициент корреляции  $\rho$  характеризует взаимосвязь между переменными  $X$  и  $Y$ , изменяется в пределах  $-1 \leq \rho \leq 1$ . Близость  $\rho$  к единице свидетельствует о наличии между переменными линейной зависимости:  $y_i$  может возрастать или убывать по мере возрастания  $x_i$ .

Коэффициенты регрессии описывают линейную функцию, являющуюся наилучшим в смысле наименьших квадратов приближением к данной зависимости. Уравнение  $y = c_0 + c_1 x$ , приближающее зависимость, называется уравнением регрессии.

# Программа СТАТИСТИКА

Распределение памяти

P0 P1, P2 P3 P4, P5 P6, P7	n, переменная Сначала $y_i$ , $x_i$ , затем $c_0$ , $c_1$ $\rho$ $\sigma_y$ , $\sigma_x$ $y$ , $\bar{x}$	P8 P9, PA PB, PC РД	$\Sigma xy$ $\Sigma y^2$ , $\Sigma x^2$ $\Sigma y$ , $\Sigma x$ n, константа
00	X→P0	40	29
01	X→ПД	4Г	30
02	CX	0Г	31
03	X→П8	48	32
04	X→П9	49	33
05	X→ПА	4—	34
06	X→ПВ	4L	35
07	X→ПС	4C	36
08	C/P	50	37
09	X→П1	41	38
10	↔	14	39
11	X→П2	42	40
12	P→XC	6C	41
13	+ X→ПС	4C	42
14	X→П1	4C	43
15	P→X1	61	44
16	P→XB	6L	45
17	+ X→ПВ	4L	46
18	X→П2	47	48
19	P→X2	62	48
20	FX <sup>2</sup>	22	49
21	P→XA	6—	50
22	+ X→ПА	4—	51
23	X→П1	4—	52
24	P→X1	61	53
25	FX <sup>2</sup>	22	54
26	P→X9	69	55
27	+ X→П9	49	56
28	X→П9	49	57
			FX <sup>2</sup> 22

Инструкция. n В/О С/П «0»,  $x_1$  В↑  $y_1$  С/П ...  $x_2$  В↑  $y_2$  С/П ...  $x_n$  В↑  $y_n$  С/П, «j». Расчет окончен. Результаты можно извлечь из регистров: П→X1 « $c_0$ », П→X2 « $c_1$ », П→X3 « $\rho$ », П→X4 « $\sigma_y$ »; П→X5 « $\sigma_x$ »; П→X6 « $\bar{y}$ »; П→X7 « $\bar{x}$ ».

Контрольный пример. Задана зависимость растворимости Y(%) тиосульфата натрия от его температуры X( $^{\circ}$ С).

X	0	10	20	30	40	50	60	70	80
Y	33,5	37,0	41,2	46,1	50,0	52,0	56,3	64,3	69,9

Вычисляем: 9 В/О С/П «0» 0 В↑ 33,5 С/П (10 c) 10 В↑ 37  
 C/P (10 c) 20 В↑ 41,2 C/P ... 80 В↑ 69,9 C/P «j».  
 Результат: П→X1 « $c_0=32.459997$ », П→X2 « $c_1=0.4393334$ », П→X3 « $\rho=0.99147648$ », П→X4 « $\sigma_y=11.441057$ », П→X5 « $\sigma_x=25.819887$ », П→X6 « $\bar{y}=50.033333$ », П→X7 « $\bar{x}=40$ ».

Таким образом, коэффициент корреляции близок к единице, а значит, зависимость линейная. Уравнение регрессии для этой зависимости  $y=32,46+0,439x$ . Постройте на миллиметровой бумаге прямую регрессии по ее уравнению и нанесите на график координаты заданных точек. Очевидно, что заданные точки расположены практически на рассчитанной прямой.

## Структура программы

Программа состоит из двух основных частей: формирования сумм (команды 00—34) и вычисления искомых характеристик по заданным формулам (команды 37—85). Переходом от первой части программы ко второй управляет счетчик (команды 35, 36).

00—08 : подготовка данных для последующих вычислений,  $n$  записывается в РД, поскольку оно потребуется для дальнейших расчетов в качестве константы, и в Р0 как начальное значение счетчика. Здесь же очищаются накопители — регистры с 8 по С, необходимые для подсчета сумм; по команде С/П (адрес 08) прерывается работа программы для ввода очередной пары значений  $x_i$  и  $y_i$  (где  $i=1, 2, \dots, n$ ).

09—11 : засылка очередной пары в Р1 и Р2.

12—34 : вычисление суммы:  $\Sigma x$  (12—14);  $\Sigma y$  (15—18);  $\Sigma x^2$  (19—23);  $\Sigma y^2$  (24—28);  $\Sigma xy$  (29—34).

37—83 : когда счетчик зафиксирован окончание ввода, калькулятор приступает к вычислению характеристик  $\bar{x}$  (37—40),  $\bar{y}$  (41—44),  $\sigma_x$  (45—52),  $\sigma_y$  (53—60),  $\rho$  (61—72),  $c_0$  (73—77),  $c_1$  (78—83).

84, 85 : формирование признака окончания вычислений и прерывание выполнения программы.

Примером программы еще более сложной структуры с поэтапным вводом и обработкой данных является программа выполнения арифметических операций с комплексными числами в алгебраической форме — «Комплекс-А». Она позволяет выполнить произвольное число арифметических операций сложения, умножения и деления с комплексными числами, заданными в алгебраической форме, т. е. с числами вида:  $a+bi$  и  $c+di$ . Первое из них часто является результатом предшествующих вычислений. Коэффициенты  $a, b, c, d$  хранятся в регистрах РА—РД. Таким образом, в РА и РВ постепенно накапливается результат выполненных операций, из этих же регистров извлекается ответ, когда задача решена. Все доступные действия закодированы: 0 — означает сложение, 1 — умножение, 2 — деление. Вычитание заменяется сложением. Расчетные формулы:

$$(a+bi) + (c+di) = (a+c) + (b+d)i;$$

$$(a+bi) (c+di) = (ac-bd) + (ad+bc)i;$$

$$1/(c+di) = c/(c^2+d^2) + (-d/(c^2+d^2))i;$$

$$(a+bi)/(c+di) = (a+bi)/(c+di).$$

# Программа КОМПЛЕКС-А

00	X→ПВ	4L	19	03	03	38	П→ХА	6—
01	F0	25	20	2	02	39	П→ХС	6C
02	X→ПА	4—	21	—	11	40	Х	12
03	C/P	50	22	FX=0	5E	41	П→ХВ	6L
04	X→ПД	4Г	23	38	38	42	П→ХД	6Г
05	F0	25	24	П→ХС	6C	43	Х	12
06	X→ПС	4C	25	B↑	0E	44	—	11
07	F0	25	26	FX <sup>2</sup>	22	45	П→ХА	6—
08	FX=0	5E	27	П→ХД	6Г	46	П→ХД	6Г
09	20	20	28	FX <sup>2</sup>	22	47	Х	12
10	П→ХА	6—	29	+	10	48	П→ХВ	6L
11	П→ХС	6C	30	:	13	49	П→ХС	6C
12	+	10	31	X→ПС	4C	50	Х	12
13	X→ПА	4—	32	FBX	0	51	—	10
14	П→ХВ	6L	33	П→ХД	6Г	52	X→ПВ	4L
15	П→ХД	6Г	34	/—/	0L	53	↔	14
16	+	10	35	↔	14	54	X→ПА	4—
17	X→ПВ	4L	36	:	13	55	БП	51
18	БП	51	37	X→ПД	4Г	56	03	03

Инструкция. а В↑ б В/О С/П — начальный ввод. Здесь а и б — действительная и мнимая части первого комплексного числа. После каждого следующего останова: k В↑ с В↑ d С/П, k — код операции (0 — сложение, 1 — умножение, 2 — деление); с и d — действительная и мнимая части очередного комплексного числа, участвующего в операциях.

Контрольный пример.  $(2+3,5i)(3-2i)^2/(4+6,2i)+(7-9,4i)$ . Возведение в квадрат второго сомножителя будем выполнять как последовательное умножение. Начальный ввод: 2 В↑ 3,5 В/О С/П. После останова вводим код операции умножения 1 и коэффициенты очередного числа: 1 В↑ 3 В↑ 2 /—/ С/П. Проверим промежуточный результат умножения  $(2+3,5i) \times (3-2i)$ . Заглянем в РА и РВ: П→ХА «13», П→ХВ «6,5». Итак, результат: 13+6,5i. Продолжим вычисления: 1 В↑ 3 В↑ 2 /—/ С/П. Контроль показывает, что числитель равен 52—6,5i. Далее вводим знаменатель: 2 В↑ 4 В↑ 6,2 С/П. Получаем значение дроби равное 3,0804555—6,3997056i. Наконец, 0 В↑ 7 В↑ 9,4 /—/ С/П. Окончательный результат записан в РА и РВ: 10,080456—15,799706i.

## Структура программы

00—03: первое число из тех, что будут участвовать в комбинированных действиях, принимается за исходное, его коэффициенты вводятся в регистры А и В.

04—06: при каждом очередном останове вводятся три числа: k — код текущей операции, которую надлежит выполнить; с — действительный коэффициент второго операнда и d — коэффициент мнимой части — в РС и РД.

07—08: подготовка к выполнению нужной операции.

10—19: сложение.

24—55: деление.

38—55: умножение.

В действиях умножения и деления используется общая часть программы, точнее деление включает умножение. Это сделано

для того, чтобы сократить программу. Коэффициенты очередного результата направляются в регистры РА и РВ, т. е. можно выполнять непрерывную цепочку вычислений. Цикл организован с помощью операторов безусловного перехода (команды 18—19 и 55—56).

## ЗАДАНИЯ

1. Вычислите  $(2,35 + 4,6i)(4,81 - 7,64i)^2(3,2 + 3i)^2/(4 - 2,8i)(7,93 + 6,15i) + (6,25 + 4,9i)$ . Ответ: 148,21043 + 89,62053i.

2. Измените программу, чтобы она стала программой-репетитором. Сделайте так, чтобы можно было вводить ответы, полученные расчетом в ручном режиме, а ПМК, сравнивая их с верными, выдавал бы на индикатор сигнализации 1, когда ответ совпадает, 0 в противном случае. Для удобства сравнения округлите результаты до 0,01.

## 2.6. ПОДПРОГРАММЫ

Если группа команд в составе программы повторяется несколько раз, то ее удобно оформить в виде подпрограммы. В языке ПМК предусмотрены операторы передачи управления подпрограмме и выхода из нее.

Обращение к подпрограмме из любой точки основной программы производится либо двухшаговой прямой командой, состоящей из операторов ПП и а (где а — адрес первой команды, входящей в состав подпрограммы) или же одношаговой командой косвенной адресации к подпрограмме: КППβ (где β — номер регистра памяти, содержимое которого есть а). Важно отметить, что β должно быть больше или равно 7, поскольку для косвенной адресации можно использовать только те регистры, при обращении к которым не вычитается единица из содержимого данного регистра. В противном случае при повторном обращении управление будет передано уже не на начало подпрограммы.

Возврат из подпрограммы выполняется по команде В/О — команде, которая записана в подпрограмме последней. Механизм подпрограммы работает следующим образом. При обращении к подпрограмме адрес команды, следующей за командой обращения, поступает в специальный стек, устроенный наподобие основного стека калькулятора. Там он хранится без изменений все время, пока выполняется подпрограмма. Встретив в подпрограмме заключающий ее оператор В/О, ПМК обращается к стеку возврата и считывает из него адрес возврата, тотчас же передавая управление команде с этим адресом. В некоторых случаях нужно из одной подпрограммы выйти в другую и т. д. Магазинное устройство стека возврата позволяет использовать до пяти вложенных подпрограмм. Каждое обращение к стеку возврата командой ППα записывает а в «нижнюю» ячейку стека, сдвигая остальные адреса «вверх» по стеку возврата. Очередная команда В/О освобождает «нижнюю» ячейку, а на освободившееся место сдвиги-

ются адреса из «верхних» ячеек. Свободная часть стека возврата заполнена адресом 01. Именно поэтому по адресу 01 передается управление командой В/О в отсутствие команд обращения к подпрограмме.

Рассмотрим в качестве примера использование подпрограмм для описания функций. Дело в том, что многие прикладные программы рассчитаны на работу с различными функциями. Удобно описывать функции в виде подпрограмм, помещаемых в конце программы, чтобы, производя расчеты, каждый раз менять лишь концовку, не вводя заново основную часть программы. Подпрограмму удобно использовать для приближенного решения уравнения методом половинного деления. Этот метод получил название метода дихотомии. Решим задачу: на сегменте  $[a, b]$  имеется единственный корень уравнения  $f(x)=0$ . В этом случае говорят, что корень изолирован на  $[a, b]$ .

На концах сегмента функция  $f(x)$  имеет противоположные знаки:  $f(a)f(b) < 0$ . Требуется вычислить корень уравнения с заданной точностью  $\varepsilon > 0$ , т. е. отыскать такое решение, чтобы абсолютная погрешность не превышала  $\varepsilon$ . Задачу в такой постановке называют уточнением корня. Например, если мы хотим уточнить корень до второго десятичного знака, то  $\varepsilon = 10^{-2}$  или  $\varepsilon = 0,5 \cdot 10^{-2}$ . Составим алгоритм решения.

#### Алгоритм 1.

1. Если длина сегмента, на котором изолирован корень, меньше  $\varepsilon$ , то задача решена. В качестве значения корня примем середину сегмента, в противном случае переход к указанию 2.

2. Сегмент делится пополам. Из двух сегментов выбирается тот, на концах которого функция имеет противоположные знаки. Выбранный сегмент принимается за сегмент, на котором изолирован корень.

#### 3. Возврат к указанию 1.

Такая запись алгоритма удобна для решения задачи человеком. Все операции: деления пополам, сравнения с  $\varepsilon$  и другие — понятны и доступны человеку. Человек мыслит и соответственно отражает это в языке укрупненными порциями информации. По этому пути пошли создатели современных языков программирования, разрабатывающие специальные процедуры, приближающие языки программирования к возможностям человеческого языка. Язык ПМК, напротив, состоит из операторов, выполняющих лишь элементарные действия: сложить, передать, сравнить с нулем и т. д. Из них еще предстоит построить сложные конструкции, необходимые для исполнения нашего алгоритма.

Приведенное выше описание последовательности действий неисполнимо для ПМК. Не удовлетворяя обязательному требованию доступности, данное описание не является алгоритмом для ПМК.

#### Алгоритм 2.

1. Вычислить координату середины сегмента, на котором изо-

лирован корень, отыскав среднее арифметическое координат концов сегмента.

2. Вычислить длину сегмента как разность координат его конца и начала.

3. Вычислить разность между длиной сегмента и  $\varepsilon$ .

4. Если полученная разность отрицательна, то задача решена. За значение корня с точностью  $\varepsilon$  принимается вычисленная при исполнении указания 1 координата середины сегмента. В противном случае переходим к указанию 5.

5. Вычислить значение  $f(x)$  для левого конца сегмента, запомнить полученное число.

6. Вычислить значение  $f(x)$  в середине сегмента, умножить его на число, найденное в указании 5.

7. Если произведение меньше нуля, то, сохранив левый конец, выбираем в качестве правого конца очередного сегмента середину текущего сегмента, переходим к указанию 1. В противном случае переходим к указанию 8.

8. Сохраняем правый конец, в качестве левого конца очередного сегмента выбираем середину текущего сегмента. Возвращаемся к указанию 1.

На этот раз каждое из описанных действий доступно для выполнения на ПМК. Описание уже является программой для машины, изложенной словами человеческого языка. Написание программы сводится тем самым к кодированию описания символов языка ПМК.

Начнем с того, что распределим адресуемые регистры памяти. Условимся, например, в РА, РВ, РС хранить заданные константы: координаты концов сегмента и значение  $\varepsilon$ . Регистр Д отведем для хранения переменной — координаты середины текущего сегмента изоляции корня. В регистрах 8 и 9 пусть хранятся переменные — координаты левого и правого концов очередного сегмента корня. Начальное содержимое этих регистров — это  $a$  и  $b$ . В Р7 будем запоминать значение функции, вычисленное на левом конце сегмента.

#### Программа ДИХОТОМИЯ

00	$P \rightarrow XA$	6—	13	$P \rightarrow XC$	6C	26	$P \rightarrow X7$	67
01	$X \rightarrow P8$	48	14	—	11	27	$\times$	12
02	$P \rightarrow XB$	6L	15	$FX < 0$	5C	28	$FX < 0$	5C
03	$X \rightarrow P9$	49	16	19	19	29	34	34
04	$P \rightarrow X8$	68	17	$P \rightarrow XD$	6Г	30	$P \rightarrow XD$	6Г
05	$P \rightarrow X9$	69	18	$C/P$	50	31	$X \rightarrow P9$	49
06	+	10	19	$P \rightarrow X8$	68	32	$BП$	51
07	2	02	20	$PП$	53	33	04	04
08	:	13	21	38	38	34	$P \rightarrow XD$	6Г
09	$X \rightarrow PD$	4Г	22	$X \rightarrow P7$	47	35	$X \rightarrow P8$	48
10	$P \rightarrow X9$	69	23	$P \rightarrow XD$	6Г	36	$BП$	51
11	$P \rightarrow X8$	68	24	$PП$	53	37	04	04
12	—	11	25	38	38			

Инструкция. а X→ПА, б X→ПВ, в X→ПС В/О С/П. Переключатель в положении Р.

Контрольный пример. Вычислить с точностью  $\epsilon=10^{-3}$  корень уравнения  $x-\cos(x)=0$ , изолированный на сегменте  $[0; 1]$ .

Подпрограмма вычисления функции  $f(x)$  записывается в программную память ПМК начиная с команды 38. Запишем подпрограмму: БП 38 F ПРГ.

38	B↑	0E	40	—	11
39	F cos	1Г	41	B/O	52

FABT 0 X→ПА 1 X→ПВ ВП 3 /—/ X→ПС В/О С/П.

Перед началом вычислений переставьте переключатель в положение Р. Через 2 мин на индикаторе появляется результат 0,73876955.

Чтобы пояснить роль искусственно введенной метки  $m$  в организации цикла, воспользуемся алгоритмом:

```
алг ДИХ (вещ a, b, ε, x)
  арг a, b, ε
  рез x
нач цел m, вещ HC, KC, c
  m:=1; HC:=a; KC:=b; c:=(a+b)/2
  пока KC - HC ≥ ε ∧ m:=1
    нц
      если f(c) = 0
        то m:=0
        иначе если f(HC)*f(c) < 0
          то KC:=c
          иначе HC:=c
          все
          c:=(HC+KC)/2
        кон
      кон
    кон
```

Дело в том, что выход из цикла совершается в двух независимых случаях: когда разность координат концов сегмента меньше  $\epsilon$  или когда  $f(c)=0$ . Выход из цикла по первому условию уже хорошо нам знаком. Для выхода по условию  $f(c)=0$  (в программе он не предусмотрен) вводится промежуточная переменная  $m$ , которая играет роль метки со значениями 1 и 0. Если в процессе вычисления окажется  $f(c)=0$ , то  $m:=0$ , и этим обеспечен выход из цикла по второму условию.

Поскольку в УАЯ отсутствует оператор безусловного перехода, то этот метод используется для досрочного выхода из цикла.

Сравним алгоритм с программой. Нетрудно заметить, что одна команда учебного языка записывается целой серией операторов. В этом плане УАЯ ближе к человеческому. Алгоритм очень похож на исходный алгоритм в словесной форме.

В приведенной программе имеется определенная избыточность вычислений  $f(a)$  в каждом повторении цикла. Между тем знак на левом конце любого сегмента изоляции корня совпадает со знаком  $f(a)$ , и это вычисление можно каждый раз не повторять. Вместо двух обращений к подпрограмме в каждом цикле получаем одно, что значительно сокращает время решения.

### Программа ДИХОТОМИЯ, модификация

00	X→ПД	4Г	12	:	13	24	П→ХС	6С
01	F <sub>O</sub>	25	13	X→ПС	4С	25	X→ПА	4—
02	X→ПВ	4L	14	ПП	53	26	П→ХВ	6L
03	F <sub>O</sub>	25	15	35	35	27	П→ХА	6—
04	X→ПА	4—	16	П→X9	69	28	—	11
05	ПП	53	17	×	12	29	П→ХД	6Г
06	35	35	18	FX<0	5C	30	—	11
07	X→П9	49	19	24	24	31	FX<0	5C
08	П→ХА	6—	20	П→ХС	6C	32	08	08
09	П→ХВ	6L	21	X→ПВ	4L	33	П→ХС	6С
10	+	10	22	БП	51	34	C/P	50
11	2	02	23	26	26			

Инструкция. а B↑ б B↑ в B/O С/П.

### ЗАДАНИЯ

1. Постройте алгоритм «Дихотомия, модификация» на УАЯ.  
2. Во всех уравнениях, приведенных ниже, корень изолирован на сегменте  $[a; b]$ . Решите уравнения с точностью  $\epsilon=10^{-3}$ .

- 2.1.  $x^3-\cos \pi x=0$ ,  $a=0$ ,  $b=0,5$ ;
- 2.2.  $\cos \pi x/4-5x=0$ ,  $a=0$ ,  $b=1$ ;
- 2.3.  $x^2-\cos \pi x=0$ ,  $a=0$ ,  $b=0,5$ ;
- 2.4.  $x^3+2x-30=0$ ,  $a=2$ ,  $b=3$ ;
- 2.5.  $2x^2-\sin \pi x=0$ ,  $a=0,5$ ,  $b=1$ ;
- 2.6.  $x^6+4x-1=0$ ,  $a=0$ ,  $b=1$ ;
- 2.7.  $\sin x-x^3/3=0$ ,  $a=1$ ,  $b=2$ ;
- 2.8.  $\cos \pi x-x^2=0$ ,  $a=0$ ,  $b=0,5$ ;
- 2.9.  $\ln(x/3)+4x^3=0$ ,  $a=0,5$ ,  $b=1$ ;
- 2.10.  $\cos x-x^2=0$ ;  $a=0$ ,  $b=1$ .

При составлении подпрограмм старайтесь обходиться стеком, не прибегая для хранения промежуточных результатов к адресуемой памяти ПМК. Проверить решения помогут ответы:

- 2.1. 0,46728515
- 2.6. 0,24951171
- 2.2. 0,1977539
- 2.7. 1,4379882
- 2.3. 0,43798828
- 2.8. 0,43798828
- 2.4. 2,8930663
- 2.9. 0,7114258
- 2.5. 0,66162115
- 2.10. 0,8237305

3. В программе «Дихотомия» выходом из цикла управляет условие: длина очередного сегмента изоляции корня меньше  $\epsilon$ . Нетрудно заметить, что число повторений цикла  $n$ , необходимое для того, чтобы достичь заданной точности, должно удовлетворять соотношению  $(b-a)/2^n < \epsilon$ . Преобразуя это неравенство, получаем  $n > \ln((b-a)/\epsilon)/\ln 2$ . Число повторений цикла, необходимых для обеспечения заданной точности,  $N=[\ln((b-a)/\epsilon)/\ln 2]+1$ .

Составьте программу «Дихотомия», которая заканчивала бы вычисления по команде счетчика  $N$  повторений.

## ОТВЕТЫ

2.1.	38	FX <sup>2</sup>	22	41	FBX	0	44	F cos	1Г
39	FBX	0	42	F <sub>л</sub>	20	45	—	11	
40	×	12	43	×	12	46	B/O	52	
2.2.	38	B↑	0E	42	:	13	46	×	12
39	F <sub>π</sub>	20	43	F cos	1Г	47	—	11	
40	×	!2	44	↔	14	48	B/O	52	
41	4	04	45	5	05				
2.4.	38	FX <sup>2</sup>	22	42	2	02	46	0	00
39	FBX	0	43	×	12	47	—	11	
40	×	12	44	+	10	48	B/O	52	
41	FBX	0	45	3	03				
2.9.	38	B↑	0E	42	4	04	46	:	13
39	FX <sup>2</sup>	22	43	×	12	47	F ln	18	
40	FBX	0	44	↔	14	48	+	10	
41	×	12	45	3	03	49	B/O	52	

## 2.7. ОПЕРАТОР В/О В ОРГАНИЗАЦИИ ЦИКЛА. ВЛОЖЕННЫЕ ЦИКЛЫ

Мы обсудили, как можно организовать цикл с помощью счетчика, операторов безусловного и условного переходов. Рассмотрим теперь, как организовать цикл с помощью оператора В/О. В качестве тестовой рассмотрим задачу извлечения квадратного корня из  $a \geq 0$ . Построим программу вычисления членов последовательности  $a_{n+1} = (a/a_n + a_n)/2$ , где  $a$  — это заданное неотрицательное число,  $n=1, 2, \dots$ . Доказано, что вне зависимости от начального значения  $a_1$  выполняется равенство  $\lim_{n \rightarrow \infty} a_n = \sqrt{a}$ . Легко

видеть, что  $\sqrt{a}$  является решением уравнения  $x = (a/x + x)/2$ . Лучший способ проверить справедливость данного числового равенства и проследить динамику приближения к пределу — это численный эксперимент. Воспользуемся для этого программой «Квадратный корень»:

00	B↑	0E	03	+	10	06	C/P	50
01	:	13	04	2	02	07	B/O	52
02	FBX	0	05	:	13			

Инструкция  $a \leftarrow B \uparrow$

Контрольный пример. Вычислить  $\sqrt{251,78}$  с точностью до седьмого десятичного знака: 251.78  $\leftarrow B \uparrow$   $B \uparrow$  В/О С/П «126.39» С/П «64.191045» С/П «34.056699» С/П «20.724832» С/П «16.436771» С/П «15.877432» С/П «15.86758» С/П «15.867577» С/П «15.867577». Чтобы убедиться в том, что необходимая точность достигнута, найдем  $\sqrt{251,78}$  прямым вычислением: 251.78 F<sub>V</sub> «15.867577». Итак, задача решена за 9 повторений цикла.

## Структура программы

01—05: основа цикла вычислений  $(a/a_n + a_n)/2$  для  $n=1, 2, \dots$ . В качестве начального приближения корня  $a_1$  для упрощения программы выбрано подкоренное число  $a$ , оно же в

роли константы участвует в вычислении последующих приближений. Для сохранения  $a$  в стеке в качестве константы используются операторы B↑ B↑ B↑. Как уже отмечалось, число, находящееся в стековом регистре T, сохраняется в стеке при выполнении арифметических операций.

06 : останов для индикации.

07 : оператор В/О с помощью которого организован цикл.

Читателю уже известно, что свободная часть стека возврата заполнена адресом 01. Именно поэтому удалось использовать оператор В/О для передачи управления на начало программы — команду деления. Команда В/О заменила двухшаговую команду БП 01.

## ЗАДАНИЕ

Нетрудно убедиться, что результат вычислений не зависит от исходного выбора  $a_1$ . Однако число повторений цикла и, соответственно, время решения в значительной степени определяются значением  $a_1$ . Чем ближе  $a_1$  к значению корня, тем меньше требуется повторений цикла для достижения заданной точности.

Преобразуйте программу так, чтобы  $a_1$  можно было взять произвольным. Выясните, как связано число повторений цикла, обеспечивающее данную точность результата, с близостью  $a_1$  к значению корня.

Легко заметить, что программа «Квадратный корень» является частным случаем программы для извлечения корня  $k$ -й степени из  $a \geq 0$ , поскольку  $z_{n+1} = ((k-1)a_n + a/a_n^{k-1})/k$ .

## Программа КОРЕНЬ

00	X→ПА	4—	08	×	12	16	13	13
01	FO	25	09	FBX	0	17	+	10
02	X→ПВ	4L	10	X→П0	40	18	P→ХВ	6L
03	FO	25	11	FO	25	19	:	13
04	X→ПС	4C	12	П→ХА	6—	20	X→ПС	4C
05	П→ХВ	6L	13	П→ХС	6C	21	С/П	50
06	1	01	14	:	13	22	БП	51
07	—	11	15	FL0	5Г	23	05	05

Инструкция  $x_1 \leftarrow B \uparrow k$

Контрольный пример. Вычислить  $\sqrt[5]{7,93}$ ,  $x_1=1,5$ ;  $k=5$ ;  $a=7,93$ ; 1.5  $\leftarrow B \uparrow$  5  $B \uparrow$  7.93 В/О С/П «1.5132839» С/П «1.5130547» С/П «1.5130547» С/П «1.5130547». Прямое вычисление 5 F<sub>1/X</sub> 7.93 FX<sub>V</sub> дает тот же результат.

## Структура программы

05—08 : после ввода данных в регистры A, B, C, вычисление слагаемого  $(k-1)a_n$ .

09—10 : счетчик P0, рассчитанный на  $k-1$  повторений цикла для вычисления слагаемого  $a/a_n^{k-1}$ .

12—16 : вычисление этого слагаемого в цикле.

17—21 : вычисление и индикация следующего приближения корня.  
22, 23 : повторение цикла (команды безусловного перехода).

Таким образом, программа содержит два вложенных друг в друга цикла: внутренний, управляемый счетчиком (FL0), и внешний, организованный двухшаговой командой БП 05.

## 2.8. ЦИКЛЫ, ИСПОЛЬЗУЕМЫЕ ПРИ РЕШЕНИИ ЛОГИЧЕСКИХ ЗАДАЧ

Задача: АГАТ =  $(\text{АГ} + \text{АТ})^2$ . Здесь цифры зашифрованы буквами, каждой букве соответствует цифра, разным буквам — различные цифры. Восстановить числа.

Задачу в принципе можно решить полным перебором всех возможных вариантов. Так как букв всего три и каждая принимает одно из десяти значений: 0, 1, 2, ..., 9, то, учитывая, что  $A \neq 0$ , получаем  $10 \cdot 10 \cdot 9 = 900$  вариантов, т. е. всевозможных троек значений А, Г, Т. Рассмотренные тройки принимаются или отклоняются в зависимости от того, удовлетворяют или нет они условию.

Организовать такой перебор для ПМК легко. Однако в этом случае время решения по программе будет измеряться часами. Следовательно, человек должен уменьшить число допустимых вариантов до того, как составить программу. Рассмотрим, как это сделать на примере решения этой задачи.

Нетрудно заметить, что  $A < 5$ , в противном случае  $\text{АГ} + \text{АТ} > 100$ , и  $(\text{АГ} + \text{АТ})^2 = \text{АГАТ}$  пятизначно. Кроме того,  $A \neq 4$ , поскольку при  $A=4$  сумма  $\text{АГ} + \text{АТ} > 80$ ,  $(\text{АГ} + \text{АТ})^2 > 6400$ , а это противоречит тому, что слово АГАТ начинается с четверки. Далее, так как АГАТ — полный квадрат, то значение Т можно выбирать из множества 1, 4, 5, 6, 9. Значение Т = 0 мы исключаем: в этом случае А = 0. Подставляя в равенство допустимые значения А и Т, можно убедиться, что Г = 0, 4, 8. Теперь удается исключить как невозможные Т = 1 и Т = 9. Итак, вместо 900 вариантов нам придется рассматривать 27 возможных сочетаний: А = 1, 2, 3; Т = 4, 5, 6; Г = 0, 4, 8.

Человек, разумеется, может справиться с этой задачей и без ПМК, однако это потребует достаточно долгих рассуждений. С другой стороны, теперь ПМК очень быстро сможет перебрать все оставшиеся варианты.

Организовать перебор можно с помощью трех вложенных циклов. Во внешнем цикле изменяется значение Т в пределах выведенных ограничений. Исходное значение Т = 4. Во втором цикле меняется значение Г, начальное значение Г = 0. Наконец, во внутреннем цикле меняется значение А, начиная с А = 1. Для каждого допустимого набора значений А, Г, Т находятся числа АГАТ =  $= A \cdot 10^3 + G \cdot 10^2 + A \cdot 10 + T$  и  $(\text{АГ} + \text{АТ})^2 = (\text{A} \cdot 10 + \text{G} + \text{A} \cdot 10 + \text{T})^2$ . Если они равны, то одно из решений найдено. Перебор будет продолжаться до тех пор, пока не будут проверены все 27 вариантов.

## Программа ГОЛОВОЛОМКА

00	CX	0Г	19	P→X6	66	38	+	10
01	3	03	20	BП	0C	39	FBX	0
02	X→P0	40	21	3	03	40	P→X4	64
03	X→P4	44	22	P→X5	65	41	+	10
04	KП→X4	Г4	23	2	02	42	+	10
05	3	03	24	F10 <sup>x</sup>	15	43	FX <sup>2</sup>	22
06	X→P1	41	25	×	12	44	—	11
07	4	04	26	+	10	45	FX=0	5E
08	/—/	0L	27	P→X6	66	46	49	49
09	X→P5	45	28	BП	0C	47	FBX	0
10	P→X5	65	29	1	01	48	C/P	50
11	4	04	30	+	10	49	FL2	58
12	+	10	31	P→X4	64	50	18	18
13	X→P5	45	32	+	10	51	FL1	5L
14	3	03	33	P→X6	66	52	10	10
15	X→P2	42	34	BП	0C	53	FL0	5Г
16	CX	0Г	35	1	01	54	04	04
17	X→P6	46	36	P→X5	65	55	Fπ	20
18	KП→X6	Г6	37	↔	14	56	C/P	50

Инструкции. Ввод: В/О С/П.

Контрольный пример. В/О С/П (2 мин) «АГАТ-2025». Полный перебор с индикацией π занимает 5 мин. Других решений не обнаружено.

## Структура программы

01, 03, 04, 07—13, 16—18 : запись начальных значений Т, Г, А, равных соответственно 4, 0, 1;

00, 02, 05, 06, 14, 15 : формирование счетчиков в регистрах Р0—Р2 для циклов на три повторения;

19—32 : вычисление числа АГАТ для фиксированных значений переменных:

33—43 : вычисление числа  $(\text{АГ} + \text{АТ})^2$  для этих же значений;

44—48 : при их совпадении индицируется число АГАТ.

В любом случае программа продолжает поиск от внутреннего к внешнему циклу, пока индикация π не даст сигнал, что перебор завершен.

Роль некоторых команд важно пояснить особо. Оператор FBX восстанавливает прежнее содержимое РХ, копируя в РХ содержимое РХ1. В программе он использован дважды: в команде 47, где происходит вызов в РХ для индикации числа  $(\text{АГ} + \text{АТ})^2$ , найденного в предыдущем действии, и в команде 39 — в РХ вызывается число 10A, необходимое нам в программе дважды. При умножении на степень 10 использован оператор ВП. Его удалось применить при формировании чисел АГАТ, АГ, АТ. Лишь в первом цикле мы столкнулись со случаем Г = 0. Здесь пришлось отказаться от использования оператора ВП, решив задачу умножения Г на  $10^2$  более длинным путем (команды 23—25). На этот раз на долю человека, не считая, разумеется, составления программы, уже в процессе решения задачи приходилась наиболее интеллектуальная задача сокращения ансамбля вариантов, которые затем предлагаются ПМК для проверки на пригодность. Калькулятору поручена наименее интересная, утомительная для человека и малопродуктивная работа по доводке в систематическом переборе вариантов до нахождения однозначного результата.

Для упрощения программы в ней отсутствуют проверки условий несовпадения цифровых значений букв. В результате не исключена возможность того, что на индикатор будут выдаваться числа с повторяющимися цифрами для разных букв. Исключение подобных «ответов» также остается за человеком.

Так организовано разумное сочетание действий человека и вычислительного устройства в диалоговом режиме работы (этот вопрос будет подробно рассмотрен в ч. II книги).

## 2.9. ВЫБОРОЧНОЕ ОБРАЩЕНИЕ. ВЛОЖЕННЫЕ ПОДПРОГРАММЫ. ПОШАГОВОЕ ИСПОЛНЕНИЕ ПРОГРАММЫ

В сложных программах, рассчитанных на решение нескольких задач с общими входными данными, приходится в зависимости от задачи обращаться к той или иной серии команд. Такое обращение, выполненное в режиме вычислений с помощью команды БП, условимся называть выборочным обращением к программе.

Если несколько серий команд имеют общую часть, то ее целесообразно выделить в подпрограмму. Подпрограммы, в свою очередь, могут включаться одна в другую. Как уже отмечалось, ПМК позволяет организовать до пяти вложенных подпрограмм.

Задача решения треугольника — хорошая иллюстрация к сканному.

В треугольнике даны стороны  $a$  и  $b$  и угол  $C$  между ними. Построить программу для вычисления третьей стороны, углов  $A$  и  $B$ , площади  $S$ , радиусов описанного круга  $R$  и вписанного круга  $r$ . Стороны и углы треугольника согласованы, т. е. против стороны  $a$  лежит угол  $A$  и т. д. Расчетные формулы ( $a < b$ ):

$$1. c = \sqrt{a^2 + b^2 - 2ab \cos C}.$$

$$2. S = ab \sin C / 2.$$

Обе эти величины, выраженные непосредственно через данные, необходимы также и для других вычислений, их будем вычислять в подпрограммах.

3.  $A = \arcsin(a \sin C / c)$ . Через синус однозначно определяется меньший угол треугольника, так как он острый.

$$4. R = c / (2 \sin C).$$

$$5. r = S/p, \text{ где } p \text{ — полупериметр.}$$

При определении  $R$  и  $r$  используются результаты работы подпрограмм, вычисляющих  $c$  и  $S$ .

6.  $B = 180^\circ - A - C$ . Для вычисления  $B$  нужно предварительно найти  $A$ . Для этого нахождение  $A$  выделено в подпрограмму, которая, в свою очередь, использует подпрограмму вычисления  $c$ .

Программа включает три подпрограммы: вычисления  $c$  и  $S$  непосредственно по заданным значениям, а также  $A$ , которая, в свою очередь, опирается на подпрограмму вычисления  $c$ .

## Программа ТРЕУГОЛЬНИК

00	ПП	53	25	:	13	50	F sin	1С
01	57	57	26	С/П	50	51	×	12
02	С/П	50	27	ПП	53	52	ПП	53
03	ПП	53	28	39	39	53	57	57
04	48	48	29	ПП	53	54	:	13
05	С/П	50	30	57	57	55	F sin <sup>-1</sup>	19
06	ПП	53	31	П→ХА	6—	56	B/O	52
07	48	48	32	+	10	57	П→ХА	6—
08	П→ХС	6C	33	П→ХВ	6L	58	FX <sup>2</sup>	22
09	+	10	34	+	10	59	П→ХВ	6L
10	1	01	35	2	02	60	FX <sup>2</sup>	22
11	8	08	36	:	13	61	+	10
12	0	00	37	:	13	62	П→ХА	6—
13	↔	14	38	С/П	50	63	П→ХВ	6L
14	—	11	39	П→ХА	6—	64	×	12
15	С/П	50	40	П→ХВ	6L	65	П→ХС	6С
16	ПП	53	41	×	12	66	F cos	1Г
17	39	39	42	П→ХС	6C	67	×	12
18	С/П	50	43	F sin	1C	68	2	02
19	ПП	53	44	×	12	69	×	12
20	57	57	45	2	02	70	—	11
21	2	02	46	:	13	71	F V	21
22	:	13	47	B/O	52	72	B/O	52
23	П→ХС	6C	48	П→ХА	6—			
24	F sin	1C	49	П→ХС	6C			

**Инструкция.** *a X→ПА, b X→ПВ, c X→ПС.* Переключатель углов устанавливается в положение Р или Г в зависимости от того, в каких единицах задан угол. Выборочное обращение к программе производится командой БП a С/П, где a — адрес команды, с которой начинается решение данной задачи; 00 — вычисление стороны, 03 — угла A, 06 — угла B, 16 — площади S, 19 — радиуса R, 27 — радиуса r.

**Контрольный пример.**  $a=3,85$ ,  $b=4,74$ ,  $C=39,5^\circ$ . Вычислить  $c$ ,  $A$ ,  $B$ ,  $S$ ,  $R$ ,  $r$ . Вводим: 3,85 X→ПА, 4,74 X→ПВ, 39,5 X→ПС. Переключатель углов устанавливаем в положение Г. Вычислим сторону *c*: БП 00 С/П «3,0211494≈≈3,02»; угол *A*: БП 03 С/П «54,15322≈54,2»; угол *B*: БП 06 С/П «86,34678≈86,3»; площадь *S*: БП 16 С/П «5,803896≈5,80»; *R*: БП 19 С/П «2,3748252≈2,37»; *r*: БП 27 С/П «0,99971088≈1,00».

Анализируя программу, нетрудно заметить, что, за исключением регистров, занятых под данные величины, адресуемая память здесь не используется — вычисления производятся в стеке без записи промежуточных результатов. Сделано это специально, поскольку при использовании программы для решения треугольника в составе других программ память нужна для основной программы. В связи с этим приходится обращать внимание на то, чтобы при обращении к подпрограммам не терять промежуточных результатов из-за переполнения стека.

Нельзя не отметить, что стороны треугольника обычно формируются в процессе решения другой задачи и заранее не известны, поэтому на случай  $a > b$ , должна быть предусмотрена соответствующая перестановка.

Так программа с выборочным обращением позволяет вычислить независимо друг от друга любой элемент треугольника.

### Структура программы

00—02 : вычисление и индикация стороны треугольника  $c$ .  
03—05 : вычисление и индикация угла  $A$ .  
06—15 : вычисление и индикация угла  $B$ .  
16—18 : вычисление и индикация  $S$ .  
19—26 : вычисление и индикация радиуса описанного круга  $R$ .  
27—38 : вычисление и индикация радиуса вписанного круга  $r$ .  
39—47 : подпрограмма вычисления  $S$ .  
48—56 : подпрограмма вычисления  $A$ .  
57—72 : подпрограмма вычисления стороны  $c$ .

### Алгоритм ТРЕУГОЛЬНИК

```
алг ТС (вещ  $a_2, b_2, C_2, c$ )
  арг  $a_2, b_2, C_2$ 
  рез  $c$ 
нач
   $c := \sqrt{a_2^2 + b_2^2 - 2*a_2*b_2*\cos C_2}$ 
кон

алг У (вещ  $a_1, b_1, C_1, A$ )
  арг  $a_1, b_1, C_1$ 
  рез  $A$ 
нач вещ  $c_1$ 
  ТС ( $a_1, b_1, C_1, c_1$ )
   $A := \arcsin(a_1 * \sin C_1 / c_1)$ 
кон

алг П (вещ  $a_1, b_1, C_1, S$ )
  арг  $a_1, b_1, C_1$ 
  рез  $S$ 
нач
   $S := a_1 * b_1 * \sin C_1 / 2$ 
кон
```

```
алг ТРЕУГОЛЬНИК (вещ  $a, b, C, y$ , лит  $x$ )
  арг  $a, b, C, x$ 
  рез  $y$ 
нач вещ  $z$ 
  если  $x = \text{«}c\text{»}$ 
    то ТС ( $a, b, C, y$ )
  иначе если  $x = \text{«}A\text{»}$ 
    то  $y(a, b, C, y)$ 
  иначе если  $x = \text{«}B\text{»}$ 
    то  $y(a, b, C, y); y := 180 - C - y$ 
    иначе если  $x = \text{«}S\text{»}$ 
      то П ( $a, b, C, y$ )
    иначе если  $x = \text{«}R\text{»}$ 
      то ТС ( $a, b, C, y$ );  $y := y / \sin C$ 
    иначе П ( $a, b, C, y$ )
      ТС ( $a, b, C, z$ )
       $y := y / (a + b + z) / 2$ 
    все
  все
все
```

все

Для контроля вычислений, поиска ошибки, допущенной при наборе программы, или для сопоставления программы с алгоритмом, записанным на УАЯ, используется пошаговый режим вычислений. Каждый раз, когда мы нажимаем клавишу ПП, происходит выполнение очередного оператора программы, а на индикатор выводится содержимое РХ, которое находится там после выполнения данного оператора. Пусть, например, требуется выполнить команды, начиная с адреса 08, контролируя их по индикатору. Нажимаем БП 08. Если теперь нажать клавишу ПП, то на индикаторе окажется результат исполнения команды по адресу 08.

При исполнении алгоритма, записанного на УАЯ, режим пошагового исполнения программы позволяет вычислять промежуточные результаты и контролировать выполнение условий. Составим алгоритм и выполним в пошаговом режиме отдельные фрагменты программы «Треугольник».

Примеры.  $a = 3,85$ ,  $b = 4,74$ ,  $C = 39,5^\circ$ . Найдите  $S$ . БП 16 ПП. Содержимое РХ (в данном случае индикатора) не изменилось, поскольку выполнялся переход по адресу 39 к подпрограмме вычисления  $S$ . ПП «3.85» — это результат выполнения команды 39, ПП «4.74» — команды 40, ПП «18.249 = 3.85 · 4.74» — команды 41, ПП «39.5» ПП «0.63607825 = sin 39,5°», ПП «11.607792 = 18.249 sin 39,5°», ПП «2» и, наконец, ПП «5.803896 = 11.607792/2», т. е. площадь треугольника (команды 42—46).

Калькулятор продемонстрировал нам всю динамику вычислительного процесса, скрытую при автоматических вычислениях по программе. Сопоставляя ход расчета с алгоритмом, можно заметить, что выполнена была вычислительная часть вспомогательного алгоритма П, соответствующего команде П ( $a, b, C, y$ ). Для тех же значений  $a, b, C$  вычислим в пошаговом режиме сторону  $c$ :

В/О ПП ПП «3.85» ПП «14.8225» ПП «4.74» ПП «22.4676» ПП «37.2901» ПП «3.85» ПП «4.74» ПП «18.249» ПП «39.5» ПП «0.77162462» ПП «14.081378» ПП «2» ПП «28.162756» ПП «9.127344» ПП «3.0211494»

— искомая сторона  $c$ . В алгоритме данному фрагменту программы соответствует вспомогательный алгоритм ТС, отраженный командой ТС ( $a, b, C, y$ ).

### ЗАДАНИЕ

Составьте программы решения треугольника по стороне и двум углам, по трем сторонам. Решая задачу, не забудьте, что, зная три стороны, по теореме косинусов можно найти наибольший угол, тогда другие, меньшие углы, однозначно определяются теоремой синусов. Постройте также алгоритмы на УАЯ для обеих программ.

## 2.10. ЦИКЛЫ И ПОДПРОГРАММЫ В ПРОГРАММАХ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ

Рассмотрим известный метод Симпсона численного интегрирования, называемый также методом парабол. Программа основана на циклическом вычислении значений функции для равноотстоящих значений аргумента. Структура программы несколько сложнее предыдущих, поэтому в школе этот материал можно использовать на факультативных занятиях по информатике или в кружковой работе.

Промежуток интегрирования  $a, b$  разбивается на четное число равных частей ( $2n$ ) точками:  $a = x_0 < x_1 < x_2 < \dots < x_{2n} = b$ ; длина шага  $h = x_1 - x_0 = x_2 - x_1 = \dots = x_{2n} - x_{2n-1}$ .

Формула приближенного интегрирования по методу парабол:

$$J = \int_a^b f(x) dx \approx 2h/3((y_0 + y_{2n})/2 + 2(y_1 + y_3 + \dots + y_{2n-1}) + (y_2 + y_4 + y_{2n-2})).$$

В таком виде использовать формулу для программирования на ПМК неудобно, поэтому несколько преобразуем ее:

$$J = \int_a^b f(x) dx \approx 2h/3((y_0 - y_{2n})/2 + (2y_1 + y_2) + (2y_3 + y_4) + \dots + (2y_{2n-1} + y_{2n})).$$

Всего во внутренних скобках  $n$  слагаемых, причем их количество зависит от требуемой точности результата  $\epsilon$ : чем меньше  $\epsilon$ , тем их будет больше. Например, чтобы добиться точности четвертого десятичного знака ( $\epsilon = 10^{-4}$ ), нужно разбить сегмент на большее число частей, чем при  $\epsilon = 10^{-3}$ . Воспользуемся для нахождения  $n$  так называемым методом двойного пересчета. Суть его в следующем: пусть  $J$  — значение интеграла, вычисленное при разбиении  $n$ , т. е. на  $2n$  частей,  $J_2$  — при разбиении  $2n$ , т. е. на  $4n$  частей. Если  $|J_2 - J_1| < \epsilon$ , то  $J_2$  — это значение интеграла, вычисленное с точностью  $\epsilon$ . По существу, вычисления производятся дважды: для разбиений  $n$  и  $2n$ . Начальное значение  $n$  выбирается произвольно. Если условие окончания не выполнено, то  $n$  увеличивается на единицу и процедура повторяется.

Алгоритм решения этой задачи:

1. Задать начальное значение  $n$ , в нашей программе оно равно 3.
2. Вычислить  $h = (b-a)/(2n)$ ; в программе  $h = (b-a)/6$ .
3. Найти значения  $f(a)$  и  $f(b)$ , затем  $(f(a) - f(b))/2 = (y_0 - y_{2n})/2$ , которое принимается за начальное значение  $S$ .
4. Вычислить очередное значение  $x_i$  с нечетным значением индекса  $i$  как сумму предыдущего значения  $x_{i-1}$  и шага  $h$ . Начальное значение  $x_i = x_1 = x_0 + h = a + h$ .

5. Вычислить  $2y_i = 2f(x_i)$ .

6. Вычислить очередную абсциссу с четным индексом:  $x_{i+1} = x_i + h$ .

7. Вычислить  $y_{i+1} = f(x_{i+1})$ .

8. Полученные при исполнении указаний 5 и 7 результаты сложить, сумма прибавляется к  $S$ .

9. Если число просуммированных пар слагаемых равно  $n$ , то получаем результат умножением содержимого накопителя на  $2/3h$ , вычисление закончено. В противном случае — возврат к указанию 4.

Алгоритм предназначен для описания непосредственного вычисления интеграла без оценки полученного результата. Запишем алгоритм на УАЯ:

```

алг ИНТЕГРАЛ-1 (вещ a, b, ε, J)
    арг a, b, ε
    рез J
    нач
        nat n, i, j,вещ h, k,вещ таб A[1:2]
        k := (f(a) - f(b)) / 2; n := 1; A[1] := 0; A[2] := 1
        пока abs(A[2] - A[1]) ≥ ε
        нц
            n := n + 1; i := 1
            пока i ≤ 2
            нц
                n := n*2; h := (b - a)/n; A[i] := k; j := 1; c := a
                пока j ≤ n/2
                нц
                    c := c + h; A[i] := A[i] + 2*f(c)
                    c := c + h; A[i] := A[i] + f(c)
                    j := j + 1
                кц
                A[i] := A[i]*2/3; i := i + 1
            кц
            J := A[2]
        кон
    
```

Программа ИНТЕГРАЛ-1

### Распределение памяти

- P0 — число частей, на которые разбит сегмент  $[a, b]$ ;
- P1 — счетчик на два повторения цикла для вычисления  $J_2$  и  $J_1$ , выбираем число 2 в качестве начального значения;
- P4 —  $n, n+1, n+2, \dots$ ;
- P5 — очередное значение  $J_2$ ;
- P6 —  $f(b)$ ;
- P7 — очередное значение абсциссы:  $a, a+h, a+2h$  и т. д.;
- P8 — константа  $(f(a) - f(b))/2$ ;
- P9 —  $h = (b-a)/(2n)$  для  $J_1$  и  $h = (b-a)/(4n)$  для  $J_2$ ;
- PA —  $a$ ;
- PB —  $b$ ;
- PC —  $\epsilon$ .

00	X→P4	44	24	P→XB	6L	48	P→X9	69
01	P→XB	6L	25	P→XA	6—	49	X	12
02	PП	53	26	—	11	50	2	02
03	70	70	27	P→X0	60	51	X	12
04	X→P6	46	28	—	13	52	3	03
05	P→XA	6—	29	2	02	53	—	13
06	PП	53	30	—	13	54	P→X4	64
07	70	70	31	X→P9	49	55	FL1	5L
08	P→X6	66	32	P→XA	6—	56	19	19
09	—	11	33	X→P7	47	57	FO	25
10	2	02	34	PП	53	58	P→X5	65
11	—	13	35	66	66	59	—	11
12	X→P8	48	36	2	02	60	P→XC	6C
13	2	02	37	—	12	61	—	11
14	X→P1	41	38	X→P6	46	62	FX<0	5C
15	KП→X4	Г4	39	PП	53	63	13	13
16	P→X4	64	40	66	66	64	P→X5	65
17	2	02	41	P→X6	66	65	C/P	50
18	—	12	42	+	10	66	P→X7	67
19	X→P0	40	43	P→X2	62	67	P→X9	69
20	↔	14	44	+	10	68	+	10
21	X→P5	45	45	X→P2	42	69	X→P7	47
22	P→X8	68	46	FL0	5Г			
23	X→P2	42	47	34	34			

Инструкция. Вводится  $a$  X→PA  $b$  X→PB  $\epsilon$  X→PC  $n$  В/O С/П, индицируется приближенное значение интеграла с точностью  $\epsilon$ .

Контрольные примеры.  $\int_0^1 e^x dx$ . 0 X→PA 1 X→PB ВП 3 /—/ X→PC; вводим подпрограмму вычисления функции: БП 70 FПРГ Fe<sup>x</sup> В/O FABT 2 В/O С/П (1 мин 45 с) «1,7182822»;  $\int_1^5 \ln x dx$ . БП 70 FПРГ Fln В/O FABT 1 X→PA 2 X→PB ВП 3 /—/ X→PC 2 В/O С/П «0,38629373».

### Структура программы

00 : в Р4 запоминается  $n=2$ , а после команды 15  $n=3$ .  
 01—12 : вычисляется  $(f(a)-f(b))/2$  и направляется в Р8.  
 32—53 : во внутреннем цикле, управляемом оператором FL0, при фиксированном содержимом счетчика P1 вычисляется значение  $J_2$  или  $J_1$ .  
 19—56 : внешний по отношению к предыдущему цикл управляет ся командой FL1 и кроме внутреннего цикла включает вычисление очередного значения  $h$ .  
 13—65 : внешний цикл, относительно которого оба предыдущих цикла являются внутренними, управляетсся условием FX < <0. Кроме двух предыдущих циклов в него входит решение вопроса об окончании цикла и завершении вычислений. Для этого проверяется выполнение неравенства  $|J_2-J_1|<\epsilon$ . Если неравенство выполнено, то на индика-

тор выводится значение  $J_2$ , хранящееся в Р5. В противном случае  $n$  увеличивается на единицу, и процедура вычисления  $J_2$  и  $J_1$  повторяется.

### ЗАДАНИЯ

1. Вычислите с точностью  $\epsilon=10^{-3}$  следующие интегралы.

1.1.  $\int_0^{\pi/2} \cos x dx$  (2 мин) «1.0000016». Нужно выполнить следующие дейст-

вия: БП 70 FПРГ F cos В/O FABT. Затем 0 X→PA Fп 2 : X→PB  
1 ВП 3 /—/ X→PC 2 В/O С/П.

1.2.  $\int_0^1 e^{x^2} dx$  (1 мин 40 с) «1.462666».

1.3.  $\int_0^1 e^{-x^2} dx$  (2 мин) «0.74682453».

1.4.  $\int_2^5 dx/\ln x$  (4 мин) «2.5894627».

1.5.  $\int_0^1 \sqrt{\cos x} dx$  (1 мин 55 с) «0.94802473».

1.6.  $\int_0^1 \sqrt{1+x^4} dx$  (1 мин 45 с) «1.0894292».

1.7.  $\int_{\pi/4}^{\pi/3} (\sin x/x) dx$  (2 мин 10 с) «0.22648297».

1.8.  $\int_0^{\pi/2} \sqrt{1-0.1 \sin^2 x} dx$  (2 мин 45 с) «1.5397576».

2. Составьте программу приближенного интегрирования по методу трапеций. Расчетная формула:  $\int_a^b f(x) dx \approx h/2(y_0+y_n+2(y_1+y_2+\dots+y_{n-1}))$ . Эту фор-

мулу нетрудно привести к более удобному для расчетов виду:  $\int_a^b f(x) dx \approx$

$\approx h((y_0-y_n)/2 + (y_1+y_2+\dots+y_{n-1}))$ .

Структура программы может быть такой же, как программы «Интеграл-1». Однако, исключив команды 34—38, 41, 42, можно существенно упростить внутренний цикл. Постарайтесь составить программу самостоятельно и лишь после этого сверьте ее с программой, приведенной ниже.

Инструкция. Ввод: БП 57 FПРГ. Набираем текст подпрограммы вычисления подынтегральной функции: В/O FABT. Исходные данные,  $a$  X→PA,  $b$  X→PB,  $\epsilon$  X→PC 10 В/O С/П. Индицируется значение интеграла, вычисленное с точностью  $\epsilon$ .  $n=10$  — параметр, необходимый для оценки точности методом двойного пересчета.

3. Составьте программу приближенного интегрирования методом прямого угольников. Расчетная формула:  $\int_a^b f(x) dx \approx h(y_1+y_2+\dots+y_n)$ .

4. Решая задачи, мы для оценки точности результата до сих пор пользовались правилами двойного пересчета. Это правило, основанное на двукратном решении задачи для каждого значения  $n$ , почти вдвое увеличивает время решения по программе, не говоря уже об усложнении самой программы.

00	X→P4	44	19	X→P0	40	38	P→X2	62
01	P→XB	6L	20	↔	14	39	+	10
02	ПП	53	21	X→P5	45	40	X→P2	42
03	57	57	22	P→X8	68	41	FLO	5Г
04	X→P6	46	23	X→P2	42	42	32	32
05	P→XA	6—	24	P→XB	6L	43	P→X9	69
06	ПП	53	25	P→XA	6—	44	×	12
07	57	57	26	—	11	45	P→X4	64
08	P→X6	66	27	P→X0	60	46	FL1	5L
09	—	11	28	:	13	47	19	19
10	2	02	29	X→P9	49	48	FO	25
11	:	13	30	P→XA	6—	49	P→X5	65
12	X→P8	48	31	X→P7	47	50	—	11
13	2	02	32	P→X7	67	51	P→XC	6C
14	X→P1	41	33	P→X9	69	52	—	11
15	КП→Х4	Г4	34	+	10	53	FX<0	5G
16	P→X4	64	35	X→P7	47	54	13	13
17	2	02	36	PП	53	55	P→X5	65
18	X	12	37	57	57	56	C/P	50

В математическом анализе известны более эффективные методы оценки погрешности интегрирования. Обозначив ее буквой  $R$ , приведем оценочные формулы: для метода парабол  $R \leq M_3(b-a)^4/(192n^3)$ , для метода трапеций  $R \leq M_2(b-a)^3/(12n^2)$ , для метода прямоугольников  $R \leq M_1(b-a)^2/(2n)$ . Здесь  $M_3$ ,  $M_2$ ,  $M_1$  — значения максимума модуля третьей, второй, первой производной интегрируемой функции на сегменте  $[a, b]$ .

С помощью этих формул вычисляется  $n$ , необходимое для достижения требуемой точности результата. Покажем, как это делается в методе парабол.

$M_3(b-a)^4/(192n^3) < \epsilon$ , отсюда  $n > \sqrt[3]{M_3(b-a)^4/(192\epsilon)}$ . За значение  $n$  берется  $\lceil \sqrt[3]{M_3(b-a)^4/(192\epsilon)} \rceil + 1$ . Отыскав значение корня, можно найти  $n$  по программе и сформировать счетчик в Р0, необходимый для вычисления интеграла; X→P4 КП→X4 P→X4 X→P0. Использование Р4 одновременно добавляет к вычисленному значению корня единицу и отделяет целую часть результата. Аналогично, хотя и по другим формулам, отыскивается  $n$  в методах трапеций и прямоугольников.

Итак, задание — пользуясь описанной процедурой, составьте программы приближенного интегрирования в каждом из перечисленных методов. Решение можно упростить по сравнению с предыдущим. Вначале вычисляется  $n$ , обеспечивающее требуемую точность, затем в Р0 организуется счетчик, рассчитанный на  $n$  слагаемых. Задача будет решена к моменту завершения цикла. Исчезла, таким образом, необходимость в повторном вычислении.

Метод прямой оценки требует значительных дополнительных усилий, поскольку значения  $M$  вычисляются вне программы. Этим и объясняется сравнительно редкое его использование.

5. Составьте алгоритмы для методов трапеций и прямоугольников и запишите их на УАЯ. Алгоритм интегрирования методом прямоугольников приведен в [12], однако интегрирование выполняется для заданного  $n$ . Постройте алгоритм, включающий процедуру оценки либо двойным пересчетом, либо с помощью соответствующей формулы.

## В МИРЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

### 3.1. ПРЕДЕЛ ПОСЛЕДОВАТЕЛЬНОСТИ

Последовательность принято обозначать так:  $a_1, a_2, \dots, a_n, \dots$ . Здесь  $a_1$  — первый член;  $a_n$  — общий член;  $n$  — номер общего члена в последовательности. Таким образом, номер является значением аргумента, значением функции можно считать соответствующий член последовательности. В связи с этим последовательность рассматривают как функцию натурального аргумента  $a_n = f(n)$ , например  $a_n = 1/n$ . Поскольку множество натуральных чисел бесконечно, бесконечно и множество членов любой последовательности.

Если члены последовательности в процессе возрастания аргумента (т. е. номера члена) приближаются к некоторому числу, то говорят, что последовательность сходится к своему пределу, понимая под пределом это число. В противном случае последовательность называется расходящейся.

С помощью ПМК можно поставить простейший численный эксперимент, который наглядно демонстрирует процесс изменения членов последовательности. Наблюдая за вычислениями, нетрудно определить, существует ли предел данной последовательности.

Рассмотрим последовательность  $a_n = n(e^{1/n} - 1)$ . Здесь  $n$  — число определяемых членов последовательности; через  $k$  обозначим длину шага — интервал, через который выводятся члены. Предлагаем следующий алгоритм вычисления:

1. Вычислить  $a_k$ .
2. Увеличить индекс при  $a$  на длину шага  $k$ .
3. Вычислить соответствующий новому индексу член последовательности.
4. Если число найденных членов не превышает  $n$ , перейти к указанию 2, в противном случае к указанию 5.
5. Конец.

#### Распределение памяти

P0 — счетчик количества вычисленных членов последовательности, начальное значение  $n$ ;  
 P4 — номер очередного индицируемого члена последовательности;  
 P5 — константа  $k$  — длина шага, прибавляемого к предыдущему номеру для получения последующего. Соответственно индицируются члены  $a_k$ ,  $a_{2k}$ ,  $a_{3k}$ , ... .

00	X→P5	45	07	+	10	14	↔	14
01	↔	14	08	X→P4	44	15	:	13
02	X→P0	40	09	F1/X	23	16	C/P	50
03	0	00	10	B↑	0E	17	FL0	5Г
04	X→P4	44	11	Fe <sup>x</sup>	16	18	05	05
05	P→X4	64	12	1	01	19	Fπ	20
06	P→X5	65	13	—	11	20	C/P	50

Инструкция:  $n$   $B\uparrow$   $k$   $B/O$   $C/P$ .

### Структура программы

- 00—04 : данные, введенные в стек, размещаются по регистрам памяти:  $k$  — в Р5,  $n$  — в Р0, 0 — в Р4.  
 05—08 : формирование в Р4 номера очередного члена последовательности.  
 09—16 : расчет и индикация очередного члена последовательности.  
 17 : проверка окончания с помощью счетчика Р0.  
 18 : возвращение к началу цикла, если необходимое количество членов последовательности еще не вычислено.  
 19, 20 : формирование сигнала об окончании расчетов — числа π.

Несложно заметить, что если выделить команды 16—23 в подпрограмму, то, меняя ее, можно будет вычислять члены произвольных последовательностей. Остальная часть программы остается неизменной. Полученную программу условимся называть универсальной.

### Программа универсальная

00	X→P5	45	08	X→P4	44	16	F1/X	23
01	↔	14	09	P/P	53	17	B↑	0E
02	X→P0	40	10	16	16	18	Fe <sup>x</sup>	16
03	0	00	11	C/P	50	19	1	01
04	X→P4	44	12	FL0	5Г	20	—	11
05	P→X4	64	13	05	05	21	↔	14
06	P→X5	65	14	Fπ	20	22	:	13
07	+	10	15	C/P	50	23	B/O	52

Инструкция:  $n$   $B\uparrow$   $k$   $B/O$   $C/P$ .

Располагая универсальной программой, можно провести числовой эксперимент.

1. Пусть  $n=5$ ,  $k=1000$ . Вводим 5  $B\uparrow$  1000  $B/O$   $C/P$  « $a_{1000}=1,0005$ »,  $C/P$  « $a_{2000}=1,0002$ »,  $C/P$  « $a_{3000}=1,0002$ »,  $C/P$  « $a_{4000}=1$ »,  $C/P$  « $a_{5000}=1$ »,  $C/P$  «π». Можно предположить, что последовательность  $a_n=n(e^{1/n}-1)$  приближается к единице. Перейдем к детальному анализу, выбрав минимальный шаг  $k=1$ .
2.  $n=10$ ,  $k=1$ . Вводим 10  $B\uparrow$  1  $B/O$   $C/P$   $a_1=1,7182818$ ,  $a_2=1,2974426$ ,  $a_3=1,1868372$ ,  $a_4=1,1361016$ ,  $a_5=1,107014$ ,  $a_6=$

$$=1,0881624, \quad a_7=1,074955, \quad a_8=1,065188, \quad a_9=1,0576719, \quad a_{10}=1,05171.$$

Обратим внимание: приближение к единице происходит медленно, но неуклонно. Чтобы в этом убедиться, достаточно проанализировать разности между единицей и членами последовательности. Начиная с первого члена, элементы последовательности отличаются по модулю от единицы менее чем на 0,8, начиная со второго члена — менее чем на 0,3 и т. д. Если взять произвольное сколь угодно малое число, то ясно, что с некоторого момента, т. е. достаточно большого номера, разность между единицей и членами последовательности станет по модулю меньше этого числа.

Теперь нетрудно определить понятие предела последовательности.

Число  $a$  называется пределом последовательности  $a_n$ , если для любого сколь угодно малого  $\varepsilon > 0$  существует такой номер  $N$ , что для всех  $n > N$  выполняется неравенство  $|a_n - a| < \varepsilon$ .

Можно поэкспериментировать с той же последовательностью, увеличив шаг сначала до 10: 10  $B\uparrow$  10  $B/O$   $C/P$ , затем до 100: 10  $B\uparrow$  100  $B/O$   $C/P$  и т. д.

В решении задачи роль вычислителя, взявшего на себя все рутинные работы, играет микрокалькулятор. Человеку остается лишь осмыслить полученные результаты. Налицо, таким образом, взаимовыгодный диалог.

### ЗАДАНИЯ

1. Используя универсальную программу, позэкспериментируйте с последовательностями: 1)  $a_n=n\sin(1/n)$ , 2)  $a_n=(1+1/n)^n$ , 3)  $a_n=\sin n/n$ . Определите значение предела грубой прикидкой, затем подтвердите решение «точнойстройкой», выбирая различные значения  $n$ .

2. Покажите, что последовательность  $a_n=\sin(\pi n/2)$  расходится. Подпрограмма каждый раз записывается начиная с адреса 16. Последней командой подпрограммы, как всегда, будет  $B/O$ .

### ОТВЕТЫ

1. 1.	16	F1/X	23	1. 2.	16	B↑	0E	2.	16	9	09
	17	Fsin	1C		17	F1/X	23		17	0	00
	18	FBX	0		18	1	01		18	×	12
	19	:	13		19	+	10		19	Fsin	1C
	20	B/O	52		20	FX <sup>y</sup>	24		20	B/O	52
						21	B/O				

Переключатель  
в положении Р  
Последовательность  
сходится к 1

Последовательность  
сходится к  $e=2,71828...$

Переключатель  
в положении Г  
Последовательность  
расходится

## 3.2. РЕКУРРЕНТНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ И ЧИСЛО $\pi$

Известно, что  $\pi$  — это иррациональное число, а значит, оно выражается бесконечной десятичной непериодической дробью  $3,141592653589793\dots$ . Калькулятор позволяет заглянуть в процесс формирования числа  $\pi$ .

Мы знаем, что  $\pi$  есть отношение длины окружности к ее диаметру:  $C/D$ . Поскольку  $C = \lim_{n \rightarrow \infty} P_n$ , где  $P_n$  — последовательность периметров правильных многоугольников, вписанных в окружность, то удобно, взяв за исходный некоторый периметр  $P_k$ , строить периметры многоугольников с удваивающимся числом сторон  $P_{2k}, P_{4k}, P_{8k}, \dots$ . Для этого воспользуемся известной формулой удвоения

$$a_{2k} = \sqrt{2R^2 - 2R\sqrt{R^2 - a_k^2}/4}.$$

Выберем произвольное, но известное значение  $a_{2k}$ , например  $a_6 = R = 1$  — сторона правильного шестиугольника, вписанного в окружность единичного радиуса. В качестве первого приближения  $\pi$  вычисляется  $ka_{2k}$ , в примере  $3 \cdot 1 = 3$ . Постепенно строится последовательность  $ka_{2k}, 2ka_{4k}, 4ka_{8k}, \dots$ , генерируемая на основе первого члена  $a_{2k}$ .

В отличие от всех последовательностей, рассмотренных ранее, последовательность периметров обладает следующей особенностью: для выражения любого ее члена необходимо знать предшествующий. Такой способ, когда вычисление последующего члена сводится к вычислению одного или нескольких предыдущих, называется рекуррентным.

Простейший пример рекуррентной последовательности — арифметическая прогрессия: каждый член начиная со второго, равен предыдущему, сложенному с разностью прогрессии.

### Распределение памяти

$P_0$  —  $n$  — число членов последовательности, которые мы собираемся вычислить;  $P_3 = k$ ,  $2k$ ,  $4k, \dots$  — порядковые номера членов последовательности;  $P_4 = a_{2k}, a_{4k}, a_{8k}, \dots$  — члены последовательности.

00	X→P4	44	11	Fп	20	22	2	02
01	FО	25	12	C/П	50	23	×	12
02	X→P3	43	13		1	01	24	FV/-
03	FО	25	14	B↑	0E	25	X→P4	44
04	X→P0	40	15	П→X4	64	26	П→X3	63
05	П→X4	64	16	FX <sup>2</sup>	22	27	2	02
06	П→X3	63	17		4	04	28	×
07	×	12	18	:	13	29	X→P3	43
08	C/П	50	19		—	11	БП	51
09	FL0	5Г	20	FV/-	21	31	05	05
10		13	13		—	11		

Инструкция.  $n$  В↑  $k$  В↑  $a_{2k}$  В/О С/П.  
Контрольные примеры.

1. Вписанный в единичную окружность правильный шестиугольник: 6 В↑ 3 В↑ 1 В/О С/П «3» С/П «3,1058285» С/П «3,1326297» С/П «3,1393612» С/П «3,141049» С/П «3,1414157» С/П «π».

Мы намеренно прекратили вычисления: из-за быстрого убывания длины стороны многоугольника накопленная погрешность вычислений начинает превышать значения членов последовательности.

2. Вписанный пятиугольник: 6 В↑ 2,5 В↑ 5 В↑ FV — 2 : FV B/О С/П «2,938926» С/П «3,0901699» С/П «3,1286898» С/П «3,1383562» С/П «3,1408024» С/П 3,1414646 С/П «π».

Рассмотренные примеры демонстрируют правило, что последовательность если она сходится, имеет единственный предел, который не зависит от начального приближения.

### Структура программы

- 00—04 : ввод начальных данных.
- 05—08 : индикация очередного приближения  $\pi$ , для продолжения счета нажать клавишу С/П.
- 09, 10 : проверка окончания вычислений.
- 11, 12 : индикация  $\pi$ , означающая завершение решения.
- 13—25 : вычисление стороны правильного многоугольника с удвоенным по сравнению с предыдущим числом сторон.
- 26—29 : очередное удвоение числа сторон многоугольника.
- 30, 31 : возврат к началу цикла.

### ЗАДАНИЯ

Постройте последовательности приближения, начав вычисления для четырехугольника  $n=6, k=2, a_{2k}=a_4 = \sqrt{2}$  и десятиугольника  $n=6, k=5, a_{2k}=a_{10} = (\sqrt{5}-1)/2$ .

Вычисления числа  $\pi$  можно ускорить, если воспользоваться методом известного математика Христиана Гюйгенса. Гюйгенс заметил, что число  $\pi$  независимо от  $n$  находится в левой трети интервала  $[p_n, q_n]$ , где  $p_n$  — полупериметр вписанного;  $q_n$  — полупериметр описанного  $n$ -угольника. В программе, основанной на использовании данного результата, в качестве начального приближения  $\pi$  выбирается середина левой трети интервала:  $\pi \approx (5p_n + q_n)/6$ . Гюйгенс обратил внимание на то, что при неограниченном возрастании  $n$  число  $\pi$  приближается к правому концу первой трети интервала  $[p_n, q_n]$ , и выбрал новое приближение  $\pi \approx (2p_n + q_n)/3$ . Этой формуле соответствует программа «Гюйгенс-2». Получена и оценочная формула — требуемая точность результата  $\pi$  связана с числом сторон вписанного в единичную окружность правильного многоугольника  $n > \sqrt[4]{64/\epsilon}$ . За  $n$  принимается целая часть корня, увеличенная на единицу. Так, для  $\epsilon = 10^{-7}$  получаем  $n=160$ , т. е. при получении правильного вписанного 160-угольника вычисление  $\pi$  можно прекратить, погрешность не превышает  $10^{-7}$ . Теперь несложно подсчитать  $m$  — необходимое число повторений вычисления периметров  $p_n$  и  $q_n$ , соответствующее требуемой точности результата:  $3 \cdot 2^{m-1} > n, m \geq \lg(n/3)/\lg 2 + 1$ . Так же, как в случае с  $n$ , берем целую часть этого выражения, увеличенную на единицу.

Сравните точность программ «Гюйгенс-1» и «Гюйгенс-2» и скорость вычисления, которые должны различаться, так как для программ выбраны различные приближения.

## Программа ГЮЙГЕНС-1

00	3	03	11	X→ПА	4—	22	X→ПС	4С
01	X→ПС	4C	12	П→ХД	6Г	23	БП	51
02	1	01	13	Ftg	1E	24	02	02
03	8	08	14	П→ХС	6С	25	П→ХА	6—
04	0	00	15	×	12	26	5	05
05	П→ХС	6C	16	X→ПВ	4L	27	×	12
06	:	13	17	ПП	53	28	+	10
07	X→ПД	4Г	18	25	25	29	6	06
08	Fsin	1C	19	П→ХС	6C	30	:	13
09	П→ХС	6C	20	2	02	31	С/П	50
10	X	12	21	×	12	32	B/O	52

Инструкция СХ В/О С/П. Переключатель в положении Г.

## Программа ГЮЙГЕНС-2

25	П→ХА	6—	28	+	10	31	С/П	50
26	2	02	29	3	03	32	B/O	52
27	X	12	30	:	13			

Эта программа отличается лишь подпрограммой реализующей формулу приближения.

Оставляя разбор программ для самостоятельного анализа читателю, перейдем к описанию программы «Гюйгенса-3». Она отличается тем, что вычисление  $\pi$  производится с предварительной оценкой числа повторений цикла, необходимых для получения результата с наперед заданной точностью.

## Программа ГЮЙГЕНС-3

00	П→Х9	69	19	П→Х4	64	38	ПП	53
01	6	06	20	X→П0	40	39	48	48
02	4	04	21	3	03	40	П→ХС	6C
03	↔	14	22	X→ПС	4C	41	2	02
04	:	13	23	1	01	42	×	12
05	F V—	21	24	8	08	43	X→ПС	4C
06	F V—	21	25	0	00	44	FL0	5Г
07	X→П4	44	26	П→ХС	6C	45	23	23
08	KП→Х4	Г4	27	:	13	46	Fп	20
09	П→Х4	64	28	X→ПД	4Г	47	С/П	50
10	3	03	29	F sin	1C	48	П→ХА	6—
11	:	13	30	П→ХС	6C	49	2	02
12	F lg	17	31	×	12	50	×	12
13	2	02	32	X→ПА	4—	51	+	10
14	F lg	17	33	П→ХД	6Г	52	3	03
15	:	13	34	Ftg	1E	53	:	13
16	X→П4	44	35	П→ХС	6C	54	С/П	50
17	KП→Х4	Г4	36	×	12	55	B/O	52
18	KП→Х4	Г4	37	X→ПВ	4L			

Инструкция: ε X→П9 В/О С/П; переключатель углов установите в положение Г.

Располагая тремя программами, можно поставить численный эксперимент (табл. 4).

Таблица 4

Гюйгенс-1	Гюйгенс-2	Гюйгенс-3	Гюйгенс-3, модифицированная
3,0310890	3,4641016	3,4641016	3,4641016
3,0773505	3,1547006	3,1547006	3,1423489
3,1240886	3,1423489	3,1423489	3,1415955
3,1371338	3,1416389	3,1416389	3,1415924
3,1404728	3,1415955	3,1415955	
3,1413125	3,1415928	3,1415928	
3,1415225	3,1415924	3,1415924	
3,1415753			
3,1415881			
3,1415915			
3,1415920			

Четвертый столбец таблицы получен после того, как мы заменили в командах 13 и 41 число 2 на 4. Вычисления в таблице выполнены для  $\varepsilon = 10^{-7}$ , погрешность в седьмом знаке появилась за счет приближенных вычислений.

## ЗАДАНИЕ

Постройте алгоритмы для программ «Гюйгенс-1» и «Гюйгенс-2» на УАЯ Помощь в работе окажет алгоритм программы «Гюйгенс-3»:

алг ГЮЙГЕНС-3 (нат  $k$ ,  $l$ . веществ  $\varepsilon$ ,  $A$ )

арг  $\varepsilon$ ,  $k$ ,  $l$

рез  $A$

нач нат  $i$ , цел  $n$ ,  $m$ , веществ  $p$ ,  $q$ ,  $B$

$n := \sqrt[4]{64/\varepsilon} + 1$ ;  $m := \lg(n/k)/\lg l + 2$

$i := 1$

пока  $i \leq m$

нц

$p := k * \sin(180/k)$ ;  $q := k * \operatorname{tg}(180/k)$ ;  $B := (2*p + q)/3$

выв  $B$

$k := k * l$ ;  $i := i + 1$

кц

$A := B$

кон

Здесь  $\varepsilon$  — допустимая погрешность результата;  $A$  — значение  $\pi$  с точностью  $\varepsilon$ ;  $k$  — число сторон в текущем многоугольнике, в программах начальное значение  $k=3$ ;  $l$  — постоянное отношение числа сторон в последующем многоугольнике к числу сторон в предыдущем, в программах  $l=2$ ;  $p$  и  $q$  — периметры вписанного и описанного многоугольников;  $B$  — выводимое приближение  $\pi$  на данном этапе вычисления;  $n$  — число сторон в многоугольнике для достижения значения  $\pi$  с точностью  $\varepsilon$ ;  $m$  — число элементов последовательности многоугольников для вычисления значения  $\pi$  с данной точностью.

### 3.3. ПОСЛЕДОВАТЕЛЬНОСТИ, ГЕНЕРИРУЮЩИЕ $\pi$ И НЕ СВЯЗАННЫЕ С ОКРУЖНОСТЬЮ

Рассмотрим последовательность  $a_1, a_2, \dots$ . На ее основе построим новую последовательность  $s_1, s_2, \dots$ , где каждый член определяется следующим образом:  $s_1 = a_1, s_2 = a_1 + a_2, \dots, s_n = a_1 + a_2 + \dots + a_n$ . Такая последовательность называется последовательностью частичных сумм исходной последовательности  $a_n$ .

Для последовательности  $a_n = 1/n^2$  известным математиком Эйлером получен следующий результат:  $\lim_{n \rightarrow \infty} s_n = \pi^2/6$ , отсюда  $\pi = \sqrt[4]{6 \lim_{n \rightarrow \infty} s_n}$ . Воспользуемся этой последовательностью для вычисления  $\pi$ . Программу будем составлять так, чтобы она была пригодна для любой последовательности подобного типа.

#### Программа ПРИБЛИЖЕНИЕ-1

Распределение памяти  
 Р0 —  $k$  — длина шага индикации очередного члена последовательности  $s$ ; индицируются приближения  $\pi$ , соответствующие  $s_k, s_{2k}, s_{3k}, \dots$ ;  
 Р4 — исходное значение — нуль, затем содержимое последовательно увеличивается каждый раз на единицу;  
 РС — сумматор для формирования очередного значения  $s_k$ , исходное значение равно нулю.

00	X→П1	41	08	ПП	53	16	+	10	
01	0	00	09		13	13	17	X→ПС	4C
02	X→ПС	4C	10	C/П	50	18	FL0	5Г	
03	X→П4	44	11	БП	51	19	06	06	
04	П→Х1	61	12	04	04	20	6	06	
05	X→П0	40	13	FX <sup>2</sup>	22	21	×	12	
06	KП→Х4	Г4	14	F1/X	23	22	FLV <sup>-</sup>	21	
07	П→Х4	64	15	П→ХС	6C	23	B/O	52	

#### Инструкция. $k$ B/O C/П.

Контрольный пример.  $k=90$ . Вычислим приближения  $\pi$ , соответствующие  $s_{90}, s_{180}, s_{270}, s_{360}, \dots, s_{630}$ . Вводим 90 B/O C/П «3,1310238», C/П «3,1362987», C/П «...», C/П «s<sub>630</sub>=3,1400809»...

Для совпадения только трех первых знаков потребовалось 30 мин и вычисление 630 членов последовательности. Процесс сходится медленно, и скорость сходимости нельзя изменить. Например, для получения  $s_{630}$  необходимо вычислить все значения  $a_n$  от 1-го до 630-го и просуммировать их. Попробуем отыскать последовательности частичных сумм, которые обеспечили бы более быструю сходимость.

#### Структура программы

- 00—03 : ввод в память начальных значений переменных.  
 04, 05 : ввод в счетчик Р0 значения  $k$ .  
 06 : получение очередного значения  $n$  добавлением единицы к предыдущему значению.

07—09, 13—14 : вычисление  $a_n$ .

15—17 : подсчет  $s_n = s_{n-1} + a_n$  в сумматоре РС.

18 : проверка завершения вычисления очередной суммы.

20—22 : вычисление очередного приближения  $\pi$ .

Для последовательности  $a_n = 1/n^4$  из математического анализа известно, что  $\lim_{n \rightarrow \infty} s_n = \pi^4/90$ , отсюда  $\pi = \sqrt[4]{90 \lim_{n \rightarrow \infty} s_n}$ . Чтобы поставить вычислительный эксперимент, воспользуемся универсальной частью программы «Приближение-1».

#### Программа ПРИБЛИЖЕНИЕ-2

13	FX <sup>2</sup>	22	18	X→ПС	4C	23	X	12
14	FX <sup>2</sup>	22	19	FL0	5Г	24	FLV <sup>-</sup>	21
15	F1/X	23	20	06	06	25	FLV <sup>-</sup>	21
16	П→ХС	6C	21	9	09	26	B/O	52
17	+	10	22	0	00			

Контрольный пример. Как и в прошлый раз, выберем шаг  $k=90$ . В/О С/П (4 мин) «3,1415923». Получено решение с точностью до шестого десятичного знака!

Продолжим эксперимент, воспользовавшись последовательностью  $a_n = (-1)^{n+1}/n^2$ , для которой  $\lim_{n \rightarrow \infty} s_n = 1/12\pi^2$ , т. е.  $\pi = \sqrt[4]{12 \lim_{n \rightarrow \infty} s_n}$ .

#### Программа ПРИБЛИЖЕНИЕ-3

13	B↑	0E	20	/ — /	0L	27	1	01
14	FX <sup>2</sup>	22	21	×	12	28	2	02
15	F1/X	23	22	П→ХС	6C	29	FLV <sup>-</sup>	12
16	↔	14	23	+	10	30	FLV <sup>-</sup>	21
17	Fπ	20	24	X→ПС	4C	31	B/O	52
18	×	12	25	FL0	5Г			
19	F cos	1Г	26	06	06			

Инструкция. Первые команды из программы «Приближение-1». Переключатель углов в положении Р. Команды с 13 по 21 вычисляют  $a_n$ . Основная трудность — вычисление множителя, который задает знак, т. е.  $(-1)^{n+1}$ . Для этой цели функция  $x^n$  непригодна: при  $x < 0$  ПМК будет выдавать сообщение об ошибке ЕГГОГ. В программе для определения знака мы использовали функцию косинус, поскольку  $\cos n\pi = 1$ , если  $n$  четно, и  $\cos n\pi = -1$ , если  $n$  нечетно.

Контрольный пример.  $k=90$ . 90 В/О С/П (7 мин) «3,1414761». т. е. получаем точность всего три десятичных знака, а значит, можно сделать вывод, что выбранная последовательность сходится медленнее других.

#### ЗАДАНИЕ

Определить динамику приближения к  $\pi$ , выяснить монотонно или нет сходится последовательность, найти результат с точностью до шести десятичных знаков.

В заключение изучим поведение последовательности  $a_n = (-1)^{n+1}(2n-1)/3^{n-1}$ , для которой  $\pi = 2 \sqrt{3} \lim_{n \rightarrow \infty} s_n$ .

#### Программа ПРИБЛИЖЕНИЕ-4

13	1	01	22	$\Pi \rightarrow X_4$	64	31	FL0	5Г
14	-	11	23	F $\pi$	20	32	06	06
15	3	03	24	X	12	33	2	02
16	FX <sup>y</sup>	24	25	F cos	1Г	34	X	12
17	↔	14	26	/ - /	0L	35	3	03
18	$\Pi \rightarrow X_4$	64	27	X	12	36	F V-	21
19	+	10	28	$\Pi \rightarrow X_C$	6C	37	X	12
20	×	12	29	+	10	38	B/O	52
21	F1/X	23	30	X → PC	4C			

**Инструкция.** Первые команды из программы «Приближение-1», переключатель углов в положении Р.

**Контрольный пример.**  $k=10$ . 10 В/О С/П (2 мин) «3.1415906» С/П (2 мин) «3.1415928». За две итерации получили шесть верных знаков.

## Глава 4 СОВРЕМЕННАЯ МИКРОВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

### 4.1. ЧТО МОГУТ И ЧЕГО НЕ МОГУТ НОВЫЕ МИКРОКАЛЬКУЛЯТОРЫ

Новые программируемые микрокалькуляторы «Электроника МК-61» и «Электроника МК-52» по сравнению со старыми моделями, получившими значительное распространение, имеют расширенный набор команд, дополнительный регистр адресуемой памяти, большую емкость оперативной памяти — 105 шагов вместо 98 у ПМК «Электроника МК-56».

Система команд у новых и старых ПМК одинакова, а «Электроника МК-52» способна хранить несколько программ в полупостоянном запоминающем устройстве (ППЗУ). К этой его особенности мы еще вернемся, а пока поговорим о том, что общего у новых калькуляторов, отличающее их от прежних моделей.

В новые модели можно вводить без изменения почти все программы, написанные для ПМК «Электроника МК-56». Почти — потому что команд  $K\bar{P} \rightarrow X_B \uparrow$ ,  $KX \rightarrow P_B \uparrow$  в МК-61 и МК-52 нет. Когда эти команды встречаются в программах, их нужно модифицировать. Например, можно заменить команду  $K\bar{P} \rightarrow X_B \uparrow$  последовательностью команд  $\Pi \rightarrow X_0$   $X \rightarrow PE$   $FO$   $K\bar{P} \rightarrow X_E$ , а команду  $KX \rightarrow P_B \uparrow$  последовательностью команд  $\Pi \rightarrow X_0$   $X \rightarrow PE$   $FO$   $KX \rightarrow PE$ . Конечно, при вводе такой видоизмененной программы придется заменить многие адреса.

Если читатель захочет использовать программы, приведенные в этой книге или любом справочнике, причем результат нужно будет получить как можно быстрее, то лучше всего применить известные приемы организации подпрограммы. Например, в программе есть фрагмент

41	X → PA	4—	43	X	12	45	42	42
42	KP → XB↑	GE↑	44	FL0	5Г	46	Fп	20

Если использовать подпрограмму, то она может выглядеть так:

41	X → PA	4—	43	98	98	45	42	42
42	PP	53	44	FL0	5Г	46	Fп	20

В этом случае подпрограмма будет такой:

98	$\Pi \rightarrow X_0$	60	A0	FO	25	A2	X	12
99	X → PE	4E	A1	KP → XE	GE	A3	B/O	52

Нетрудно заметить, что основная часть программы практически не изменилась. Конечно, для каждого конкретного случая следует искать оптимальный вариант. Иногда удается заменить команду  $K\bar{P} \rightarrow X_B \uparrow$  командами  $\Pi \rightarrow X_0$   $X \rightarrow PE$   $K\bar{P} \rightarrow X_E$  или  $\Pi \rightarrow X_0$   $1 + X \rightarrow PO$   $K\bar{P} \rightarrow X_0$ .

Новых команд 14. Четыре из них выполняют логические операции в шестнадцатеричной системе счисления и, по-видимому, не пригодятся для инженерных или научных расчетов. Некоторые логические команды могут быть использованы в программированном контроле и самоконтроле знаний учащихся в процессе обучения (см. ч. II, § 4.6). Надо иметь в виду, что есть случаи, когда при использовании логических команд возникают неожиданные сбои в работе вычислительного устройства. Четыре из оставшихся десяти команд предназначены для перевода временных и угловых мер в десятичную систему и обратно. Работа этих команд достаточно подробно освещена в инструкции, поэтому не станем на них останавливаться.

Гораздо чаще в практических программах встречаются команды  $K[X]$ ,  $K[X]$ ,  $K\{X\}$ ,  $Kmax$ ,  $KZN$ ,  $KSC$ . При работе с микрокалькулятором «Электроника МК-56» для реализации операции отделения целой части числа обычно требовалось три команды и один регистр адресуемой памяти, например  $X \rightarrow PA$   $K\bar{P} \rightarrow XA$   $\Pi \rightarrow XA$ . Однако это применимо лишь для чисел, не меньших единицы. Чтобы использовать операцию для отрицательных чисел, приходилось записывать последовательность команд:

00	$\Pi \rightarrow XA$	6—	03	$F10^x$	15	05	$FBX$	0
01	+	10	04	+	10	06	—	11
02	7	07						

Кроме того, необходимо было предварительно записывать число  $5,555555 \cdot 10^{-1}$  в РА. В новых калькуляторах во всех этих случаях достаточно одной команды K[X]. Ее действие напоминает действие всех одноместных операций, таких, как, например, Fex, т. е. число из РХ перемещается в РХ1, а в регистре X (и на индикаторе) оказывается результат — целая часть от предыдущего содержимого РХ, содержимое же регистров Y, Z, T не меняется. Аналогично работает и команда K{X}, только ее результатом будет дробная часть числа, содержащегося в РХ. Команды K[X] и K{X} позволяют хранить в одном регистре по два числа: целое и дробное (меньшее единицы). Общее количество значащих цифр в обоих числах, естественно, не может превышать восьми. Чтобы вызвать из регистра M целое число, нужно дать команды П→ХМ K[X], а чтобы вызвать хранящееся в этом же регистре дробное число, — команды П→ХМ K{X}. При этом нужно учитывать, что знак числа относится к дробной и к целой части.

К одноместным относятся также команды K[X] и КЗН. Вновь обратимся к микрокалькулятору «Электроника МК-56». С его помощью абсолютную величину числа можно было вычислить двумя способами. Первый — это замена операции двумя: FX<sup>2</sup> FV. Однако его можно применить лишь тогда, когда порядок числа не больше 50; для вычисления же модуля больших чисел необходимо использовать команды FX<0 A /—/, где A — адрес команды, следующий за /—/. В новых моделях для вычисления модуля любого числа достаточно использовать всего одну команду K[X].

Намного проще стало определять знак числа. Новая команда КЗН анализирует знак числа, находящегося в РХ, и засыпает в этот же регистр —1, если число было отрицательным, +1, если число положительное, и 0, если число равно нулю. Прежнее же содержимое РХ пересыпается в РХ1, а остальные регистры остаются с прежним содержимым.

Единственная новая двухместная команда — Ктхах. Из двух чисел, находящихся в РХ и РY, эта команда засыпает в регистр X большее число. Пусть, например, в РХ находится число 2, а в РY — число 3; после исполнения команды Ктхах число 3 попадает в РХ. Если же числа поменять местами, то работа команды выразится лишь в копировании числа 3 в РХ1. Содержимое же остальных регистров, Y, Z, T, не изменится — команда Ктхах перемещает числа в стеке подобно команде FXy, а прежнее содержимое РХ опускается в РХ1. Команда Ктхах одинаково хорошо спрашивается с положительными и отрицательными числами, но пасует перед нулем. Оба новых ПМК упорно считают нуль самым большим числом, а значит, нужны специальные меры, чтобы предупредить возможные ошибки.

Наконец, команда КСЧ — команда обращения к генератору псевдослучайных чисел. Эта команда могла бы найти применение

во многих серьезных программах, например в программах интегрирования методом Монте-Карло. Замечено, однако, что при использовании этой команды в программе генератор зацикливается и начинает выдавать одно и то же число. Если ввести в калькулятор «Электроника МК-52» программу, приведенную ниже, можно было бы ожидать, что случайные числа, вычисленные по команде КСЧ, будут равномерно распределены по регистрам.

00	КСЧ	3L	05	+	10	09	+	10
01	1	01	06	X→ПЕ	4E	10	KX→ПЕ	LE
02	0	00	07	КП→ХЕ	ГЕ	11	FLO	5Г
03	×	12	08	1	01	12	00	00
04	1	01						

Однако, заполнив один-два регистра, генератор зацикливался. Чтобы исключить зацикливание, необходимо дополнить команду КСЧ двумя другими, например В↑ ↔ или K{X} K{X}. Этот способ не совсем удачный — возрастает длина программы, «засоряется» стек.

Генератор способен вырабатывать ряд псевдослучайных чисел, длина которого равна 153. После включения калькулятора первыми будут выводиться числа начиная с 53-го члена. Убедитесь в этом позволяет следующая программа.

#### Программа СЛУЧАЙНОЕ ЧИСЛО

99	CX	0Г	03	КСЧ	3L	05	03	03
01	K{X}	35	04	FLO	5Г	06	C/P	50
02	B↑	0E						

Запишем 53 в Р0 и запустим программу В/О С/П. На индикаторе «0.6876331». Теперь запишем в Р0 число 153, очистим стек (CX) и вновь запустим программу, только теперь с адреса 03: БП 03 С/П «0.6876331». Такой короткий цикл — серьезный недостаток; для интегрирования команду использовать нельзя. Лучше при необходимости по-прежнему применить какую-либо известную программу, генерирующую псевдослучайные числа.

#### 4.2. РАБОТА С УСТРОЙСТВОМ ПАМЯТИ ПМК «ЭЛЕКТРОНИКА МК-52»

Термин «постоянное запоминающее устройство» относится, вообще говоря, к устройствам разных типов. Прежде всего, это устройство, куда информация записывается раз и навсегда. Примером может служить грампластинка: ее содержимое может храниться сколь угодно долго, но изменить его нельзя. Постоянные запоминающие устройства типа магнитофонной ленты или магнитного диска тоже могут хранить информацию практически вечно, однако эту информацию можно в любой момент обновить. Наконец, есть устройства, содержимое которых можно обновлять, но хранить информацию в них можно лишь весьма ограниченное

время, например день, месяц, год. К устройствам последнего типа относится и ППЗУ «Электроники МК-52».

Эти устройства реализованы на интегральных микросхемах, сохраняющих свое состояние и при отключении питания. Однако заряды, обеспечивающие хранение информации, хотя и медленно, но рассеиваются, и спустя несколько месяцев прочесть то, что записано в ППЗУ, уже нельзя. Правда, при включении калькулятора память каждый раз «подпитывается» и, таким образом, срок хранения информации увеличивается. Но если упустить время, то программу придется вводить и записывать заново.

Впрочем, главный недостаток ППЗУ заключается не в этом. Хотя емкость его памяти 512 команд, использовать его достаточно неудобно. Во-первых, обратиться к ППЗУ для записи и считывания из программы нельзя, это можно сделать только вручную. Во-вторых, хотя оперативная память ПМК «Электроника МК-52» позволяет размещать 105 команд, записать в запоминающее устройство можно только 98 — как будто его проектировали для ПМК «Электроника МК-56». То же касается и регистров: записать в память можно только содержимое первых 14 регистров. В-третьих (и это, наверное, главное), правила записи программ в ППЗУ настолько громоздки, что им впору посвящать фундаментальные монографии.

Нумерация элементов идет по ячейкам. Команда, которой мы привыкли измерять программу, занимает две ячейки. Стирание информации — эту операцию необходимо выполнить непосредственно перед записью новой программы — происходит по строкам, в каждой из которых по 16 ячеек, т. е. по восемь команд. Запись же происходит пачками по семь команд. Составляя адрес обращения к ППЗУ, следует учитывать оба этих требования, иначе можно испортить программы, записанные ранее.

Адрес обращения состоит из трех частей. Первая — незначащая цифра от 1 до 9, вторая (начальный адрес программы) — четырехзначное число от 0000 до 1023 и третья часть — число команд в программе, которую нужно записать. Это число должно быть кратно семи. Если длина программы не удовлетворяет этому условию, то необходимо выбрать наименьшее число, превышающее длину программы и делящееся на семь без остатка. Например, если реальная длина программы 25 команд, то следует взять число 28. Получать начальный адрес нужно так. Адрес предыдущей программы складывается с числом ячеек, необходимых для ее записи, и дополняется до ближайшего числа, кратного 16.

**Контрольный пример.** Требуется записать в ППЗУ первую программу, длина которой 98 команд. Так как число это кратно семи, то полный адрес запишется как 1000098. Переключатель Д—П ставим в положение Д, переключатель С—З—СЧ — в положение С. Нажимаем клавиши A↑ ↑↓. Область памяти в ППЗУ очищена, можно переходить к записи. Ставим переключатель

Д—П в положение П, переставляем второй переключатель в положение З. Осталось нажать клавишу A↑, и запись будет произведена.

Пусть вторая программа состоит из 30 команд. Начальный адрес для нее вычисляем как  $0000 + 98 + 98 = 0196$ . Но это число не кратно 16. Выбираем ближайшее удовлетворяющее этому условию число 0208. Далее находим число, превышающее длину записываемой программы и кратное семи. Это будет число 35. Теперь можно сформировать полный адрес 2020835. Последовательность действий при записи уже описана.

Для считывания программы из ППЗУ достаточно набрать на клавиатуре нужный адрес, установить переключатели в положение П и СЧ и нажать клавиши A↑ ↑↓. Это, несомненно, быстрее, чем набирать всю программу вновь.

Мы не стали подробно говорить о переключателях С—З—СЧ и Д—П, поскольку их назначение подробно описано в инструкции по эксплуатации ПМК «Электроника МК-52».

Таким образом, общая емкость памяти 512 байт, организованных в 1024 четырехразрядные (четырехбитовые) ячейки. Так как емкость ячейки программной памяти и регистра числового памяти составляет соответственно 8 и 56 бит, то для хранения одного шага программы и содержимого одного регистра памяти требуется соответственно 2 и 14 ячеек ППЗУ.

Адресное поле памяти ППЗУ разбито на 64 строки по 16 ячеек с адресом начальной ячейки в *n*-й строке  $A_{no} = 16(n-1)$  и адресом последней ячейки  $A_{nk} = 16n - 1$ .

Все время до и после обращения к ППЗУ переключатель С—З—СЧ должен находиться в положении СЧ, чтобы случайное нажатие клавиши ↑↓ не привело к стиранию или искажению содержимого ППЗУ или программной и числовой памяти. Если стирание, запись и считывание выполняются при обращении к одному и тому же участку памяти ППЗУ, то при выборе другого режима обращения достаточно лишь изменить положение переключателей и, не вводя повторно адрес обращения, нажать клавишу ↑↓.

Для микрокалькулятора «Электроника МК-52», снабженного разъемами для подключения внешних устройств, разработан блок постоянной памяти — ПЗУ емкостью 1 Кбайт. Весь объем ПЗУ разбит на две страницы, емкость каждой из них равна емкости ППЗУ. Переключателем, установленным на корпусе блока расширения памяти, выбирают ту или иную страницу. Считывание выполняют с помощью адреса обращения. Процедура выбора адреса не отличается от той, что мы рассмотрели применительно к ППЗУ.

«Электроника МК-85» не только может выполнять большой ассортимент вычислительных операций, как этого требует название «микрокалькулятор», но и, подобно «настоящему» компьютеру, открывает немалые возможности для решения в автоматическом режиме довольно сложных задач по введенной заранее программе. Для пользователя — так называют людей, работающих с компьютерами, — главное отличие состоит в том, что общение с МК-85 происходит на алгоритмическом языке Бейсик. Пользователь общается с МК-85, давая указания общепринятыми в этом языке командами-словами, а не комбинациями из цифр и нескольких букв, как в ПМК. Это чрезвычайно важное отличие позволяет говорить о том, что МК-85 имеет все основания называться портативной вычислительной машиной, несмотря на ее малые габаритные размеры — 165×72×13 мм и массу — 150 г.

Вариант Бейсика, используемый в микрокомпьютере «Электроника МК-85», легко изучить даже тому, кто не знает еще ни одного языка программирования и ни разу не писал программу для компьютеров. Для этого даже не нужно знать всех особенностей устройства компьютера. Так же, как на любой персональной ЭВМ, программа вводится с помощью клавиатуры — на лицевой панели компьютера есть для этого клавиши с буквами латинского и русского алфавита. Во время ввода данных и команд подтверждение своих действий в виде слов и цифр вы видите на небольшом дисплее компьютера. Здесь же появляется все, что машина в процессе работы хочет сообщить пользователю.

Команды и функции Бейсика — их всего 54, которые понимает МК-85, позволяют запрограммировать решение достаточно сложных задач, выполнять многие реально встречающиеся инженерные и статистические расчеты, проводить не очень громоздкий численный научный эксперимент, составлять игровые программы.

«Электроника МК-85» — первый компьютер с универсальным питанием. Он может работать не только от сети через блок питания, но и от встроенной батарейки, энергии которой хватает для работы микроЭВМ в течение нескольких месяцев. Еще одно достоинство МК-85 — он может хранить программы в течение многих месяцев, до тех пор, пока работоспособны гальванические элементы, независимо от того, включен или выключен компьютер, запоминающее устройство МК-85 получает питание напрямую, минуя основной выключатель. Ток, потребляемый микросхемой памяти, настолько мал, что даже во время замены элементов все программы сохраняются, если пауза между отключением старых и установкой новых источников питания не превысит 15 мин.

Решающий вклад в высокую экономичность МК-85 дает его жидкокристаллический дисплей — ЖК-индикатор. Обычно в ПМК используют катодлюминесцентные индикаторы, на которых четко

высвечиваются достаточно яркие зеленоватые цифры. Есть у таких индикаторов и немало недостатков. Этот непростой электровакуумный прибор очень чувствителен к толчкам и ударам, при слабом внешнем освещении он быстро утомляет зрение, а при ярком солнечном свете на экране трудно что-либо разобрать, несмотря на обязательный светофильтр. ЖК-индикатор выдержит даже сильные толчки и удары. Однако за достоинства нужно платить: ЖК-индикатор сам не светится, им можно пользоваться только при внешнем освещении. Кстати, в МК-85 есть регулятор контрастности, который позволяет наилучшим образом согласовать изображение на экране с условиями освещения.

На индикатор одновременно может быть выведено 12 символов. В программах, написанных на Бейсике, встречаются строки из 40—50 символов. Чтобы не удлинять индикатор, используется принцип бегущей строки — аналог того, что можно увидеть в световых рекламах. Текст, появляясь буквой за буквой у правого края индикатора, сдвигается влево до тех пор, пока не пройдет вся строка. Индикатор — это как бы окно длиной в 12 символов, которое может скользить по неподвижной строке из 63 символов. В определенный момент на индикаторе появляется мерцающий темный квадратик, предупреждая о том, что строка заканчивается. Положением окна можно управлять, сдвигая его вправо и влево по строке специальными клавишами. В верхней части индикатора — служебная строка, в ней постоянно отображаются все сведения о текущем режиме работы машины, указывается емкость оставшейся свободной памяти.

Оперативная память микрокомпьютера МК-85 сейчас имеет емкость 2К. В распоряжении пользователя предоставлена не вся имеющаяся оперативная память. Часть ее используется для хранения информации, которая выводится на индикатор, другая необходима для работы самого Бейсика, ведь Бейсик — это тоже программа, которая переводит введенные с клавиатуры указания на язык, понятный компьютеру. Всего для свободного пользования остается 1000 байт оперативной памяти. С учетом этого максимальная длина программы, записанной в памяти МК-85, может составлять примерно 100 операторов Бейсика. Для хранения данных и необходимых констант можно использовать 26 ячеек памяти. К ним можно обращаться как напрямую, называя имя переменной, например *A* или *C*, так и косвенно, называя номер ячейки.

Если важнее иметь больше ячеек для переменных и констант, то их число можно увеличить за счет длины программы (по цепи 8 байт за ячейку памяти для записи каждой переменной либо константы). Максимально можно получить  $1000 : 8 + 26 = 151$  такую ячейку; одна в этом случае на программу не останется ни байта.

Микропроцессор, который установлен в «Электронике МК-85», 8-разрядный. Такой процессор производит все операции одновременно с 8 разрядами многоразрядного двоичного числа. Для оценки быстродействия МК-85 в него была введена программа на Бейсике для вычисления факториала  $33!$ , ответ будет получен через 4,7 с.

В МК-85 предусмотрен и режим повышенного быстродействия, когда на решение той же задачи компьютеру нужно всего 1,2 с. Платой за скорость является повышенное потребление энергии. Для сравнения заметим, что факториал  $33!$  на калькуляторе «Электроника МК-52» вычисляется за 37 с. Поскольку потребление электроэнергии в режиме повышенного быстродействия может быть значительным, то, чтобы сохранить энергию батареи, используют преимущества универсального питания: МК-85 включают в сеть через блок питания. Впрочем, ускоренный режим необходим лишь для вычислений по сравнительно сложным программам, поэтому необходимость подключения к сети возникает сравнительно редко.

Все вычисления компьютер производит с 12-разрядными числами, однако результат вычислений выдается на индикатор с 10-разрядной мантиссой. Если результат получен в экспоненциальной форме, мантисса будет иметь 8 разрядов.

Кроме четырех основных арифметических действий, «Электроника МК-85» автоматически вычисляет следующие функции:

тригонометрические функции  $\sin x$  (SIN),  $\cos x$  (COS),  $\operatorname{tg} x$  (TAN), обратные тригонометрические функции  $\sin^{-1}(x)$  (ASN),  $\cos^{-1}(x)$  (ACS),  $\operatorname{tg}^{-1} x$  (ATN),

квадратный крен  $\sqrt{x}$  (SQR), логарифмы  $\ln x$  (LN),  $\lg x$  (LOG), экспоненциальная функция  $e^x$  (EXP),

модуль числа  $|x|$  (ABS), целая часть числа  $[x]$  (INT),

дробная часть числа  $\{x\}$  (FRAC),

знак числа  $\operatorname{sgn} x = \begin{cases} 1, & x > 0, \\ 0, & x = 0, (\operatorname{SGN}), \\ -1, & x < 0 \end{cases}$

округление числа  $\operatorname{RND}(x, y)$  ( $\operatorname{RND}(\dots, \dots)$ ), генерация случайного числа  $r$  ( $\operatorname{RAN}\#$ ),

вызов результата предыдущего вычисления (ANS), Список команд и операторов

INPUT — присвоение переменной численного значения, вводимого в процессе выполнения программы,

KEY — запись в текстовую переменную символа, соответствующего нажатой во время исполнения оператора клавише,

PRINT — вывод информации на индикатор,  
CSR — указание позиции, с которой должна начинаться индикация,

GOTO — безусловная передача управления на строку программы с указанным номером,

IF ... THEN — передача управления в зависимости от истинности какого-либо условия,

GOSUB — переход к первому оператору подпрограммы, RETURN — возвращение в основную программу после выполнения подпрограммы,

FOR ... TO ... STEP ... — формирование цикла с заданным шагом,

NEXT — обозначение конца цикла, STOP — временное прерывание выполнения программы, END — обозначение конца программы, VAC — очистка содержимого всех переменных, включая текстовые,

LIST — выдача на индикатор текста программы, RUN — запуск программы на счет, CLEAR — стирание обозначенной программы, CLEARA — стирание всех программ, MODE — задание нужного способа представления угловых величин,

LEN — определение числа символов в составе текстовой переменной, MID — выделение символов из состава текстовой переменной, VAL — преобразование текстовой переменной в числовую, AUTO — автоматическая нумерация строк программы, DEFM — увеличение количества адресуемых регистров памяти,

DEL — удаление ошибочно введенного символа, AC — очистка регистра индикации и рабочего регистра, INS — раздвижка строки,

SET — определение количества значащих цифр в числе при округлении, т. е. задания числа разрядов мантиссы.  
«Электроника МК-85» открывает новую серию микроЭВМ; две новые модификации, которые разрабатываются промышленностью, — это электронная записная книжка и микрокомпьютер, который можно будет подключить к печатающему устройству и бытовому магнитофону, чтобы использовать его в качестве внешнего запоминающего устройства.

#### 4.4. МИКРОЭВМ «МИКРОША»

Эта микроЭВМ относится к классу простейших, или, как их принято называть, бытовых компьютеров. Он создан на основе комплекта микросхем серии K580, программы хранятся на магни-

тофонных кассетах в закодированном виде, текст и результаты расчета выводятся на экран телевизора. Для подключения к телевизору в комплект компьютера входит небольшая, размером чуть больше спичечного коробка приставка-модулятор.

При выводе информации на экране можно разместить 24 строки из букв, русских и английских, или цифр по 64 символа в строке. В свою очередь, каждый символ формируется на точечном растре из  $8 \times 8 = 64$  точек, один вертикальный ряд уходит на пробел между символами. Небольшие растровые элементы можно использовать для рисования на экране. Рисовать можно полностью закрашенным растровым пятном — символом размером  $4 \times 4$  точки или 15 другими знаками: вертикальными и горизонтальными штрихами, Г-образными и другими символами. В режиме псевдографики изображение на экране может состоять из  $128 \times 50 = 6400$  элементов.

Компьютер выполняет простейшую операцию сложения за 3 мкс, его среднее быстродействие — 200—300 опер./с.

Оперативное запоминающее устройство «Микроши» составляет 32 Кбайт, постоянное запоминающее устройство — 2 Кбайт. Предусмотрен в машине и разъем внутреннего интерфейса, позволяющий подключать к компьютеру блоки ОЗУ или ПЗУ емкостью 24 Кбайт. Такая конструкция компьютера диктуется его малой стоимостью, однако создает определенные неудобства в работе: каждый сеанс работы с компьютером на языке Бейсик требует предварительной записи в компьютер кодов этого языка с магнитофонной ленты. Это требует примерно 1,5 мин. Важно понимать, что введенная программа занимает часть памяти, а именно около 9 Кбайт. Еще одна значительная часть оперативной памяти выделяется для хранения всего того, что отображается на экране, лишь оставшаяся часть доступна для создаваемых пользователем программ и результатов расчета.

Чтобы облегчить изучение языка Бейсик, компьютер снабжен магнитофонной кассетой, на которой записаны семь обучающих программ, объединенных в один цикл — «Уроки Бейсики». Общий объем уроков — 80 Кбайт, и этого достаточно, чтобы в диалоге с машиной освоить сначала основы, а затем и тонкости Бейсики. На той же кассете записаны несколько игровых программ.

На «Микроше» можно работать и в кодах, тогда в распоряжении программиста оказывается чуть больше памяти, причем особенно значительно повышается быстродействие: программы, составленные в машинных кодах, выполняются компьютером примерно в 500 раз быстрее.

#### 4.5. МИКРОЭВМ «ЭЛЕКТРОНИКА БК-0010»

МикроЭВМ БК-0010 относится к бытовым компьютерам. Так же, как и «Микроша», БК-0010 рассчитан на работу совместно с телевизором и кассетным магнитофоном. При выводе информации

на экране можно разместить 24 строки по 32 или 64 символа в строке. В отличие от «Микроши» на БК-0010 предусмотрена возможность работы с графикой. Экран состоит из  $256 \times 512$  адресуемых точек, которые нередко называют пикселями. При необходимости вместо этого можно пользоваться режимом, когда на экране доступны  $256 \times 240$  точек, однако в этом случае пользователь получает в свое распоряжение четыре градации яркости — от белого до черного.

Среднее быстродействие БК-0010 такое же, как у «Микроши», — 300 тыс. операций в секунду. Емкость оперативного запоминающего устройства «Электроники БК-0010» 32 Кбайт. Их можно использовать в двух режимах — 16 Кбайт для хранения содержимого экрана, 16 Кбайт — в распоряжении пользователя. Это основной режим. Есть и другой, когда пользователю выделяется 24 Кбайт. В этом случае, однако, от экрана остается лишь небольшое «окно»; такой режим имеет смысл использовать только для математических расчетов.

Постоянное запоминающее устройство БК-0010 составляет 24 Кбайт, а в некоторых модификациях (например, БК-0010 Ш) и 32 Кбайт. Такая компоновка компьютера означает, что сразу же после включения в сеть компьютер готов к диалогу на языке высокого уровня. Таким языком для БК-0010 является Фокал. Предусмотрена возможность работать и с программами, составленными в кодах. Кроме того, пользователь может загрузить в компьютер другие языки программирования: Бейсик, Форт, Паскаль, Ассемблер, Х-Фокал или Т-язык. В этом случае, однако, емкость памяти, остающаяся в распоряжении пользователя, существенно уменьшается.

Серия БК продолжает развиваться, новые модели БК-0010.01 и БК-0011 характеризуются расширенными возможностями. Так, например, емкость ОЗУ БК-0011 составляет 128 Кбайт.

## Часть II

# ПРОГРАММИРОВАНИЕ РЕШЕНИЯ ИГРОВЫХ И ЛОГИЧЕСКИХ ЗАДАЧ

### Глава 1

#### ЧЕЛОВЕК И МАШИНА В ПРОТИВОСТОЯНИИ И СОЮЗЕ

##### 1.1. ВРАЩАЮЩИЙСЯ БОЧОНОК

Ты помнишь, конечно, с детства чудесные арабские сказки про волшебную лампу Алладина, коварных разбойников и могучих джиннов.

Жил в старину злой и жестокий царь Шахрияр. Каждый день он брал новую жену, а наутро убивал ее. Дошел черед до красавицы Шахразады. Заметив ночью, что царь не спит, Шахразада рассказала ему сказку. Когда наступило утро, она остановилась на самом интересном месте. Царю очень хотелось услышать продолжение и он отложил казнь до следующего утра. Это продолжалось другую, третью ночь — тысячу ночей. Так говорит древняя легенда.

Когда пришла тысяча первая ночь и Шахразада окончила последний рассказ... Нет, нет, была и тысяча вторая ночь, и еще много дней и ночей. Но этого продолжения ты, клянусь Аллахом, не знаешь.

— Ты истощила свои знания, — сказал царь, ворочаясь от бессонницы и злобы. — Прояви же мудрость. Я подарю тебе жизнь, если угадаешь, что я с тобой сделаю: казню или помилую.

— О всемогущий царь! — молвила Шахразада. — Милость твоя беспрепредельна, ты велишь меня казнить.

— Она угадала, подумал царь. — Я должен ее помиловать?!... Но если помилую, то не угадала — надо казнить. Если казнить... О, великий Аллах!

Шахразада читала мысли царя.

— О мудрейший из великих, — сказала она. — Есть много интересного, чего не найдешь ни в одной сказке. Будем решать задачи. Прикажи принести бочонок и прорезать в его верхней крышке отверстия для продевания рук.

...Шахразада так увлеклась игрой, что забыла, кто ее противник. Перед ней сидел просто царь, царь природы, как издавна называют человека, и она учила его думать. Потому что разум человеческий — всего превыше, потому что есть на свете великое чувство любопытства, которое, помнишь, победило даже жестокое сердце персидского владыки.

Присядем и мы, читатель, около бочонка.

В верхней крышке бочонка четыре симметричных отверстия, и под каждым бутылка горлышком вверх — обозначим кодом 6 —

или вниз — 9 (рис. 3,а). Сделать ход — значит продеть руки в два отверстия и поставить находящиеся под ними бутылки в любое из двух возможных положений. Если после удаления рук из отверстий все четыре бутылки стоят одинаково — все коды 6 или все 9, то бочонок неподвижен, задача решена. В противном случае бочонок начинает быстро вращаться вокруг вертикальной оси, и к моменту остановки играющий теряет из виду использованные отверстия. Требуется при любом начальном положении бутылек решить задачу не более чем в 5 ходов.

Планируя в дальнейшем совместное решение задачи человеком и ПМК, распределим обязанности между ними. Человек задает начальное положение бутылек в отверстиях и после каждого хода ПМК называет число поворотов бочонка. ПМК выполняет ходы, индицирует возникшие коды бутылек, производит вращение согласно указаниям человека.

Для удобства рассуждения положим, что бочонок останавливается после определенного числа поворотов по часовой стрелке на углы  $\pi n/2$  ( $n = 1, 2, 3, \dots$ ). Картина повторяется через каждые 4 значения  $n$ . Число поворотов на  $\pi/2$  равно остатку  $r$  от деления  $n$  на 4. В дальнейшем будем считать, что при  $r=0$   $n=4$ . Условимся рассматривать пары противолежащих и смежных отверстий. Ясно, что независимо от числа оборотов противолежащие отверстия остаются противолежащими, смежные — смежными. Говорят, что свойства отверстий быть противолежащими или смежными являются инвариантами поворота.

Ситуацией будем называть упорядоченную четверку цифр 6 и 9:  $S = (a_1, a_2, a_3, a_4)$ , где  $a_i = 6, 9; i = 1, 2, 3, 4$ ;  $a_1$  — код левого верхнего отверстия неподвижного бочонка;  $a_2, a_3, a_4$  — коды других отверстий при отсчете с интервалом  $\pi/2$  по часовой стрелке. Так, ситуации (6, 9, 9, 9) отвечает конфигурация  $\begin{smallmatrix} 6 & 9 \\ 9 & 9 \end{smallmatrix}$ . Всего ситуаций 16.

Обратим внимание на принципиальное отличие структуры описания ситуации в игре «Бочонок» от других задач (см., например, § 1.2). Суть в том, что ситуация, возникшая после вращения, ПМК неизвестна. Неопределенность раскрывается в процессе хода ПМК, но не до него, как часто бывает. Речь идет об игре с неполной информацией, в этом ее сложность и оригинальность. Диалог человек — ПМК напоминает, скорее, разговор глухих: ПМК: «Не знаю, что вы сказали, но это вам не поможет».

а)

б)

Рис. 3

Вернемся к описанию. В игре две целевые ситуации  $S^* = (6, 6, 6)$  и  $S^{**} = (9, 9, 9, 9)$ . Расстоянием между ситуациями  $S_1$  и  $S_2$ , возникающими в игре, считается целое неотрицательное число, равное количеству несовпадений элементов в соответствующих ситуациях  $\rho(S_1, S_2)$ . Так, если  $S_1 = (9, 6, 6, 9)$ , а  $S_2 = (6, 6, 9, 9)$ , то  $\rho = 2$  — элементы в первой и третьей позициях не совпадают.

Можно доказать, что относительно  $\rho$  множество ситуаций образует метрическое пространство (оно называется пространством Хемминга). Одной из замечательных особенностей этого пространства является свойство треугольника: для любых  $S_1, S_2, S_3$  верно  $\rho(S_1, S_2) \leq \rho(S_1, S_3) + \rho(S_2, S_3)$ . Неравенство можно интерпретировать так. Если  $S_1$  и  $S_2$  поразнь близки к  $S_3$ , то они недалеки друг от друга. В частности, две ситуации, близкие к целевой, различаются не очень сильно. Это служит основанием к построению постепенного, без больших скачков, приближения к целевой ситуации.

Главным ориентиром в построении последовательности приближений к цели является исключение повторений ранее встречавшихся в игре ситуаций. Чтобы этот ориентир «заработал», уточним в решении данной задачи само понятие повторения ситуации. Дело в том, что в любой ситуации нас прежде всего интересует расстояние до цели, поэтому две ситуации, отличающиеся значениями всех четырех компонентов, равны. Например,  $S_1 = (9, 9, 9, 6)$  и  $S_2 = (6, 6, 6, 9)$ ,  $S_1 = S_2$ , так как  $\rho(S_1, S^{**}) = 1$ ,  $\rho(S_2, S^*) = 1$ . Это можно сформулировать так. Перемена значений всех компонентов на противоположные (6 на 9 и наоборот) не изменяет ситуации, т. е. является инвариантным преобразованием, сохраняющим ситуацию. Равенство расстояний надо понимать в смысле «одинаковой близости» к цели соответствующих ситуаций — нас интересует именно это, а не равенство ситуаций в пространстве Хемминга.

Предлагаем следующий алгоритм игры.

1. Продеваем руки в два противолежащих отверстия, ставим каждую бутылку под ними горлышком вверх независимо от предыдущего положения — ход (6, 6). Если коды двух других отверстий также (6), то задача решена — бочонок неподвижен. В противном случае переходим к указанию 2.

2. Код (6,6) в смежных отверстиях. Теперь имеем три гарантированных кода (6). Если четвертый код также (6), то задача решена. Если четвертый код (9), переходим к указанию 3.

3. Выбираем противолежащие отверстия. Пусть рука оказалась в отверстии с кодом (9), тогда этот код изменяем на (6), и задача решена. В противном случае код (6) заменяется кодом (9). Переходим к указанию 4.

4. В двух смежных отверстиях изменяем коды на противоположные. Если до этого коды были одинаковые, то задача решена, иначе, переходим к указанию 5.

5. Противолежащие отверстия. Изменяем коды на противоположные, задача решена.

Рассмотрим, как строится алгоритм. Будем последовательно исключать ситуации, ранее встречавшиеся в игре. Начнем с третьего хода, когда снята неопределенность с количества кодов (6) и (9).

Возьмем противолежащие отверстия. Предположим менее благоприятный случай — под руками две 6. Оставить ситуацию как есть — значит повторить ее. В условиях, когда добрый ход, это непозволительная роскошь. Замена (6,6) на (9,9) опять ведет к повторению. Мы знаем, что три 6 и одна 9 образуют в смысле близости к цели такую же ситуацию, как три 9 и одна 6. Единственная возможность исключить повторение — замена одного кода (6) на (9). Это действие выполнено в третьем ходе. В последующих ходах (четвертом и пятом) исключить повторения можно только изменив на противоположные коды в фиксированных отверстиях.

**Замечание.** Наибольшую трудность в решении задачи обычно вызывает третий ход, когда код (6) необходимо заменить кодом (9). Человек обычно воспринимает это как отступление, многие отказываются от дальнейших действий, утверждая, что решения нет. Анализ этого наблюдения позволяет сделать вывод: не всем удается преодолеть необходимость увеличения расстояния  $\rho$  до цели  $S^*$  в ситуации, когда цель, казалось бы, почти достигнута. Возможно, именно метрика Хемминга служит математической моделью умственной деятельности человека при выработке игровой стратегии.

Одного алгоритма выполнения ходов недостаточно для составления программы. Важно правильно организовать взаимодействие

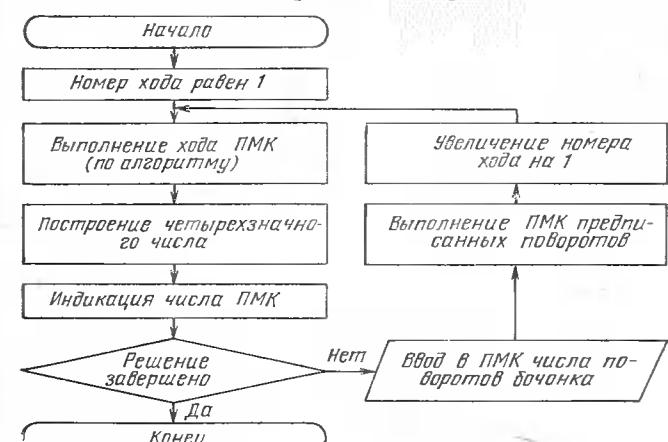


Рис. 4

вие человек — ПМК в игре (рис. 4). По этой схеме программу составить несложно. Блоки, входящие в состав цикла, используются во всех ходах одинаково, а значит, их целесообразно оформить подпрограммой (вспомогательным алгоритмом на УАЯ).

### Программа БОЧОНОК

#### Распределение памяти

- P1 — счетчик числа поворотов,  
P6 — P9 — коды бутылок начиная с левой верхней позиции в направлении по часовой стрелке,  
PA — для вызова (засылки) содержимого из P7, P8,  
PC — для кольцевого перемещения информации в P6 — P9,  
RD — адрес начала подпрограммы при косвенном обращении.

00	CX	0Г	23	KППД	—Г	45	1	01
01	4	04	24	7	07	46	0	00
02	0	00	25	X→PA	4—	47	X	12
03	X→ПД	4Г	26	1	01	48	KП→X5	G5
04	6	06	27	5	05	49	+	10
05	X→П6	46	28	П→X6	66	50	FL1	5L
06	X→П8	48	29	—	11	51	45	45
07	KППД	—Г	30	X→П6	46	52	C/P	50
08	6	06	31	1	01	53	X→П1	41
09	X→П6	46	32	5	05	54	П→X9	69
10	X→П7	47	33	KП→XA	Г—	55	X→ПС	4C
11	KППД	—Г	34	—	11	56	П→X8	68
12	П→X6	66	35	KХ→PA	L—	57	X→П9	49
13	П→X8	68	36	KППД	—Г	58	П→X7	67
14	—	11	37	8	08	59	X→П8	48
15	FX≠0	57	38	БП	51	60	П→X6	66
16	21	21	39	25	25	61	X→П7	47
17	6	06	40	5	05	62	П→XС	6C
18	X→П6	46	41	X→П5	45	63	X→П6	46
19	X→П8	48	42	3	03	64	FL1	5L
20	KППД	—Г	43	X→П1	41	65	54	54
21	9	09	44	KП→X5	G5	66	B/O	52
22	X→П6	46						

Инструкция. В/О С/П (12 с). После включения ПМК находится в режиме вычислений. Переход к режиму ввода и редактирования программы производится нажатием клавиш В/О F ПРГ, возврат в режим вычислений — клавиш F АВТ. Задан вектор исходной ситуации — начального положения бутылок в отверстиях, начиная с левой верхней позиции и далее по часовой стрелке с интервалом  $\pi/2$ . Положение бутылки горлышком вверх кодируется как (6), вниз (9). Введите в Р6 код первой бутылки, в Р7 — код следующей по часовой стрелке бутылки и т. д. Запустите программу, нажав клавиши В/О С/П. Примерно через 10 с на индикаторе вы увидите ситуацию, возникшую в игре после хода ПМК, — четырехзначное число из (6) и (9), где первая цифра — код бутылки в первом отверстии, следующая — во втором и т. д.

Если все цифры одинаковы — все 6 или все 9, то задача решена. В противном случае наберите на клавиатуре цифры 1, 2, 3 или 4 — число поворотов бочонка по часовой стрелке на  $\pi/2$ ; нажав клавишу С/П, запустите программу на продолжение вычислений. ПМК выполняет предписанное число поворотов бочонка — преобразует вектор, делает следующий ход и индицирует результат — четырехзначное число. Если задача не решена, снова наберите число

поворотов и т. д. На каждый ход ПМК требуется в зависимости от числа поворотов до 17 с. В худшем случае независимо от исходной ситуации задача решается пятью ходами за 1,5 мин.

Напомним правило краткой записи инструкций\*. Число в кавычках означает, что приводится результат, который читается на индикаторе. Время между двумя индикациями будем указывать в круглых скобках. Так, запись 3 В/О С/П (2 с) «5» означает: вводим 3, после нажатий клавиш В/О С/П через 2 с индицируется 5.

Контрольный пример. Исходная ситуация 9669 или на плоскости 9 6. 9 X→П6 X→П9 6 X→П7 X→П8 В/О С/П (12 с) «6669» 3 С/П (25 с) «6696» 1 С/П (17 с) «9669» 2 С/П (23 с) «9696» 4 С/П (32 с) «6666» — задача решена.

#### Структура программы

- 00 : ввод нуля в регистр X.  
01—03 : формирование параметра для косвенного обращения к подпрограмме построения числа, индикации, поворотов. Обращение происходит после каждого хода ПМК, поэтому одношаговое косвенное обращение предпочтительнее двухшагового прямого.  
04—07 : первый ход ПМК, обращение к подпрограмме.  
08—11 : второй ход ПМК, обращение к подпрограмме.  
12—23 : третий ход, обращение к подпрограмме. Возможны два случая. Если содержимое Р6 и Р8 неодинаково (т. е. бутылки в первом и третьем отверстиях стоят по-разному), то содержимое обоих регистров равно 6 (команды 12—15, 17—19). В противном случае содержимое Р6 равно 9 (команды 12—15, 21, 22). В первом варианте после индикации хода ПМК станет ясно, что задача решена, во втором — продолжение вращения и переход к команде 24.  
24—39 : четвертый и пятый ходы, производятся по единой схеме — замена содержимого соответствующих регистров на противоположное, достигаемое вычитанием данного содержимого из числа 15. В четвертом ходе процедура относится к Р6 и Р7, в пятом к Р6 и Р8. Выбор варианта (команды 33 и 35) в программе таков. Число 7 засыпается в РА (команда 24). По команде 33 в РХ вызывается содержимое Р7, вычитается из 15, и разность по команде 35 становится содержимым Р7, которое, таким образом, заменяется на противоположное (команды 31—35). Однако по возвращении после поворота для выполнения пятого хода, если, разумеется, задача еще не решена, в РА заносится число 8, и по командам 31—35 теперь уже заменяется на противоположное содержимое Р8 — выполняется пятый ход.

\* Способ записи инструкций разработан В. В. Козловым и А. Б. Бойко.

40—52 : на основе содержимого Р6—Р9 в подпрограмме строится четырехзначное число из 6 и 9, число индицируется (команда 52). В этот момент проверяется окончание: если число составлено из четырех 6 или 9, то задача решена. В противном случае человек вводит число поворотов; нажатием клавиши С/П запускается продолжение вычислений.

53—66 : выполнение вращения — в Р1 вводится число поворотов  $n$  (команда 53), затем в Р6—Р9 производятся круговые перестановки по числу введенных поворотов. По команде 66 ПМК возвращается к выполнению очередного хода.

### Алгоритм БОЧОНОК на УАЯ

Потребуются вспомогательные алгоритмы.

1. Вспомогательный алгоритм Горнер — для выражения  $n$  элементов линейной таблицы с десятичным числом ( $T$ ). В программе «Бочонок» ему соответствует часть подпрограммы формирования, индикации и вращения (команды 40—52).

Замечание. Для вывода таблицы на УАЯ, разумеется, специальных средств не требуется. Нам же алгоритм необходим для приближения к ограниченным возможностям ПМК, у которого нет процедур вывода массива.

алг ГОРНЕР (нат  $n$ , Т, нат таб  $c[1:n]$ )

```
арг  $n$ ,  $c$ 
рез Т
нач nat  $j$ .
     $j := 2$ ;  $T := c[1]$ 
    пока  $j \leq n$ 
        нц
             $T := T * 10 + c[j]$ ;  $j := j + 1$ 
        кц
кон
```

2. Вспомогательный алгоритм «Вращение» — выполнение последовательно  $n$  круговых перестановок массива  $x$  на 4 элемента. В программе «Бочонок» ему соответствует часть подпрограммы формирования, индикации и вращения (команды 54—65).

алг ВРАЩЕНИЕ (нат  $n$ , нат таб  $x[1:4]$ )

```
арг  $n$ ,  $x$ 
рез  $x$ 
нач nat  $k$ ,  $b$ 
     $k := 1$ 
    пока  $k \leq n$ 
        нц
             $b := x[4]$ ;  $x[4] := x[3]$ ;  $x[3] := x[2]$ ;
             $x[2] := x[1]$ ;  $x[1] := b$ ;  $k := k + 1$ 
        кц
кон
```

алг БОЧОНОК (нат таб  $B[1:4]$ , нат  $R$ , лит  $y$ )

```
арг  $B$ 
рез  $R$ ,  $y$ 
```

```
нач nat  $i$ 
     $R := 1$ ;  $i := 1$ 
    пока  $(R - 6666) \cdot (R - 9999) \neq 0$ 
        нц если  $i := 1$ 
            то  $B[1] := 6$ ;  $B[3] := 6$ 
            иначе если  $i = 2$ 
                то  $B[1] := 6$ ;  $B[2] = 6$ 
                иначе если  $(i = 3) \wedge (B[1] \neq B[3])$ 
                    то  $B[1] := 6$ ;  $B[3] := 6$ 
                    иначе если  $i = 3$ 
                        то  $B[1] := 9$ 
                        иначе если  $i = 4$ 
                            то  $B[1] := 15 - B[1]$ ;
                             $B[2] := 15 - B[2]$ 
                            иначе  $B[1] := 15 - B[1]$ ;
                             $B[3] := 15 - B[3]$ 
```

все

все

все

ГОРНЕР ( $4$ ,  $R$ ,  $B$ )
 выв  $R$ ; ввод  $\Pi_i$ ; ВРАЩЕНИЕ ( $\Pi_i$ ,  $B$ )
  $i := i + 1$

кц  
кон

Примечание.  $\wedge$  — знак логической конъюнкции (И); выв и ввод — команды, используемые для организации диалога человек — ЭВМ.

### Алгоритм БОЧОНОК на УАЯ с использованием команды Выбор

Адресс  
команды алг БОЧОНОК (нат таб  $B[1:4]$ , нат  $R$ , лит  $y$ )

```
арг  $B$ 
рез  $R$ ,  $y$ 
нач nat  $i$ 
     $R := 1$ ;  $i := 1$ 
    пока  $(R - 6666) * R - 9999 \neq 0$ 
        нц
            выбор
                04 — 07     при  $i = 1$ :  $B[1] := 6$ ;  $B[3] := 6$ 
                08 — 11     при  $i = 2$ :  $B[1] := 6$ ;  $B[2] := 6$ 
                12 — 23     при  $i = 3$ : если  $B[1] \neq B[3]$ 
                                то  $B[1] := 6$ ;  $B[3] := 6$ 
                                иначе  $B[1] := 9$ 
```

```
                24 — 39     при  $i = 4$ :  $B[1] := 15 - B[1]$ ;  $B[2] := 15 - B[2]$ 
                                иначе  $B[1] := 15 - B[1]$ ;  $B[3] := 15 - B[3]$ 
```

```
                все
                40 — 52     ГОРНЕР ( $4$ ,  $R$ ,  $B$ )
                выв  $R$ ; ввод  $\Pi_i$ 
                53 — 65     ВРАЩЕНИЕ ( $\Pi_i$ ,  $B$ )
                66          $i := i + 1$ 
```

кц  
 $y := \text{«задача решена»}$   
кон

Обратим внимание на одно принципиальное отличие в структуре программы и алгоритма на УАЯ. В алгоритме выход из пока-цикла совершается в начале цикла, если условие не выполнено, а в программе контроль и выход из до-цикла происходит в конце, по крайней мере один цикл ПМК всегда пройдет. Это обусловило необходимость ввода в алгоритм фиктивного вращения после пятоого хода, когда задача уже решена, его можно взять произвольным. Только после этого алгоритм вернет к началу цикла и обеспечит выход из него.

## ЗАДАНИЯ

1. Преобразуйте программу, чтобы она наряду с ситуациями, возникающими после ходов ПМК, индицировала также ситуации, возникающие после вращения, т. е. перед ходами ПМК.

2. В программе предусмотрено число поворотов  $k=1, 2, 3, 4$ . Дополните ее, чтобы она стала пригодной для любого  $k$ .

Указание. Введите в Р1 остаток от деления  $k$  на 4. Если остаток нуль, замените его четырьмя.

53. В↑ 54.4 55.: 56. X→PA 57.1 58.+ 59. X→P3 60. КП→X3  
61. P→XA 62. P→X3 63.— 64.4 65.× 66. FX=0 67.69 68.4 69. X→P1  
и т. д., как в нашей программе.

3. Для ПМК «Электроника МК-61», как и для МК-52 (ч. I, § 4.1) открывается перспектива использования в программе (см. задание 2) функции целая часть  $[x]$ , исключающей в совокупности с командой В↑, всю группу команд 56—62. Более того, можно воспользоваться имеющимся датчиком случайных чисел (с4) для генерирования числа поворотов бочонка: выдаваемые случайные числа из промежутка  $[0, 1]$  умножаются, например, на  $10^7$ . Впрочем, легко сформировать датчик случайных чисел из того же интервала и для других ПМК по формуле  $x_{n+1}=\arccos(\cos 10^9 x_n)/\pi$ , где начальное значение  $x_0$  произвольно ( $n=0, 1, 2, \dots$ ); можно также  $x_{n+1}=[10x_n+\pi]$  — дробная часть. Постройте программу «Бочонок» для МК-61.

4. Замените в алгоритме на УАЯ пока-цикл циклом с параметром  $i$ .

Замечание.  $i$  изменяется от 1 до 5; если  $R=6666$  или  $R=9999$ , то  $i:=5$ , и алгоритм «досрочно» выводит из цикла.

5. Вариация на тему «Бочонок». Отверстия в крышке бочонка находятся в вершинах правильного пятиугольника (рис. 3,б). Бочонок неподвижен, если по крайней мере в четырех отверстиях бутылки стоят одинаково. Коды по-прежнему 6 и 9. Постройте программу для решения задачи не более чем в 3 хода. В помощь обучающимся приводим алгоритм игры.

Ход 1. Фиксируются руками два любых отверстия, ход (6, 6). Если сумма  $S$  кодов всех пяти отверстий после хода удовлетворяет условию  $S=30$  или  $S=33$ , то задача решена, бочонок неподвижен.

Ход 2. Фиксируются два любых отверстия. Если коды равны, то они меняются противоположными: шестерки девятками и наоборот. Задача решена: по крайней мере в четырех отверстиях коды одинаковы.

Если коды не равны, то выполняется ход (6, 6), и три шестерки гарантированы. Однако в этом случае, возможно потребуется еще один ход.

Ход 3. Действия те же, что во втором ходе. Однако теперь задача решена.

Напомним: если в указании алгоритма содержится условие и оно не выполнено, то переходят к следующему указанию.

Постройте также алгоритм «Бочонок» — вариация на УАЯ.

## 1.2. ДОРОЖКА

— О великий царь! — сказала Шахразада в следующую ночь. — Вот мы с тобой и остановили бочонок. Не властью, а силой ума. Я рада, что победа разума по душе всемогущему. Тебе, конечно, ведома древняя притча про двух

козлов, которые встретились на узкой тропинке над пропастью. Как поступил бы величайший из царей, окажись он в положении одного из них?

— Я приказал бы сбросить упрямца в пропасть!

— Но это игра, а перед законами игры противники равны, и нет исключений для царей...

— Я растоптал бы эти законы!

— Твоими законами, о Великий, держится власть. Как может требовать их соблюдения человек, нарушающий другие законы? В игре, мы видели, побеждает мудрость.

И Шахразада поведала царю Игру на дорожке.

Играют двое: человек и ПМК, их фишки находятся на противоположных концах дорожки, разбитой на  $n$  клеток. Ходы делаются поочередно. Сделать ход — значит продвинуть фишку не более чем на  $m$  клеток вперед или назад, не выходя за пределы дорожки и не перескакивая через фишку противника. Проигрывает тот, у кого не окажется ходов. ПМК выигрывает независимо от значений  $n$  и  $m$ , если ему предоставлено право решать, кто делает первый ход;  $0 < m \leq n - 2$ .

Введем обозначения:  $p$  — длина хода ПМК, число пройденных им клеток;  $q$  — длина хода человека;  $S$  и  $P$  — алгебраические суммы длин всех ходов, сделанных человеком и ПМК от начала игры до данного момента. Ходу человека навстречу противнику присваивается знак +, в противоположном направлении —. Расстояние между противниками, изменяющееся в процессе игры, обозначим  $N$ , в начале игры  $N=n-2$ . Предлагаем следующий алгоритм игры.

1. Вычислить остаток  $r$  от деления  $N$  на  $m+1$ .

2. Если  $r=0$ , то первый ход предоставить человеку; в противном случае:  $p=r$  — первый ход ПМК.

3. На каждое  $q>0$  ПМК отвечает  $p=(m+1)-q$ , на  $q<0$  ответ  $p=-q$ .

Первую часть указания 2 можно перефразировать так: ПМК начинает всегда, но при  $r=0$  ему дано право сделать ход длиной нуль, т. е. пропустить ход.

Итак, ПМК своим первым ходом делает и в дальнейшем сохраняет переменное расстояние  $N$  между противниками кратным  $m+1$ . Это достигается после первого хода, отбрасывающего остаток, тем, что сумма длин двух последовательных ходов  $p+q=m+1$ . Через определенное число ходов  $N=m+1$ , и ход предстоит сделать человеку. Если окажется  $q>0$ , то после очередного хода ПМК  $N=0$ ; человек вынужденно отступает ( $q<0$ ), ПМК преследует его ходами такой же длины, пока «не прижмет» к краю дорожки. Если окажется, что  $q<0$  то  $p=-q$ , и предыдущее рассуждение повторяется.

Инвариантом, т. е. условием, сохраняемым ПМК в течение всей игры, является кратность переменного  $N$  числу  $m+1$  в ситуации, когда очередной ход предстоит сделать человеку.

Алгоритм нетрудно упростить: в первом ходе

$$p=r=N-[N/(m+1)](m+1), \quad (1)$$

в остальных ходах

$$p = \begin{cases} m+1-q, & \text{если } q>0, \\ -q, & \text{если } q<0. \end{cases} \quad (2)$$

Как видим, порознь и независимо производятся три типа ходов. Однако оказывается, что формула первого хода  $p=N-[N/(m+1)](m+1)$  пригодна для описания последующих ходов. Действительно, для сохранения переменного  $N$  кратным  $m+1$  ПМК каждый раз отбрасывает остаток от деления  $N$  на  $m+1$ , вычисляемый по этой формуле.

Алгоритм, положенный в основу программы, обобщает все рассуждения.

1. Начальное значение  $N=n-2$ .

2.  $p=N-[N/(m-1)](m+1); N:=N-p$  ( $N$  присваивается значение разности между предшествующим значением  $N$  и  $p$ ).

3. Проверка окончания. Если после хода ПМК окажется, что  $N=0$ , а человек занимает угловую клетку, то ПМК выиграл.

4. Индикация  $p$ ; ввод  $q$ .

5. Проверка, удовлетворяет ли  $q$  следующим четырем правилам игры:  $q \neq 0; |q| \leq m$ ; человек «не перепрыгнул» через ПМК;  $q$  не выводит за пределы дорожки.

6. Если хотя бы одно условие нарушено, то возврат к указанию 4, повторение индикации последнего  $p$ .

7.  $N=N-q$ . Возврат к указанию 2.

Напомним, что если в алгоритме содержится условие и оно не выполнено, то переходят к следующему указанию.

Для программы потребуются математические модели перечисленных правил игры. Между тем в указании 3 содержится «нечисловое» условие: человек занимает угловую клетку. Для его тематического выражения воспользуемся сумматором  $S$  — в нем, мы знаем, вычисляется алгебраическая сумма всех значений  $q$ , полученных до данного момента. Вначале  $S=0$ , при каждом ходе человека к предыдущему значению  $S$  прибавляется  $q$ , разумеется, со своим знаком. Условие  $S=0$  «в натуре» означает занятие человеком угловой клетки. Это возможно либо (как было сказано) в начале игры, либо если сумма всех отрицательных  $q$  равна сумме положительных; человек, отступая, «истратил» все накопленное в  $S$ .

Указание 5 также требует перевода на математический язык. Неравенство  $|q| (|q|-(m+1))<0$  означает одновременно  $q \neq 0$  и  $|q| \leq m$ . Действительно, из  $|q|-(m+1)<0$  следует  $|q| < m+1$ , откуда  $|q| \leq m$ . Далее,  $q < N$  означает, что человек не допустил нарушения типа «прыжок». Если  $N \neq 0$ , то это неравенство

выполняется автоматически вместе с ранее полученными  $|q| \leq m$ . Специальная проверка нужна лишь при  $N=0$ .

Осталось математически выразить условие выхода человека за пределы дорожки. При  $q>0$  выход заблокирован из-за невозможности «прыжка». Что касается  $q<0$ , то такое нарушение наблюдается при  $S+q < 0$  (в левой части сумма предшествующих значений  $q$  с его новым значением).

Теперь мы непосредственно подошли к составлению программы.

### Программа ДОРОЖКА

Распределение памяти

+P2 — вначале  $n$ , затем переменная  $N$  с начальным значением  $N=n-2$ ;

P3 — для вычисления целой части числа;

+P5 — сначала  $m$ , затем  $m+1$ , константа;

P6 — текущее значение  $q$  — длина очередного хода человека;

P7 — текущее значение  $p$  — длина хода ПМК;

P8 —  $S$  — алгебраическая сумма всех  $q$  от начала игры до данного момента, начальное значение  $S=0$ ;

PA — разность  $q-N$  для решения вопроса о нарушении типа «прыжок».

Начальное содержимое P2 и P5, отмеченных знаком «+», вводится вне программы, это позволяет при изменении входных данных не вносить изменений в программу.

00	CX	0Г	19	X→П7	47	38	П→Х6	66
01	X→П8	48	20	П→Х2	62	39	П→Х2	62
02	КП→Х2	Г2	21	П→Х8	68	40	—	11
03	КП→Х2	Г2	22	+	10	41	Х→ПА	4—
04	КП→Х5	Г5	23	FX=0	5E	42	FX<0	5C
05	П→Х2	62	24	27	27	43	27	27
06	П→Х5	65	25	Fπ	20	44	П→Х6	66
07	:	13	26	C/П	50	45	П→Х8	68
08	I	01	27	П→Х7	67	46	+	10
09	+	10	28	C/П	50	47	FX≥0	59
10	X→П3	43	29	X→П6	46	48	27	27
11	КП→Х3	Г3	30	FX <sup>2</sup>	22	49	Х→П8	48
12	П→Х3	63	31	FV—	21	50	П→ХА	6—
13	П→Х5	65	32	B↑	0E	51	/—/	OL
14	X	12	33	П→Х5	65	52	X→П2	42
15	П→Х2	62	34	—	11	53	БП	51
16	↔	14	35	×	12	54	06	06
17	X→П2	42	36	FX<0	5C	27		
18	—	11	37	27	27			

Инструкция  $n$  X→П2  $m$  X→П5 В/О С/П "p". После ввода данных и запуска программы индицируется первое значение  $p$ . Появление нуля означает, что ПМК отказался от хода, предоставленного его партнеру. Человек вводит свой ход  $q$ , после С/П на индикаторе высвечивается ответное  $p$ ; снова вводится  $q$  и т. д. Индикация  $\pi$  означает, что игра окончена. ПМК выиграл.

Если после ввода очередного  $q$  через 1—2 с повторилось последнее значение  $p$ , значит, человек нарушил правила игры. Необходимо ввести другое  $q$ . Повторения продолжаются до тех пор, пока  $q$  не окажется верным. Значение  $p$ , разумеется, может повториться и при верном  $q$ , если  $p$  последующее равно  $q$  предыдущему. Но в этом случае процедура вычислений  $p$  занимает более 10 с.

Контрольный пример.  $n=25, m=3, 25$  X→П2 3 X→П5 В/О С/П (10 с).

Номер хода	<i>q</i>	<i>p</i>	Комментарий
1		3	Первый ход за ПМК: остаток от деления 23 на 4
2	1 (15 с)	3	Из $q=1$ следует $p=4-1=3$
3	4 (4 с)	3	$q=4 > m=3$ , запрещенный ход, предыдущая индикация повторилась; в связи с этим значение <i>p</i> подчеркнуто
4	1 /—/ (15 с)	1	$q=-1$ — человек отступил на одну клетку, ПМК преследует его ходами такой же длины

Индикация π является сигналом о выигрыше ПМК. После нажатия С/П высвечивается последнее значение *p*.

### Структура программы

- 00—03: присвоение  $S=0$ ; вычисление  $N=n-2$ .  
 04—14: вычисление  $[N/(m+1)]$  и его запоминание в Р3; вычисление  $[N/(m+1)] \cdot (m+1)$ . Учитывая, что  $p=r=N-[N/(m+1)](m+1)$ , мы фактически нашли разность  $N-p$ , т. е. новое значение *N*.  
 15—19: вычисленное очередное значение *N* передается на хранение в Р2, вычисляется  $p=r$  и засыпается на хранение в Р7, остаток от деления  $N/(m+1)$ .  
 20—28: проверка окончания игры по условию  $N+S=0$ . Если после хода ПМК окажется  $N=0$  и одновременно  $S=0$ , т. е. его противник находится в угловой клетке, то, мы знаем, ПМК выиграл; индикация π означает завершение игры. Если игра не завершена, то на индикаторе очередное значение *p*, вызванное из Р7. Далее вводится *q*, и нажатием С/П запускается продолжение вычислений.  
 29—48: проверка соответствия *q* правилам игры. Контроль по условию  $|q|(|q|-(m+1))<0$ ; его нарушение означает ошибочный ход, повторяется индикация последнего *p* — сигнал к вводу нового значения *q* (команды 29—37). Блокируются ошибки человека типа «прыжок», одновременно в РА сохраняется разность  $q-N$ ; взятая с противоположным знаком, она в случае верного *q* выразит новое значение *N* (команды 38—43). Блокируется выход за пределы дорожки (команды 44—48).  
 49—52: значение *q* соответствует правилам игры, новые значения *S* направляются в Р8, а *N* — в Р2.  
 53, 54 : возврат к циклу вычисления остатка от деления  $N/(m+1)$ .

```

алг ДОРОЖКА (цел n, цел m, лог y)
    арг n, m
    рез y
    нач цел p, q, S
        S := 0; n := n - 2; m := m + 1; y := «неверно»
        пока n + s ≠ 0
            нач
                p := n - [n/m] * m; n := n - p
                выв p
                пока y = «неверно»
                    нач
                        ввод q
                        если ( $|q| * (|q| - (m + 1)) \geq 0$ )  $\vee ((q - n) \geq 0) \vee ((q + S) < 0)$ 
                            то y := «неверно»
                            иначе y := «верно»
                        все
                    кон
                    y := «неверно»
                    S := S + q
                    n := n - q
                кон
            кон
    кон

```

Примечание.  $\vee$  — знак логической дизъюнкции (ИЛИ). Служебные слова ВЫВ и ВВОД используются для организации диалога.

### ЗАДАНИЯ

1. Введите в программу учет числа сделанных человеком неверных ходов и наказание в виде проигрыша за определенное число нарушений правил игры. Сигналом нарушения мог бы служить ЕГГОГ, генерируемый, например, с помощью команды К—. Не забудьте, что после ЕГГОГ, полученного указанным способом, одна команда пропускается.
2. Исключите в программе «Дорожка» в расчете на «честного противника» ПМК команды проверки (29—48). Не забудьте после команды 28 сформировать новые значения *S* и *N*:  $S := S + q$ ,  $N := N - q$ .
3. Преобразуйте программу «Дорожка», пользуясь алгоритмом, приведенным в начале § 1.2: своим первым ходом ПМК по-прежнему отбрасывает остаток *r*, вычисленный по формуле (1); последующие ответные ходы ПМК строятся в соответствии с формулой (2). Теперь *N* постоянно:  $N = n - 2$ , *P* — сумма всех *p* (см. обозначения). Условием завершения игры является  $P = N$ . Для простоты исключите процедуру контроля *q*.
4. Игра НИМ. Две школы с равным основанием претендовали на комплект ПМК «Электроника МК-61». В роне сказали:

— Пусть представители школ поочередно берут по одной, две или три машинки не более, но каждый раз хотя бы одну. Взявшему последнюю достается весь комплект.

— Так это же известная задача, — сказал один. — Давненько не играл я в НИМ. Начинайте, коллега.

— Нет уж начните вы, я тоже знаю алгоритм.

— Раз так, — сказал инспектор, — пусть решает сама машина. Составьте каждый для нее программу с минимальным числом команд, по которой ПМК выигрывает у человека, если имеет право сделать первый ход или уступить его партнеру независимо от числа разыгрываемых машинок.

Включитесь и вы, читатель, в это увлекательное состязание. Однако прежде формулируем задачу в общем виде.

Имеются *n* машин. Каждый из двух игроков — человек, ПМК — берет любое число машин, но не более *m*. Ходят поочередно. Выигрывает тот, кто сде-

дал последний ход. ПМК предоставлено право решения вопроса о том, кто делает первый ход. Используя программу «Дорожка», составьте программу игры НИМ для ПМК.

Нетрудно заметить, что НИМ является упрощенным вариантом игры «Дорожка». Однако начальное значение  $N$  определено условием  $N=n$ . Проверка окончания производится по равенству  $N=0$ . Это избавляет от вычисления  $S$ , высвобождает Р8. Неравенство  $0 < q < m+1$  — это правила игры.

5. НИМ-модификация. Взявший последнюю машину проигрывает. Составьте программу для игры НИМ-модификация. Взяв за основу программу НИМ (см. задание 4), достаточно для нашего случая с самого начала значение  $N=n$  уменьшить на 1. Нетрудно показать, что, когда останется одна машина, последний ход сделает человек.

6. Игра «Фальшивая монета». Среди  $n$  одинаковых по достоинству монет одна фальшивая, отличная лишь по весу. Требуется с помощью весов без разновесов определить фальшивую монету наименьшим числом взвешиваний. (Под взвешиванием понимается помещение на чаши весов монет для сравнения по тяжести.) Сформулируйте алгоритм игры и постройте программу для ПМК, производящего разбиение монет на кучки.

ПМК запоминает количество монет в кучке, начиная с данной; делит ее на три по возможности равные части и индицирует число монет в каждой. Это наиболее трудная, «интеллектуальная» работа. Человек выполняет роль весов, «взвешивает» кучки, вводя в ПМК номер той из них, в которой находится фальшивая монета; ПМК запоминает количество монет в этой кучке и т. д. Ходы чередуются. Число взвешиваний  $v$  определяется по формуле

$$v = \begin{cases} \log_3 n, & \text{если } n \text{ — целая степень числа 3,} \\ \lceil \log_3 n \rceil + 1 & \text{в противном случае.} \end{cases}$$

Обратите внимание на распространенную ошибку: монеты обычно делят пополам. Легко убедиться, что в этом случае при  $n=32$  потребуется пять взвешиваний вместо четырех.

### 1.3. ДОРОЖКА, МОДИФИКАЦИЯ

— О, великий из сильных и сильный среди великих! Победа досталась царю познанием законов игры, но этого мало.

Ты мог начинать сближение сам или выжидать выхода противника навстречу из своей крепости. У противника такой возможности нет. Согласись, условия у вас неравные. Мудрецы говорят — разум великих сродни великодушию. Отдай свое преимущество противнику, и пусть победит мудрейший!

И не дождавшись ответа растерявшегося царя, Шахразада начала свой рассказ.

В рассмотренном ранее варианте игры «Дорожка» ПМК выигрывает всегда независимо от действий противника-человека. После нескольких партий человек обычно теряет интерес к игре, обнаружив, что результат неизменно не в его пользу.

Ниже предлагается модификация игры, в которой человек и ПМК как бы поменялись местами: человеку дана возможность выбора значений  $M$  и  $N$  и право первого хода. Владея алгоритмом, он, разумеется, выигрывает. Достаточно ему хотя бы один раз отклониться от алгоритма, и ПМК перехватывает инициативу, добивается выигрыша.

Человек и ПМК действуют по одному и тому же алгоритму. Если к началу очередного хода расстояние  $\rho$  между противниками не кратно  $M+1$ , то ход делается на величину остатка от де-

ления  $\rho$  на  $M+1$ . Противник, которому удалось выйти к такой позиции, придерживаясь алгоритма, выигрывает. В задаче это преимущество предоставлено человеку. Теперь все дело в том, знает ли он алгоритм и сумеет ли воспользоваться им.

Если же  $\rho$  кратно  $M+1$ , то делающий ход в этой позиции неизбежно проигрывает при последовательном использовании алгоритма противником. В нашей программе ПМК, оказавшись в этой ситуации, делает «выжидательные» ходы 1 или -1.

### Программа ДОРОЖКА, модификация

#### Распределение памяти

- P2 —  $\rho$  — переменное расстояние между противниками, начальное значение  $N-2$ ;  
P3 — оперативный регистр для вычисления целой части от деления  $\rho$  на  $M+1$ ;  
P4 — сумматор числа нарушений игры, допущенных человеком, начальное значение 0;  
P5 —  $M+1$  — константа;  
P6 —  $q$  — очередной ход человека;  
P7 —  $p$  — очередной ход ПМК;  
P8 — сумматор ходов ПМК  $S_1$ , начальное значение 0.  
P9 — сумматор ходов человека  $S_2$ , начальное значение 0.

00	$B \uparrow$	0E	28	+	10	56	1	01
01	2	02	29	$FX \geq 0$	59	57	БП	51
02	—	11	30	75	75	58	61	61
03	$X \rightarrow P2$	42	31	$X \rightarrow P9$	49	59	1	01
04	$\leftrightarrow$	14	32	$\Pi \rightarrow XA$	6—	60	/ — /	0L
05	$X \rightarrow P5$	45	33	$X \rightarrow P2$	42	61	$X \rightarrow P7$	47
06	$K\bar{P} \rightarrow X5$	45	34	$\Pi \rightarrow X8$	68	62	$\Pi \rightarrow X8$	68
07	$CX$	0Г	35	+	10	63	+	10
08	$X \rightarrow P4$	44	36	$FX = 0$	5E	64	$X \rightarrow P8$	48
09	$X \rightarrow P18$	48	37	39	39	65	$\Pi \rightarrow X2$	62
10	$X \rightarrow P9$	49	38	$C/P$	50	66	$\Pi \rightarrow X7$	67
11	$\Pi \rightarrow X7$	67	39	$\Pi \rightarrow X2$	62	67	—	11
12	$C/P$	50	40	$\Pi \rightarrow X5$	65	68	$X \rightarrow P2$	42
13	$X \rightarrow P6$	46	41	:	13	69	$\Pi \rightarrow X9$	69
14	$FX^2$	22	42	1	01	70	+	10
15	$F' \bar{V}$	21	43	+	10	71	$FX = 0$	5E
16	$\Pi \rightarrow X5$	65	44	$X \rightarrow P3$	43	72	11	11
17	—	11	45	$K\bar{P} \rightarrow X3$	Г3	73	$F\pi$	20
18	$FX < 0$	5C	46	$\Pi \rightarrow X2$	62	74	$C/P$	50
19	75	75	47	$\Pi \rightarrow X5$	65	75	$K\bar{P} \rightarrow X4$	Г4
20	$\Pi \rightarrow X2$	62	48	$\Pi \rightarrow X3$	63	76	$\Pi \rightarrow X4$	64
21	$\Pi \rightarrow X6$	66	49	×	12	77	3	03
22	—	11	50	—	11	78	—	11
23	$X \rightarrow P4$	4—	51	$FX = 0$	5E	79	$FX \geq 0$	59
24	$FX \geq 0$	59	52	61	61	80	12	12
25	75	75	53	$\Pi \rightarrow X2$	62	81	$F\pi$	20
26	$\Pi \rightarrow X9$	69	54	$FX \neq 0$	57	82	$C/P$	50
27	$\Pi \rightarrow X6$	66	55	59	59			

Инструкция.  $M \leq N \leq B/O \leq C/P$ . При первом останове — через 3 с после  $B/O \leq C/P$  — вводится начальный ход человека; при других — индицируется очередной ход ПМК и вводится ответный ход человека. В любом случае нажатием клавиши  $C/P$  включается продолжение вычислений. Время между

двея индикации порядка 20 с. Если ПМК останавливается через 5 с после С/П, то это означает, что введенный ход человека ошибочный — не отвечает правилам игры. В этом случае необходимо ввести другой ход. Индикация 0 означает, что ПМК проиграл, индикация восьми знаков π — выиграл.

#### Контрольные примеры.

1.  $M=3, N=17$ . 3 В $\uparrow$  17 В/О С/П (3 с) «0».

Человек: 3 С/П (17 с) — ПМК «1». Человек: 3 С/П (20 с) — ПМК «1». 3 С/П (20 с) — «1». 3 С/П (20 с) — «-1». 2 С/П (5 с) — «-2» — ошибочный ход, человек вводит другой ход: 1 С/П (20 с) — «-1» 5 С/П (5 с) — «-1» — ошибочный ход человека, ввод другого хода: 1 С/П (20 с) — «-1» 1 С/П (7 с) — «0», т. е. ПМК проиграл.

2.  $M=3, N=17$ . 3 В $\uparrow$  17 В/О С/П (3 с) «-1».

Человек: 1 С/П (17 с) — ПМК «2». Человек в первом ходе допустил ошибку, создалась ситуация, выигрышная для ПМК, и ПМК сразу же воспользовался этим, сделав ход — 2. После этого расстояние между противниками уменьшилось: 15—3=12. Число 12 кратно  $M+1=4$ , и ход предстоит сделать человеку. Следуя алгоритму, ПМК выигрывает.

#### Структура программы

00—12: человек вводит значения  $M$  и  $N$ , ПМК запоминает эти числа в соответствующих регистрах памяти, формирует начальные данные игры и останавливается для ввода первого хода.

13—30: проверка введенного хода на соответствие условию задачи  $q \leq M$  — длина хода не превышает максимально допустимой (команды 13—18);  $q \leq p$  — ход не более расстояния между противниками, т. е. человек «не перепрыгнул» через фишку ПМК (команды 20—24);  $S_2 + q \geq 0$  — человек, двигаясь назад, не вышел за пределы дорожки (команды 26—29).

При нарушении условия любого типа к содержимому сумматора прибавляется единица (команда 75). Если содержимое сумматора равно 3, то за ошибочную игру ПМК объявляет о поражении противника индикацией восьми знаков числа π (команды 76—82). В противном случае ПМК предложит сделать другой ход (команда 80).

31—38: путь хода человека удовлетворяет требованию задачи, тогда ПМК запоминает возникшую ситуацию: новое значение  $S_2$  в Р9 (команда 31), расстояние  $p$  между противниками после хода в Р2 (команды 32, 33). Далее проверяется наличие выигрыша у человека по условию  $p+S_1=0$ , т. е.  $p=S_1=0$ , когда ПМК «прижат» к своему краю (команды 34—36). Если это условие выполнено, то ПМК индикацией 0 объявляет о своем поражении (команда 38); если не выполнено, переходит к построению очередного хода (команда 37).

39—61: вычисление остатка от деления текущего значения  $p$  на  $M+1$ . Если он равен нулю (ситуация выигрышная для ПМК), то при  $p>0$  ПМК сдвигнет фишку на 1, при  $p=$

=0 — на —1. Если же остаток не нуль (ситуация выигрышная для ПМК), ПМК делает ход на величину остатка. В любом случае ход ПМК запоминается в Р7 (команды 51—61).

62—68: преобразование позиции в связи с очередным ходом ПМК — сумматор Р8 учитывает длину очередного хода (команды 62—64); на столько же уменьшается расстояние  $p$  между противниками (команды 65—68). В сложившейся ситуации проверяется наличие выигрыша у ПМК по условию  $p+S_2=0$  или  $p=S_2=0$  — человек «прижал» к своему краю.

69—74: если условие выполнено, то ПМК индикацией π объявляет о своем выигрыше, в противном случае управление передается команде 11, ПМК индицирует свой ход, предоставив человеку ввести ответный ход и т. д.

#### ЗАДАНИЯ

1. В программе предусмотрено, что при остановах, не завершающих игру, т. е. без индикации 0 или π, выводится очередной ход ПМК и выводится ответный — человека. Исключением является первый останов, сразу же после нажатия В/О С/П. В этом случае, вообще говоря, на индикаторе будет любое число, в первой игре — 0, который можно ошибочно принять за проигрыш ПМК. Внесите изменения в программу, чтобы она при первом останове выдавала число  $10^6$ .

2. Если игра завершается в пользу ПМК, индицируется π, последний ход ПМК на индикаторе не выводится. Дополните программу, чтобы можно было после индикации π (команды 73, 74) нажатием клавиши С/П вывести последний ход ПМК (вспомните — очередной ход ПМК запоминается в Р7).

3. Согласно программе за три допущенные ошибки человеку присуждается поражение. Желатель более гибкий критерий проигрыша за нарушение правил игры. Внести изменение в программу и начальный ввод, чтобы можно было варьировать число ошибок, за которое присуждается поражение.

4. В нашей программе человеку дано право первого хода. Для уравнения условий игры человека и ПМК необходимо «справедливое» решение вопроса об исполнителе первого хода генерированием случайного числа. На ПМК «Электроника МК-61» для этого используется оператор СЧ, для других ПМК можно использовать фрагмент, программирующий формулу  $\text{arccos}(\cos 10^\circ x)/\pi$  для произвольного  $x$ . В любом случае индицируется число  $x$  из интервала  $0 < x < 1$ . Остаток от деления его первого десятичного знака на два будет нуль или единица. В первом случае начальный ход за человеком, во втором — за ПМК. Внесите соответствующие изменения в программу.

5. Постройте на УАЯ алгоритм решения задачи «Дорожка», модификация.

6. Игра НИМ, усложненный вариант. Кучка состоит из  $N$  предметов. Как обычно, играют человек и ПМК, делая ходы поочередно; максимальная длина хода  $M < N$ . Сделать ход — значит взять из кучки любое число предметов ( $q$  — человек,  $p$  — ПМК) от 1 до  $M$ . В известный вариант игры вводится дополнительное ограничение: очередной ход не должен повторять предшествующий ход противника. Проигрывает тот, у кого не оказалось ходов: кучка разобрана либо остался один предмет, если в предшествующем ходе соперник также взял один предмет. Построить усложненный вариант программы НИМ. В помощь обучающимся приводится алгоритм игры.

Ограничение усложняет задачу. Так, если  $q$  — ход человека, то ПМК, как и ранее, отвечает  $p=(M+1)-q$ , если  $M$  четно. В противном случае, если  $q=$

$= (M+1)/2$ , будет  $p=q$ , такой ход теперь недопустим. В связи с этим рассмотрим несколько случаев.

1.  $M$  четно. Обладая правом решения вопроса о том, кто первым вступает в игру, ПМК, как обычно, ходит  $p$ , равное остатку от деления  $N$  на  $M+1$ , где  $N$  число невыбранных предметов. Начальное  $p=0$  означает, что первый ход уступается сопернику.

2.  $M$  нечетно. Введем понятие показателя кратности числа  $k$  двум. Показатель кратности числа двум равен наибольшему  $k$ , при котором это число кратно  $2^k$ , например  $k(24)=3$ , так как  $24=2^3 \cdot 3$ .

2.1. Пусть  $k$  для  $M+1$  четно. ПМК вначале делает ход на величину остатка от деления  $N$  на  $M+1$ . Если ответное  $q=(M+1)/2$ , то  $p=q/2$ ; при последующем  $q=p/2$  ПМК ходит  $p=q/2$  и т. д. Если  $q \neq (M+1)/2$  или в дальнейшем  $q \neq p/2$ , то  $p$  равно остатку от деления  $N$  на  $M+1$  ( $N$  — число невыбранных предметов).

2.2. Пусть  $k$  для  $M+1$  нечетно. ПМК ходит на величину остатка от деления  $N$  на  $M+2$ . Если окажется  $q=1$ , то  $p=(M+1)/2$  и далее на каждое  $q=p/2$  ответ  $p=q/2$ . Если  $q \neq p/2$  или (вначале)  $q \neq (M+1)/2$ , то  $p$  равно остатку от деления  $N$  на  $M+2$ . Таким же ходом ПМК отвечает на  $q \neq 1$ .

### Программа НИМ, усложненный вариант

Распределение памяти

РА, РВ, РД — значения  $p$ ,  $q$ ,  $N$ ;

Р6 —  $M+1$ ;

Р7—Р9 — параметры обращения к подпрограммам.

00	КПП7	-7	33	X→P1	41	66	БП	51
01	FX ≠ 0	57	34	С/П	50	67	41	41
02	08	08	35	КПП9	-9	68	П→X3	63
03	КПП8	-8	36	БП	51	69	П→XC	6C
04	ПП	53	37	22	22	70	:	13
05	85	85	38	КПП8	-8	71	X→P0	40
06	БП	51	39	БП	51	72	1	01
07	03	03	40	20	20	73	+	10
08	КП→X5	Г5	41	П→X6	66	74	X→P2	42
09	П→X0	60	42	1	01	75	КП→X2	Г2
10	X→P3	43	43	+	10	76	П→X3	63
11	КПП7	-7	44	ПП	53	77	П→XC	6C
12	FX ≠ 0	57	45	84	84	78	П→X2	62
13	08	08	46	П→XB	6L	79	×	12
14	П→X5	65	47	1	01	80	-	11
15	X→P3	43	48	-	11	81	X→PA	4—
16	КПП7	-7	49	FX = 0	5E	82	В/О	52
17	FX = 0	5E	50	41	41	83	П→X6	66
18	41	41	51	П→X6	66	84	Х→PC	4C
19	КПП8	-8	52	Х→PB	4L	85	П→XD	6Г
20	П→X6	66	53	П→XB	6L	86	Х→P3	43
21	X→P1	41	54	2	02	87	КПП7	-7
22	П→XB	6L	55	:	13	88	С/П	50
23	П→X1	61	56	X→PA	4—	89	КПП9	-9
24	2	02	57	С/П	50	90	В/О	52
25	:	13	58	КПП9	-9	91	Х→PB	4L
26	-	11	59	П→XB	6L	92	+	10
27	FX = 0	5E	60	П→XA	6—	93	П→XD	6Г
28	38	38	61	2	02	94	↔	14
29	П→XB	6L	62	:	13	95	-	11
30	2	02	63	-	11	96	X→PD	4Г
31	:	13	64	FX ≠ 0	57	97	В/О	52
32	X→PA	4—	65	53	53			

Инструкция. 68 X→P7, 83 X→P8, 91 X→P9, N X→PD, M+1 X→P6, X→P3, 2 X→PC, СХ X→P5 В/О С/П. При каждой индикации выводится значение  $p$  — ход ПМК. Если в создавшейся ситуации у человека нет ответных ходов — все предметы выбраны либо остался один предмет, а предшествующее  $p=1$ , то он сдает партию. В противном случае человек вводит ход  $q$  и, нажав С/П, продолжает вычисления.

### Контрольные примеры

1.  $N=31$ ,  $M=11$  (нечетно),  $k=2$  (четно):  $M+1=12=2^2 \cdot 3$ . 68 X→P7, 83 X→P8, 91 X→P9, 31 X→PD, 12 X→P6, X→P3, 2 X→PC, СХ X→P5 В/О С/П. Запись параметров в память производится только при решении первой задачи.

(37 с) «7», 6 С/П (8 с) «3», 2 С/П (13 с) «1», 4 С/П (15 с) «8». ПМК выиграл:  $7+6+3+2+1+4+8=31$ .

2.  $N=43$ ,  $M=7$  (нечетно),  $k=3$  (нечетно):  $M+1=8=2^3$ . 43 X→PD, 8 X→P6 X→P3, 2 X→PC СХ X→P5 В/О С/П.

(47 с) «7» (остаток от деления  $N=43$  на  $M+2=9$ ) 1 С/П (7 с) «4», 2 С/П (7 с) «1», 6 С/П (15 с) «4», 7 С/П (15 с) «2», 1 С/П (7 с) «4», 2 С/П (7 с) «1». Контроль:  $7+1+4+2+1+6+4+7+2+1+4+2+1=42$  — остался один предмет, но человек его взять не может: предшествующий ход ПМК 1. Человек проиграл.

### Структура программы

Основу составляют три подпрограммы. Подпрограмма 1 служит для вычисления и сохранения в Р8 целой части, а в РА и РХ остатка от деления содержимого РЗ на содержимое РС (команды 68—82). Подпрограмма 2 выполняет роль «диспетчера»: подготовливает в РЗ и РС данные для обращения к подпрограмме 1; в ней выводится очередное значение  $p$ , вводится ответное  $q$  и обращением к подпрограмме 3 вычисляется количество невыбранных предметов, которое запоминается в РД. После выполнения команды 00 в РА и РХ хранится остаток от деления  $M+1$  на 2, в Р0 — частное, Р5 подготовлен для подсчета показателя кратности числа  $M+1$  двум. Пусть  $M+1$  нечетное, т. е.  $k=0$ , тогда в качестве  $p$  берется остаток от деления  $N$  на  $M+1$ . Одновременно в подпрограмме 3 вычисляется число оставшихся предметов, в качестве нового значения  $N$  оно хранится в РД.

Игра проходит в цикле: на каждое значение  $q$  ПМК отвечает  $p$ , равным остатку от деления  $N$  на  $M+1$  (команды 03—07). Анализируются случаи нечетного  $M$  (команды 08—67). Вычисляется значение  $k$ , которое запоминается в Р5 (команды 08—13), число  $k$  проверяется на четность (команды 14—16). Команды 17—19 — первый ход для четного  $k$ , команды 20—40 — остальные ходы. Точнее, вычисляется  $p$  при  $q=(M+1)/2$  (команды 20—37), а при  $q \neq (M+1)/2$  (команды 38—40). Команды 41—67 — случай нечетного  $k$  при нечетном  $M$  для  $q \neq 1$  (команды 41—50) и для  $q=1$  (команды 51—67).

## 1.4. ОДИН ФЕРЗЬ И ДВА ИГРОКА

— Сегодня играем в чатранг\*, — сказал царь, едва наступил вечер. — В этой игре, мудрая Шахразада, тебе со мной не совладать!

— Принимаю вызов, великий царь! Но только предлагаю не обычную игру. В моем чатранге участвует только один ферзан (ферзь), играют им оба противника поочередно, по особым правилам. Вот послушай, мой повелитель...

Приоединимся и мы, любознательный читатель.

На восьмой горизонтали шахматного поля находится «ферзь», которому разрешено перемещаться обычными ходами ферзя из занятой клетки, но только по горизонтали влево, по вертикали, вниз, по диагонали влево и вниз. Противники (человек и ПМК) ходят «ферзем» поочередно. Выигрывает тот, кто займет клетку в левом нижнем углу, после чего у противника не окажется ходов (рис. 5).

Для удобства программирования нумерация игрового поля начинается с нуля. Условимся начальную и последующие позиции «ферзя» обозначать координатами занятой клетки — номерами столбца и строки. Например, координаты (4; 7) обозначают, что после очередного хода игрок занял клетку на пересечении 4-го столбца и 7-й строки. Нетрудно догадаться: чтобы занять конечную позицию (0; 0), ПМК должен оказаться в одной из следующих позиций: (2; 1), (1; 2) либо (3; 5), если первые две недостижимы. Отсюда следует, что если «ферзь» вначале занимает клетку (4; 7), из которой ни одна из перечисленных или конечная клетка недостижимы, то первый ход предоставляемся человеку. Во всех остальных случаях первый ход делает ПМК, занимая одну из выигрышных позиций.

Условимся обозначать переменные координаты клетки, занимаемой «ферзем», ( $a$ ;  $b$ ) и сформулируем алгоритм игры.

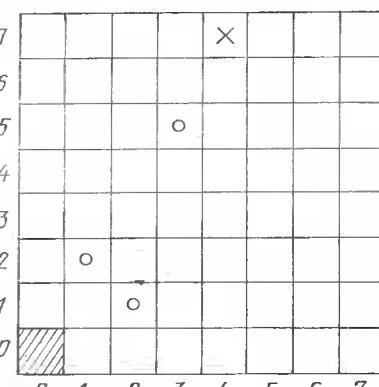
1. Если  $a=4$ , то первый ход делает человек, в противном случае — ПМК.

2. Если в сложившейся ситуации  $a=0$  или  $b=0$  или  $a=b=0$ , то ПМК занимает клетку (0; 0). Задача решена.

3. При  $a=1$  или  $b=2$  или  $b-a=1$  ПМК занимает клетку (1; 2); после очередного хода человека — возврат к указанию 2.

4. При  $a=2$  или  $b=1$  или  $a-b=1$  ПМК занимает клетку (2; 1), после хода человека — возврат к указанию 2.

5. Если ни одно из предыдущих условий не выполнено, то



Rис. 5

\* Название шахмат в Иране.

ПМК занимает клетку (3; 5); после хода человека — возврат к указанию 2.

## Программа ФЕРЗЬ

Распределение памяти  
РА, РВ — начальные координаты клетки, занимаемой «ферзем» на седьмой горизонтали, в дальнейшем — текущие координаты позиции «ферзя».

00	X→ПВ	4L	23	X→ПС	4C	45	50	50
01	↔	14	24	П→ХВ	6L	46	3	03
02	X→ПА	4—	25	X→ПД	4Г	47	5	05
03	4	04	26	ПП	53	48	БП	51
04	—	11	27	53	53	49	07	07
05	FX = 0	5E	28	FX = 0	5E	50	С/П	50
06	11	11	29	34	34	51	БП	51
07	С/П	50	30	1	01	52	20	20
08	X→ПВ	4L	31	2	02	53	П→ХС	6С
09	↔	14	32	БП	51	54	1	01
10	X→ПА	4—	33	50	50	55	—	11
11	П→ХА	6—	34	П→ХА	6—	56	П→ХД	6Г
12	П→ХВ	6L	35	X→ПД	4Г	57	2	02
13	—	11	36	П→ХВ	6L	58	—	11
14	П→ХА	6—	37	X→ПС	4C	59	Х	12
15	П→ХВ	6L	38	ПП	53	60	П→ХД	6Г
16	×	12	39	53	53	61	П→ХС	6G
17	×	12	40	FX = 0	5E	62	—	11
18	FX = 0	5E	41	46	46	63	1	01
19	22	22	42	2	02	64	—	11
20	Fπ	20	43	1	01	65	Х	12
21	С/П	50	44	БП	51	66	В/О	52
22	П→ХА	6—						

Инструкция  $a \text{ В} \uparrow b \text{ В/О С/П}$ , где  $a$  и  $b$  — столбец и строка клетки, занимаемой «ферзем» в начальной позиции на седьмой горизонтали. Предусмотрены индикации трех типов: 0 означает, что первый ход предоставлен человеку; π — ПМК выиграл; двузначное число (первая цифра — абсолютна клетки, вторая — ордината) — позиция, занятая ПМК в очередном ходе.

После индикации, за исключением π, человек вводит свой ответный ход:  $a \text{ В} \uparrow b \text{ С/П}$  и т. д., пока сигналом π не будет сообщено о завершении игры выигрышем ПМК.

Контрольный пример. 5 В $\uparrow$  7 В/О С/П (20 с) «35» 3 В $\uparrow$  1 С/П (20 с) «21» 1 В $\uparrow$  С/П (1 с) «π».

## Структура программы

- 00—07 : указание 1 алгоритма;
- 08—21 : указание 2 алгоритма;
- 22—33 : указание 3;
- 34—45 : указание 4;
- 46—52 : указание 5;
- 53—66 : подпрограмма.

Адрес  
команды

алг ФЕРЗЬ (нат  $a, b$ , лит  $A$ )  
арг  $a, b$   
рез  $A$   
нач нат  $x, y$ , цел  $k$   
ком  $a, b$  — координаты начальной позиции ферзя и клетки, занятой ферзем после хода ПМК, выводятся  
 $x, y$  — координаты клетки, занятой ферзем после хода человека, вводятся  
если  $a = 4$   
то ввод  $x, y$   
иначе  $x := a; y := b$   
все  
 $k := 0$   
пока  $k = 0$   
нц  
выбор  
при  $(x=0) \vee (y=0) \vee (x-y=0)$ :  $a := 0; b := 0; k := 1$   
при  $(x=1) \vee (y=2) \vee (y-x=1)$ :  $a := 1; b := 2$   
при  $(x=2) \vee (y=1) \vee (x-y=1)$ :  $a := 2; b := 1$   
иначе  $a := 3; b := 5$   
все  
выв  $a, b$   
если  $k = 0$   
то ввод  $x, y$   
все  
кц  
 $A := \text{«ПМК выиграл»}$   
кон

00—07  
08—21  
22—33, 53—66  
34—45, 53—66  
46—52

### ЗАДАНИЯ

1. Используя симметрию клеток  $(2; 1)$  и  $(1; 2)$ , совместите программно команды, реализующие указания 3 и 4 алгоритма. Проверьте, сократится ли при этом программа.

2. Игра «Двое и король». Задача прежняя, только решается она ходами короля. Алгоритм игры:

2.1. Если обе координаты позиции четны, то ход предоставляется человеку, в противном случае — ПМК.

2.2. ПМК передвигает короля на клетку с четными координатами.

2.3. Если координаты позиции, занятой ПМК в очередном ходе,  $(0; 0)$ , то игра завершена, ПМК выиграл.

2.4. Ход человека. Возврат к указанию 2.

3. Интерпретация игры «Двое и король» (см. задание 2). Имеются две кучки предметов. Сделать ход — значит взять один предмет из любой кучки либо по одному из каждой. Ходы делаются человеком и ПМК поочередно. Выигрывает тот, кто взял последние предметы. ПМК выигрывает независимо от начального числа предметов в кучках, если ему предоставлено право решать, кто сделает первый ход. Составьте программу игры.

Замечание. Нетрудно заметить, что здесь переформулирована задача «Король»: исходное число предметов в первой кучке — абсцисса, во второй — ордината начальной позиции «короля». Взятие ПМК последнего предмета означает занятие «королем» позиции  $(0; 0)$ .

### 1.5. ХАНОЙСКАЯ БАШНЯ

— О мудрейший из мудрых! — начала Шахразада. — Ты выиграл на долгое знаний, познал радость от игры в мой чатранг. Это нелегко, но, согласись, такая победа дает душе истинное наслаждение. В предшествующих играх ты одержал победу, но впереди — новая игра. Слушай же, великий царь!

Послушаем и мы с тобой, читатель. Много у нас еще будет задач, в которых человек и машина помогают друг другу. Но эта — первая.

Требуется перенести кольца со стержня 1 на стержень 2, используя в качестве вспомогательного стержень 3 (рис. 6). Обязательно выполнение условия, чтобы в процессе игры ни на одном стержне большее кольцо не лежало поверх меньшего. Задачу необходимо решить с помощью минимального числа ходов.

Условимся об обозначениях:  $S_i^-$  — кольцо снимается с  $i$ -го стержня;  $S_i^+$  — кольцо надевается на  $i$ -й стержень ( $i=1, 2, 3$ ). Предлагаем следующий алгоритм:

1. Ход начинается с нахождения стержня  $i$  для выполнения действия  $S_i^-$ . Вначале  $i=1$ . В каждом из последующих ходов стержень, на который перенесено кольцо в предыдущем ходе, исключается. Из колец на двух оставшихся стержнях снимается меньшее; если один из этих стержней свободен, кольцо снимается с другого.

2. Выбор стержня, на который надевается кольцо в данном ходе.

Пусть кольцо снято со стержня с номером  $N$ ,  $n$  — количество колец на этом стержне ( $N=1, 2, 3$ ;  $n$  произвольно). Если  $n$  четно, то при  $N=1$  выполняется  $S_3^+$ , при  $N=2$  или  $N=3$   $S_1^+$ . Если  $n$  нечетно, то при  $N=2$  выполняется  $S_3^+$ , при  $N=1$  или  $N=3$   $S_2^+$ .

Поясним сказанное схемой на рис. 7.

Пример пользования алгоритмом.  $N=1, n=3$ : на стержне 1 три кольца.

Ход 1.  $N=1, n=3 : S_2^+$ .

Ход 2. Исключаем стержень 2 по правилу 1 алгоритма. Из оставшихся стержней 1 и 3 выбрать нетрудно; стержень 3 пусту-

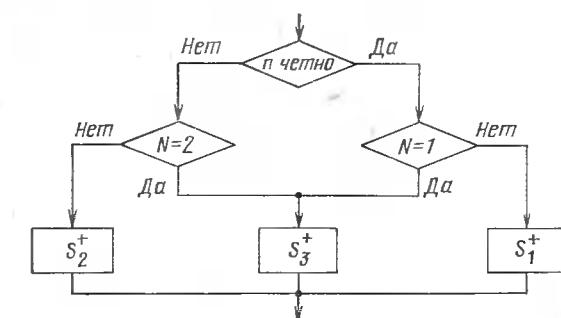


Рис. 7

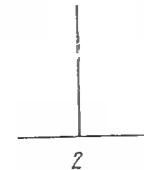
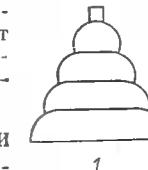


Рис. 6

ет, значит, по правилу 1 алгоритма выбираем стержень 1, действие  $S_1^-$ . На стержне 1 два кольца (четное число),  $S_3^+$ .

Ход 3. На стержне 2 кольцо меньшее, чем на стержне 1,  $N = 2$ ,  $S_2^-$ ;  $n=1$  (нечетно),  $S_3^+$ .

вие  $S_1^-$ . На стержне 1 два кольца (четное число),  $S_3^+$ .

Ход 5.  $N=3$ : стержень 1 пуст,  $S_3^-$ ;  $n=2$  (четно) на стержне 3 два кольца,  $S_1^+$ .

Ход 6.  $N=3$ ,  $n=1$ ,  $S_3^-$ ,  $S_2^+$ .

Ход 7.  $N=1$ ,  $n=1$ ,  $S_1^-$ ,  $S_2^+$ .

Функции между человеком и ПМК распределяются следующим образом. Человек реализует указание 1 без труда. Глядя на картинку или «натуру», он просто визуально, без раздумий находит стержень, с которого снимается кольцо, выполняет  $S_i^-$ . С калькулятором дело обстоит иначе. Автомат «не видит»: требуется специальная процедура формирования и запоминания номеров верхних колец и т. д.

Указание 2, напротив, требует большого умственного напряжения человека, но зато несложно описывается в программе для ПМК.

Чтобы организовать диалог в игре, распределим функции между ПМК и человеком. Человек в начале хода вводит в ПМК номер стержня, с которого снимается кольцо; нажатием С/П запускает программу; визуально проверяет окончание решения. ПМК вычисляет и индицирует номер стержня, на который надевается кольцо в данном ходе. Итак, ход описывается двумя цифрами: первая — номер стержня, с которого снимается кольцо, вводится человеком; вторая — номер стержня, на который переносится кольцо, генерируется ПМК. Диалог, таким образом, взаимовыгодный.

### Программа ХАНОЙСКАЯ БАШНЯ

00	X→P1	41	17	31	31	33	KП→XA	Г—
01	CX	0Г	18	1	01	34	1	01
02	X→P2	42	19	X→ПВ	4L	35	—	11
03	X→P3	43	20	БП	51	36	КХ→ПА	L—
04	1	01	21	33	33	37	КП→XB	ГL
05	X→PA	4—	22	П→XA	6—	38	1	01
06	KП→XA	Г—	23	2	02	39	+	10
07	9	09	24	—	11	40	KХ→ПВ	LL
08	0	00	25	FX≠0	57	41	П→XB	6L
09	×	12	26	31	31	42	П→XA	6—
10	F sin	1C	27	2	02	43	ВП	0C
11	FX=0	5E	28	X→ПВ	4L	44	1	01
12	22	22	29	БП	51	45	+	10
13	P→XA	6—	30	33	33	46	C/П	50
14	1	01	31	3	03	47	БП	51
15	—	11	32	X→ПВ	4L	48	05	05
16	FX≠0	57						

Инструкция. Переключатель углов на Г.  $n$  В/О С/П (12 с).  $n$  — число колец на 1-м стержне при  $n$  нечетном индицируется 12, при  $n$  — четном — 13.

Контрольный пример. При  $n=5$

5 В/О С/П (12 с) «12» 1 С/П (11 с) «13» 2 С/П (11 с) «23» и т. д., пока человек на рисунке или модели не увидит, что задача решена.

### Структура программы

```

00—03 : ввод начальной позиции — числа колец на стержнях
        1—3.
04—11 : Проверка на четность числа колец  $n$  на стержне, с которого
        человек решил снять кольцо.
12—16 :  $n$  четно; проверка, будет ли выбран стержень 1.
17—21 :  $n$  четно,  $N \neq 1$ .
22—25 :  $n$  нечетно; проверка, будет ли выбран стержень 2.
26—30 :  $n$  нечетно,  $N \neq 2$ .
31, 32 :  $n$  четно,  $N=1$  или  $n$  нечетно,  $N=2$ . После этих команд в
        РВ хранится номер стержня, на который надевается кольцо.
33—36 : выполнение действия  $S_i^-$  в очередном ходе ( $i=1, 2, 3$ ).
37—40 : выполнение  $S_j^+$  в очередном ходе ( $j=1, 2, 3; j \neq i$ ).
41—46 : формирование и индикация хода двузначным числом:
        первая цифра — номер стержня, с которого снимается кольцо, вторая — номер стержня, на который оно надевается.
47, 48 : возврат к началу цикла вычислений.

Адреса алг ХАНОЙСКАЯ БАШНЯ (нат  $n$ , лит  $y$ )
команд арг  $n$ 
      рез  $y$ 
      нач цел таб  $A[1:3]$ , нат  $N$ ,  $p$ , цел  $C$ ,вещ  $B$ 
       $A[1]:=n; A[2]:=0; A[3]:=0$ 
      пока  $A[2]<n$ 
      иц
      ввод  $N; B:=A[N]/2; C:=B; A[N]:=A[N]-1$ 
      выбор
          при ( $B=C$ )  $\wedge ((N=2) \vee (N=3))$ :  $p:=1$ 
          при ( $B \neq C$ )  $\wedge ((N=1) \vee (N=3))$ :  $p:=2$ 
          иначе  $p:=3$ 
          все
           $A[p]:=A[p]+1; p:=10*N+p$ ; выв  $p$ 
      кон
       $y:=«задача решена»$ 

```

Примечания.  $\wedge$ ,  $\vee$  — логическое умножение (конъюнкция) и логическое сложение (дизъюнкция); ввод, выв — команды ввода и вывода, необходимые для оперативного обмена информацией в диалоге между человеком и ПМК;  $N$  — номер стержня, с которого снимается кольцо, находится человеком;  $p$  — номер стержня, на который переносится кольцо, вычисляется ПМК. При завершении хода  $p$  — двузначное число: первая цифра — значение  $N$ , вторая — предшествующее значение  $p$ ; в таком виде ход выводится.

## ЗАДАНИЯ

1. В нашей программе вопрос об окончании решения визуально решает человек. Внесите изменения, чтобы ПМК сам решал этот вопрос, информируя об этом числом  $\pi$ . (После ввода очередного хода проверяется содержимое Р1 и Р3. К моменту окончания игры на стержнях 1, 3 колец нет.)

2. ПМК индицирует ходы двузначными числами. Программа, конечно, упростится, если после ввода человеком номера стержня, с которого снимается кольцо, ПМК будет сообщать номер стержня, на который оно переносится. Внесите соответствующие изменения в программу. (см. команды 41—46).

## ПОСЛЕСЛОВИЕ К ГЛАВЕ

Мы запрограммировали решение четырех игровых задач. Подумаем, какова роль человека в каждой из них.

В задаче «Бочонок» человек по существу является датчиком случайных чисел — начальной ситуации и количества поворотов бочонка. Его роль пассивна. В задачах «Дорожка» и «Ферзы» человек выступает противником ПМК — выигрыш ПМК одновременно является проигрышем человека. Но если он и оказался «прижатым к краю», то в этом не слабость человека, а сила алгоритма, секретом которого «владеет» ПМК. В задаче «Ханойская Башня» человек и ПМК уже союзники во взаимовыгодном диалоге. Этот тип задач особенно интересен: он является игровой моделью реальных жизненных ситуаций — люди объединяются в коллективы для выполнения работы, недоступной или трудной для каждого в отдельности. Во взаимодействии человека и ЭВМ кроются огромные возможности для развития техники, производства, экономики, науки.

— Чего только не бывает! — начал сам царь в следующую ночь. — Ты мудрая женщина, я отдаю тебе все свои сокровища, если скажешь, чего в жизни все-таки не бывает?

— Нет и не будет, о великий царь, в мире людей богаче тебя, — молвила Шахразада.

— А ведь это правда... — подумал царь. — Но тогда я должен отдать ей все богатство... Она станет богаче меня... Значит, она все же соглашается, и отдавая богатство не надо. Но тогда... О, Аллах!

Царь в смятении — он узнал, что есть в мире вещи, недоступные его разуму и впервые в жизни неподвластные его силе.

Тут мы пока расстанемся с ним.

## Глава 2

### РЕШЕНИЕ ЗАДАЧ В ДИАЛОГОВОМ РЕЖИМЕ ЧЕЛОВЕК — ПМК

В ряде случаев ПМК может участвовать в решении вместе с человеком. Каждый из двух участвующих партнеров — человек и ПМК — выполняет наиболее доступную ему работу. На долю

человека обычно приходится творческая часть: ограничение области поиска решения — числа вариантов, обсчитываемых ПМК (на языке кибернетики — разрежение дерева целей), недоступный ПМК анализ информации при ее мгновенном зрительном восприятии человеком\*. На долю ПМК достаются рутинные вычислительные процедуры, требующие, однако, от человека больших усилий и времени.

Поясним сущность «союза» человек — ПМК на решении конкретных задач.

#### 2.1. ШЕСТИЗНАЧНОЕ ЧИСЛО

Найти шестизначное число с последовательно возрастающими цифрами, являющееся полным квадратом (см. «Наука и жизнь», 1984, № 1).

Проведем основной и контрольные эксперименты.

Основная функция человека — ограничение области поиска числа. Полный квадрат оканчивается цифрами 1, 4, 5, 6, 9. В нашем случае цифры 1, 4, 5 исключены: минимальное из возможных чисел 123 456. Но это число не квадрат:  $351^2 = 123\ 201$ ;  $352^2 = 123\ 904$ . Следовательно, последняя цифра искомого числа 9. Тогда третья цифра трехзначного основания квадрата 3 или 7. Далее, 317 — наименьшее трехзначное число, квадрат которого шестизначен  $317^2 = 100\ 489$  и оканчивается 9, поскольку  $313^2 = 97969$  — пятизначно. Число 673 — наибольшее трехзначное число, квадрат которого не превышает максимально допустимого по условию задачи числа 456 789:  $673^2 = 452\ 929$ , а  $677^2 = 458\ 329$ .

Итак, основание искомого квадрата есть трехзначное число, заключенное между числами 317 и 673 и оканчивающееся 3 или 7. Вместо 900 возможных значений трехзначных оснований искомого квадрата ( $10^2 \cdot 9 = 900$ ) мы получаем 70: в каждом из 35 десятков между 320 и 670 содержатся два числа, оканчивающиеся цифрами 3 и 7. Трудоемкость последующего выбора уменьшена более чем на порядок. Дерево целей сильно разрежено, доступно обработке ПМК в приемлемое время.

Как распределяются функции между ПМК и человеком в диалоге (см. программу 1)? ПМК формирует в ограниченном диапазоне «допустимых» оснований искомое число и вычисляет шестизначный квадрат этого основания. Человек визуально проверяет полученное ПМК число: «упорядочены ли цифры по возрастанию?» — это наиболее трудоемкая для ПМК и доступная человеку функция. Затем ПМК проверяет окончание и т. д.

Чтобы доказать эффективность союза человек — ПМК, было поставлено два контрольных эксперимента. В первом задачу полностью решал человек. Ему представлена возможность в «сво-

\* Замечательную способность человека к одновременному (параллельному, симультанному) «хватыванию» информации по многим параметрам не удается воспроизвести в ЭВМ и тем более на микрокалькуляторах.

одном поиске» пользоваться ПМК как автоматом для введения чисел в квадрат. В эксперименте участвовало 30 студентов четвертого курса физико-математического факультета КПИ, на решение был отпущен 1 час. Во втором контрольном эксперименте ПМК независимо решал задачу по программе без вмешательства человека (см. программу 2). Разумеется, во всех трех экспериментах (основном и контрольных) были уравнены начальные условия: сформулирована ограниченная область поиска — интервал от 317 до 673, указано правило формирования оснований квадратов (последовательное добавление чисел 6 и 4 к предыдущему основанию начиная с 317).

Сравним результаты. В основном эксперименте задача решается в диалоговом режиме. ПМК в разреженном человеком множестве альтернатив генерирует гипотезы — формирует основания, находит их квадраты, предоставляя человеку решить вопрос о возрастании цифр в полученному квадрате. Затем «слово» снова дается ПМК и т. д. По этой методике ответ 134 689 получен за 45 с. Для полного перебора всех вариантов, необходимого для того, чтобы убедиться в отсутствии других решений, потребовалось 7 мин.

Проанализируем результаты контрольных экспериментов.

В первом контрольном эксперименте с решением задачи из 30 человек справилось 24. Среднее время, затраченное участниками эксперимента, 30 мин. В течение часа ни один не осуществил полного перебора, поэтому вопрос о единственности решения остался открытым, хотя в постановке задачи это предусмотрено. Основные трудности и ошибки связаны с необходимостью запоминать и записывать большое количество промежуточных данных, а также с монотонностью процесса вычисления, что ведет к потере интереса и быстрой утомляемости.

Во втором контрольном эксперименте на поиск искомого числа 134 689 по программе 2 потребовалось 6 мин. Полный перебор выполнен за 45 мин. Нетрудно догадаться, что основные затраты времени приходятся на упорядочение по возрастанию цифр очередного квадрата, т. е. на процедуру, которая человеком фактически выполняется мгновенно.

Подытожим результаты затрат времени на решение задачи в каждом эксперименте:

Задача	Диалоговый режим	Человек	ПМК
Поиск числа 134 689	45 с	30 мин	6 мин
Полный перебор в очерченных границах	7 мин	Испытуемые не справились с задачей	45 мин

Следовательно, диалоговый режим работы при решении задач рассмотренного комбинаторно-переборного типа эффективнее независимого решения этих задач человеком или ПМК. Человек и машина — не взаимоисключающие компоненты системы, они образуют союз по дополнению.

Приведем программы, на основе которых строился эксперимент.

### Программа 1 (диалоговая)

#### Распределение памяти

- P0 — если содержимое P0 равно нулю, то для построения очередного основания квадрата к предыдущему основанию прибавляют 6; если это содержимое равно единице, прибавляют 4. В результате чередования нулей и единиц основание квадрата оканчивается поочередно 3 и 7.  
 +PA\* — Число 317, преобразуется в 323, 327 и т. д. — последовательно все «допустимые» основания, т. е. не превышающие 670.  
 +PB — константа  $673^2 = 452\ 929$  для проверки окончания перебора

00	CX	0Г	09	0	00	18	1	01
01	X→P0	40	10	C/P	50	19	X→P0	40
02	P→XA	6—	11	P→X0	60	20	БП	51
03	FX <sup>2</sup>	22	12	FX=	5E	21	02	02
04	C/P	50	13	22	22	22	P→XA	6—
05	P→XB	6L	14	P→XA	6—	23	4	04
06	—	11	15	6	06	24	+	10
07	FX $\geqslant$ 0	59	16	+	10	25	X→PA	4—
08	11	11	17	X→PA	4—	26	БП	51
						27	00	00

Инструкция. 317 X→PA, 452 929 X→PB В/О С/П (2 с) «100489». Индикация 0 означает, что задача решена. В остальных случаях на индикаторе квадраты чисел для проверки человеком возрастания цифр. Нажатием С/П запускается продолжение вычислений.

### Структура программы

- 02—04 : формирование числа, квадрат которого индицируется и проверяется человеком на возрастание цифр.  
 05—10 : проверка окончания — если задача решена, индицируется 0.  
 13—21 : если содержимое P0 равно нулю, то основание квадрата оканчивается 7.  
 22—25 : если содержимое P0 равно единице, то основание квадрата оканчивается на 3.

### Программа 2. Человек исключен из контура решения

#### Распределение памяти

- P0 — регулятор одного из двух способов построения основания квадрата (см. планирование к программе 1);

\* В регистры, отмеченные знаком «+», содержимое вводится вне программы. В остальных регистрах оно формируется ПМК.

- Р1 — счетчик для нахождения справа налево цифр предполагаемого квадрата, начальное значение — 6;  
 Р2 — регистр служит вычислению целой части числа;  
 +Р4 — константа 10;  
 Р6 — запоминается предыдущая цифра шестизначного квадрата для решения вопроса о монотонном возрастании цифр;  
 +РА — вначале 317, затем 323; 327 и т. д. (см. планирование к программе I);  
 +РВ — 452 929 — константа для проверки окончания перебора;  
 РС — квадрат очередного основания, хранящегося в РА.

00	СХ	0Г	21	+	10	42	-	11
01	X→Р0	40	22	X→РА	4—	43	Р→Х4	64
02	6	06	23	1	01	44	Х	12
03	X→П1	41	24	X→Р0	40	45	В↑	0Е
04	Р→ХА	6—	25	БП	51	46	Р→Х6	66
05	FX <sup>2</sup>	22	26	04	04	47	—	11
06	X→ПС	4C	27	Р→ХА	6—	48	FX<0	5C
07	БП	51	28	4	04	49	56	56
08	33	33	29	—	10	50	↔	14
09	Р→ХС	6C	30	X→РА	4—	51	X→П6	46
10	Р→ХВ	6L	31	БП	51	52	FL1	5L
11	—	11	32	00	00	53	60	60
12	FX≥0	59	33	X→П6	46	54	Р→ХС	6C
13	16	16	34	Р→Х4	64	55	C/П	50
14	0	00	35	:	13	56	6	06
15	C/П	50	36	1	01	57	X→П1	41
16	Р→Х0	60	37	+	10	58	БП	51
17	FX=0	5E	38	X→П2	42	59	09	09
18	27	27	39	KП→Х2	Г2	60	Р→Х2	62
19	Р→ХА	6—	40	FО	25	61	БП	51
20	6	06	41	Р→Х2	62	62	34	34

Инструкция. 10 X→Р4, 317 X→РА, 452 929 X→ПВ В/О С/П (6 милли)  
«134689».

Программа состоит из двух частей. В первой (команды 00—32) вычисляются основания квадратов, как в программе I, во второй (команды 33—78) ПМК проверяет квадрат на возрастание цифр.

В РС формируется полный шестизначный квадрат, от него, начиная с правого края, отделяется и анализируется по одной цифре. Если рассматриваемая цифра меньше ранее найденной, то она поступает на ее место в Р6; процесс поиска цифр справа налево продолжается (команды 33—41).

Если на некотором этапе условие команды 48 не выполнено, то счетчик в Р1 устанавливается на 6 (команды 49, 56—59). В зависимости от того, вышел ли квадрат за допустимую границу, индицируется нуль (нет решения) либо проверяется новая гипотеза об искомом числе. Если условие команды 48 на всех этапах выполнено, индицируется содержимое РС — задача решена.

## ЗАДАНИЯ

1. Задача КРЕСТ (упрощенный вариант игры «Брюссельская капуста», см. Гарднер М. Математические новеллы: Пер. с англ. Ю. А. Данилова. — М.: Мир, 1974. — 454 с.). Дан крест +. Сделать ход — значит соединить дугой без пересечений две его вершины и, проведя поперечную прямую к линии пересе-

чения, образовать новый крест. Каждая вершина креста может быть концом только одного соединения. Требуется выполнить все возможные соединения. Составить программу для решения задачи ПМК.

В сложившейся ситуации интересно, как в диалоге с человеком ПМК решает типично топологическую задачу. Вершины креста про- нумерованы: 1, 2, 3, 4. Независимо от порядка их соединения игра завершается в три хода.

Вершина, из которой исходит соединительная дуга, называется *связной*, остальные вершины — *свободными*. Свободная вершина, находящаяся внутри замкнутого контура (ее нельзя без пересечений соединить с другой вершиной) называется *изолированной* (рис. 8).

Машине в переборе находит две свободные неизолированные вершины, засыпает в соответствующие регистры нули, выясчивает номера выбранных вершин. Человек соединяет их на рисунке, проводит поперек к линии связи прямую, нумерует новые вершины, вводит в ПМК номер возникшей свободной вершины. ПМК запоминает введенный номер в одном из регистров, где записан 0, проверяет окончание. Если задача не решена, то процедура повторяется.

Каждый из двух компонентов системы решает свою задачу: человек рисует, считывает с чертежа, набирает на клавиатуре. ПМК считает, сравнивает, осуществляя перебор.

Чтобы облегчить составление программы, приведем алгоритм на УАЯ:

```

алг КРЕСТ (нат N[1:4], лит A)
  арг N
  рез A
  нач нат i, j, k, l, r, цел p
    для i от 1 до 3
    нц!
      p := 0
      для j от 1 до 2
      нц
        для k от p+1 до 4
        нц
          если N[k] ≠ 0
            то l := k; r := N[k]; N[k] := 0; p := k; k := 4; выв r
            все
          кц
        ввод s
        ком Работает человек: рисует, нумерует и вводит s
        N[l] := s
      кц
      A := «задача решена»
    кон
  
```

2. Обобщите задачу в игре на  $n$  крестах. ПМК и человек делают ходы поочередно. ПМК выигрывает, если ему дано право решать, кто делает первый ход ( $n = 5p - 2$ , при нечетном  $n$  ПМК начинает).

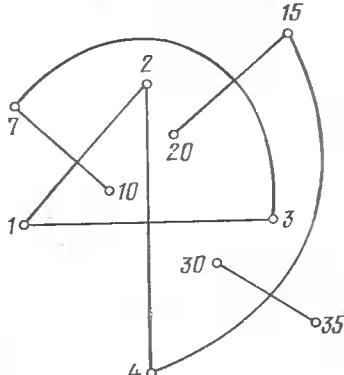


Рис. 8

## 2. ЧЕЛОВЕК И ПМК СОВМЕСТНО РАСШИФРОВЫВАЮТ ЧИСЛА

ТРАССА+ТРАССА=КОСМОС. Однаковым буквам соответствуют одинаковые цифры, разным буквам — разные цифры. Расшифруйте числа.

Чтобы не путать букву О с цифрой 0 (нуль), мы в этом параграфе, как это принято для ЭВМ, цифру нуль перечеркиваем — Ø.

При решении рассматриваемой задачи типично переборного типа важно выяснить ту грань уменьшения человеком множества вариантов (разрежения дерева целей), начиная с которой целесообразно передать доводку решения задачи ПМК. Наблюдая за решением задачи человеком, можно заметить, что снятие «верхнего слоя» неопределенности на первом этапе проходит сравнительно легко. Трудности начинаются, когда прямое решение уже не очевидно и необходимо применить метод допущений и отсечения непригодных вариантов. Но есть и другое продолжение, почти недоступное человеку, — полный перебор всех оставшихся вариантов. Как раз этот метод решения наиболее естествен для ПМК.

На первом этапе решения  $T \neq \emptyset$ , иначе слагаемые пятизначны;  $T \leq 4$ , в противном случае сумма семизначна, С четно, поскольку  $A+A$  должно оканчиваться четной цифрой.  $C = \emptyset$ : если  $C = \emptyset$ , то  $M = C + C = \emptyset$ , что невозможно.  $C \leq 4$ , иначе при сложении переходит единица из третьего разряда в четвертый, и цифра С в разряде тысяч слова КОСМОС оказывается нечетна:  $A+A+1$ . В итоге получаем  $C=4$  или  $C=2$ .  $A \geq 5$ , в противном случае нет переноса при сложении из разряда единиц в разряд десятков, и  $M=O=C+C$ . Отсюда следует, что в разряде десятков  $O=2C+1$ , а в разряде сотен  $M=2C$ . Сравнив два значения О, убеждаемся, что  $O=2C+1$  и, с другой стороны,  $O=2P+1-1\emptyset$ . Отсюда  $P=C+5$ , а тогда  $K=2T+1$ .

Мы подошли к той черте в анализе, когда дальнейшее решение целесообразно осуществить методом допущений. Прежде чем продолжить, соберем полученные результаты:  $T=1, 2, 3, 4$ ;  $A=5, 6, 7, 8, 9$ ;  $C=2, 4$ ;  $P=C+5$ ;  $O=2C+1$ ;  $M=2C$ ;  $K=2T+1$ .

Так как Р, О, М, К выражаются через Т и С, то начнем перебор с буквы С, имеющей минимум возможных значений.

1. Пусть  $C=4$ , тогда  $A=7$ . Действительно, число  $2A$  оканчивается четверкой только при  $A=2$  или  $A=7$ . Но  $A=2$  исключено:  $A>4$ . Из  $P=C+5$  следует  $P=9$ ; из  $O=2C+1$   $O=9$ ; из  $M=2C$   $M=8$ . Получилось  $P=O=9$  — допущение  $C=4$  привело к противоречию. Остается надежда на  $C=2$ . Если и это продолжение завершится противоречием, то задача не имеет решения.

2. Пусть  $C=2$ , тогда  $A=6$ . В самом деле, для  $A \geq 5$  это единственный случай, когда число  $2A$  оканчивается двойкой. Из  $P=C+5$  вытекает  $P=7$ ; из  $O=2C+1-O=5$ ; из  $M=2C-M=4$ . Остается вычислить значения Т и К.  $K=2T+1$ , следовательно,  $K \geq 3$ . Так как К, кроме того, нечетно, то возможны  $K=3, 5, 7, 9$ .

Но  $K \neq 5, 7$  ввиду  $O=5$  и  $P=7$ , а также  $K \neq 9$ , иначе  $T=M=4$ . Остается  $K=3$ , и тогда  $T=1$ . Противоречия как будто не возникло. Запишем все результаты и проверим:  $C=2, M=4, A=6, O=5, P=7, K=3, T=1; 176226 \cdot 2 = 352452$ .

Вернувшись к концу первого этапа анализа, попытаемся продолжить решение в полном переборе оставшихся вариантов. Так как А, Т, С выбираются соответственно из пяти, четырех и двух допустимых значений, то всего предстоит рассмотреть  $4\emptyset$  вариантов. Для упрощения программы мы уменьшили число ограничений, исключив условие разные буквы — разные цифры. ПМК, естественно, будет выдавать «расширенный репертуар» ответов, среди них содержится и искомый. Отсеивание «лишних решений» производится человеком визуально, легко, тогда как для ПМК требуется значительное усложнение программы. Напротив, полный перебор оставшихся вариантов для человека затруднителен, а решение методом допущений требует больших умственных усилий. В диалоге человек — ПМК каждый, как видим, делает «свое дело».

Если для ПМК естествен полный перебор оставшихся вариантов, то для ЭВМ обычно выбирают просто полный перебор вариантов, что часто приводит к неоправданным затратам времени. Так, полным перебором вариантов задача решается микроЭВМ за 85 ч, полный перебор оставшихся вариантов, где А, Т, С выбираются из пяти, четырех и двух допустимых значений, занимает на ПМК всего 15 мин.

### Программа ТРАССА+ТРАССА=КОСМОС

#### Распределение памяти

P0 — счетчик Т на 4 повторения цикла;  
P1 — счетчик А на 5 повторений;  
+P2 — счетчик С на 2 повторения;  
+P3 — 10 — константа для формирования числа из отдельных цифр;  
P4 — переменное значение Т, вначале 1;  
P5 — переменное значение А, вначале 5;  
P6 — переменное значение С, вначале 2;  
+P9, PA — параметры для косвенного обращения к подпрограммам: 68, 65;  
PB — числовой код слова ТРАССА;  
PC — вычисленное в данном цикле переменное значение буквы О;  
PD — числовой код слова КОСМОС.

Инструкция. 2 X→P2, 10 X→P3, 65 X→PA, 68 X→P9 B/O С/П (3 мин) «176226». Наберите программу и вернитесь в режим вычислений. Чтобы запустить программу, нажмите B/O С/П. Теперь придется подождать 3 мин, пока на индикаторе не появится код слова ТРАССА. Нажмите С/П — на индикаторе код слова КОСМОС. Еще раз нажмите С/П, и ПМК вновь примется за поиск решения. Я на индикаторе означает, что новых решений не появится, задача решена. ПМК не исключает, что разные буквы могут принимать одинаковые численные значения, среди выдаваемых «корней» могут оказаться «посторонние». Условие разные буквы — разные цифры поможет вам отсеять непригодные ответы.

Контрольный пример. После набора и редактирования программы: F ABT B/O С/П (3 мин) «176226» — слово ТРАССА, С/П (2 с) «352452» — слово КОСМОС. Нетрудно убедиться, что это решение задачи, все условия выполнены.

00	СХ	0Г	24	КПП19	-9	48	:	13
01	X→П6	46	25	П→Х5	65	49	П→ХВ	6L
02	КП→Х6	Г6	26	+	10	50	—	11
03	КП→Х6	Г6	27	X→ПВ	4L	51	FX=0	5E
04	5	05	28	П→Х4	64	52	57	57
05	X→П1	41	29	2	02	53	П→ХВ	6L
06	4	04	30	×	12	54	C/П	50
07	X→П5	45	31	1	01	55	П→ХД	6Г
08	КП→Х5	Г5	32	КПП1A	—	56	C/П	50
09	4	04	33	2	02	57	FL0	5Г
10	X→П0.	40	34	×	12	58	13	13
11	СХ	0Г	35	1	01	59	FL1	5L
12	X→П4	44	36	+	10	60	08	08
13	КП→Х4	Г4	37	X→ПС	4C	61	FL2	58
14	П→Х4	64	38	КПП1A	—	62	02	02
15	П→Х3	63	39	КПП1A	—	63	F π	20
16	×	12	40	2	02	64	C/П	50
17	П→Х6	66	41	×	12	65	КПП19	-9
18	5	05	42	КПП19	-9	66	П→Х6	66
19	+	10	43	П→ХС	6C	67	B/O	52
20	КПП19	-9	44	КПП1A	—	68	+-	10
21	П→Х5	65	45	+	10	69	П→Х3	63
22	КПП1A	—	46	X→ПД	4Г	70	×	12
23	КПП1A	—	47	2	02	71	B/O	52

Может быть, что найденное решение не единственное. Продолжим перебор. С/П «276226» С/П «552452». Не выполнено условие: разные буквы — различные цифры. У нас получилось: буквы Т и С равны по 2, буквы К и О — по 5. С/П «376226» С/П «752452» К=Р — решение постороннее, С/П «476226» С/П «952452». Т=М — решение постороннее и т. д., пока на индикаторе не появится π — сигнал о завершении перебора.

Полученное ранее решение окажется единственным. Полный перебор занимает 15 мин. МикроЭВМ для полного перебора без ограничений, введенных человеком, требуется около 100 ч. Такова роль человеческого фактора в диалоге человек — ЭВМ.

## Структура программы

Программа состоит из трех вложенных друг в друга циклов. Внутренний цикл (счетчик Р0) формирует числовой код буквы Т на четыре повторения, начиная с Т=1, с шагом 1. Следующий цикл (счетчик Р1) формирует числовой код буквы А на пять повторений, начиная с А=5, с шагом 1. Наконец, внешний цикл (счетчик Р2) формирует значения буквы С на два повторения, от С=2 с шагом 2 (рис. 9). Структура особенно четко прочитывается в алгоритме на УАЯ.

Буквы Т, А, С в программе принимают значения независимо. Остальные — Р, О, М, К — вычисляются по известным формулам через них. Эти буквы, за исключением О, используемой два раза в слове КОСМОС, не запоминаются, а сразу применяются для построения слов ТРАССА и КОСМОС. Когда слова построены, проверяется равенство КОС-

Рис. 9

МОС/2=ТРАССА. Если оно выполнено, то соответствующие числа индицируются, затем ПМК продолжает поиск других решений. Сигналом π машина информирует, что решение закончено.

- 00—13 : занесение начальных данных. Подготовка счетчиков циклов и текущих значений переменных в циклах.
  - 14—20 : вычисление значения  $(T \cdot 1\varnothing + P) \cdot 1\varnothing$  для найденных значений букв.
  - 21—27 : построение в РВ кода слова ТРАССА по формуле  $((...)).1\varnothing + A$ .
  - 28—37 : содержимое РС равно О. Эта буква имеет два вхождения в слове КОСМОС; мы сохраняем ее в РС, чтобы второй раз не формировать.
  - 38—46 : построение числового кода слова КОСМОС по рекурсивной формуле  $((((K \cdot 1\varnothing + O) \cdot 1\varnothing + C) \cdot 1\varnothing + M) \cdot 1\varnothing + O) \cdot 1\varnothing + C$  в РД.
  - 47—51 : проверка равенства КОСМОС/2=ТРАССА.
  - 53—56 : равенство выполнено, индицируется числовой код слова ТРАССА и после нажатия С/П — слова КОСМОС.
  - 57—64 : проверка содержимого счетчиков и обращение к началу цикла вычислений, если перебор оставшихся вариантов еще не завершен. Индикация π свидетельствует об окончании решения.
  - 65—67 : подпрограмма, обращение к которой производится в соответствии с содержимым РА.
  - 68—71 : подпрограмма, обращение к которой производится в соответствии с содержимым Р9.
- ```

алг БУКВЫ—ЦИФРЫ
нач цел С, А, Т, Р, О, М, К, μ, X, Y, nat i, j, цел таб R[1:7]
    С := 2
    пока С ≤ 4
    нц
        Р := С+5; О := 2*С+1; М := 2*С; R[1]:=С; R[4]:=Р; R[5]:=О; R[6]:=М
        Т := 1
        пока Т ≤ 4
        нц
            К := 2*Т+1; R[3]:=Т; R[7]:=К
            А := 5
            пока А ≤ 9
            нц
                R[2]:=А; i := 1; μ := 1
                пока i ≤ 6 ∧ μ = 1
                нц
                    j := i + 1
                    пока j ≤ 7 ∧ μ = 1
                    нц
                        если R[i] = R[j]
                            то μ := Ø
                            иначе j := j + 1
                        все
                    кц
                    i := i + 1
                кц
            кц
        кц
    кц

```

```

кц
если μ = 1
    то X := 1Ø⁵*T + 1Ø⁴*P + 1Ø³*A + 1Ø²*C + 1Ø*C + A
    Y := 1Ø⁵*K + 1Ø⁴*O + 1Ø³*C + 1Ø²*M + 1Ø*O + C
    ком не смешивайте букву Ø как множитель с нулем
если Y = 2*X
    то вывод X, Y
    ком выводятся значения X и Y
все
все
A := A + 1
кц
    T := T + 1
    C := C + 2

```

## ЗАДАНИЯ

- 1. + ШИФЕР**      Указание.  $\overline{\text{ШИФЕР}}$       **КРЫША**

Ш $\neq$ 0, Ш $<$ 4. Пусть Ш четно, тогда Р $<$ 5, Е=7; 6 или Е=1; 2. Рассмотрите варианты. Затем аналогично рассмотрите случай: Ш нечетно — Ш=1; 3

Задача имеет несколько решений:

$$\begin{array}{r}
 + 41523 & 18456 & 14358 \\
 + 41523 & + 18456 & + 14358 \\
 \hline
 83046 & 36912 & 28716
 \end{array}$$

Найдите все варианты решения.

2. НАЛИМ $\times$ 4-ЛИМАН. Указание. Н, Л $\neq$ 0: числа пятизначные. Н $\leqslant$ 2, в противном случае произведение шестизначно, но Н четно, так как четно число ЛИМАН, следовательно, Н=2. Тогда Л=4 $\times$ Н=8 или Л=4 $\times$ Н+1=9. Далее, А $<$ 5, иначе 4 $\times$ А $\geq$ 20, и, учитывая, что Н=2, получим Л $\geqslant$ 4 $\times$ Н+2=10. Но ЛИМАН кратно 4, следовательно, кратно 4 и число АН—признак кратности 4. В итоге А=1 или А=3. Перебором убеждаемся, что другие значения невозможны. 30 $\leqslant$ 4 $\times$ Л $\leqslant$ 40 независимо от Л=8 или Л=9; тогда 4 $\times$ А+3 заканчивается на И, следовательно, И=5 или И=7. Наконец, 4 $\times$ М заканчивается буквой Н, принимающей значение 2, откуда М=3 или М=8.

Еще одно ограничение. Если  $A=1$ , то вследствие отсутствия переноса цифры в старший разряд произведения  $L=4 \times H=8$ , и тогда  $M=3$ . Если  $A=3$ , то  $M=8$ ,  $L=9$ .

Собрав вместе полученные результаты, приходим к следующим четырем вариантам слова НАЛИМ: 21853 21873 23958 23975, подлежащим проверке условиям НАЛИМ $\times$ 4=ЛИМАН. Поручим эту работу машине. Составьте соответствующую программу.

Ответ: НАЛИМ=23958.

3. Расшифруйте с помощью программы: АГАТ=(АГ+АТ)<sup>2</sup>. Указание. А<5, в противном случае АГ+АТ>100 и  $(\text{АГ}+\text{АТ})^2=\text{АГАТ}$  пятизначно. А≠4, так как если А=4, то АГ+АТ>80,  $(\text{АГ}+\text{АТ})^2>6400$ . Следовательно, слово АГАТ не начинается с 4. Слово АГАТ как полный квадрат заканчивается одной из цифр: 1, 4, 5, 6, 9. Из этого множества выбирают Т. Заметим, что Т≠0 в противном случае для полного квадрата А=0.

Подставьте допустимые значения  $A$  и  $T$  в данное равенство. Убедитесь, что  $G$  принимает одно из значений 0, 4, 8. С их помощью исключите как невозможные  $T=1$  и  $T=-9$ .

Мы получили следующие результаты:  $A=1, 2, 3$ ;  $T=4, 5, 6$ ;  $\Gamma=0, 4, 8$ . Возможных значений троек  $A, \Gamma, T$  — 27 вместо 900, как это было при неограниченном переборе.

В программе для каждого допустимого набора значений А, Г, Т находится число  $A\bar{G}AT = A \cdot 10^3 + G \cdot 10^2 + A \cdot 10 + T$ . Далее вычисляются  $(A\bar{G} + AT)^2 = (A \cdot 10 + G + A \cdot 10 + T)^2$ . Перебор продолжается до тех пор, пока не будут проанализированы все 27 вариантов.

Заметим, что при более глубоком разрежении дерева целей можно доказать, что  $T \neq 4$ . Преобразуйте свою программу с учетом этого ограничения.

4. ТОКИО : ИО = КИО. Ответ: ТОКИО = 15625.  
 5.  ПЯТЬ Введите ограничения и составьте программу для ПМК, решющую задачу. Ответ: ПЯТЬ = 2486.

6. Задача-шутка. Вычислите отношение ТХБХИХЛХИХСХИ/ГХРХУХ+ЗХИХЯ?

7. В шестизначном числе  $*1022*$  восстановить недостающие цифры, если число делится на 7, 8, 9. В полном переборе крайних цифр числа имеются 90 вариантов: первая цифра принимает значения от 1 до 9, последняя — от 0 до 9. Исскомое число делится на 9, следовательно, сумма его цифр делится на 9. Тогда сумма крайних цифр либо 4, либо 13. Давая в переборе значения первой цифре (их 9), последнюю получаем вычитанием этих значений из 4, если они не больше 4, и из 13 — в противном случае. Число вариантов перебора сведено к 9.

### Решение задачи 7

## Распределение памяти

- P6 — первая искомая цифра, ее начальное значение 1, последующие значения образуются прибавлением 1 к предыдущим;  
P7 — вторая цифра;  
PB — сформированное число;  
PA — 1022.

|    |       |    |    |      |    |    |       |    |
|----|-------|----|----|------|----|----|-------|----|
| 00 | X→ΠΑ  | 4— | 13 | X→Π7 | 47 | 26 | .     | 0— |
| 01 | 0     | 00 | 14 | Π→XA | 6— | 27 | 8     | 08 |
| 02 | X→Π6  | 46 | 15 | Π→X6 | 66 | 28 | :     | 13 |
| 03 | 9     | 09 | 16 | ΒΠ   | 0C | 29 | F sin | 1C |
| 04 | X→Π0  | 40 | 17 | 4    | 04 | 30 | FX=0  | 5E |
| 05 | KΠ→X6 | Γ6 | 18 | +    | 10 | 31 | 34    | 34 |
| 06 | 4     | 04 | 19 | 1    | 01 | 32 | Π→XB  | 6L |
| 07 | Π→X6  | 66 | 20 | 0    | 00 | 33 | C/Π   | 50 |
| 08 | —     | 11 | 21 | ×    | 12 | 34 | FL0   | 5Γ |
| 09 | FX<0  | 5C | 22 | Π→X7 | 67 | 35 | 05    | 05 |
| 10 | 13    | 13 | 23 | +    | 10 | 36 | F π   | 20 |
| 11 | 9     | 09 | 24 | X→ΠB | 4L | 37 | C/Π   | 50 |
| 12 | +     | 10 | 25 | 2    | 02 |    |       |    |

Инструкция.. 1022 В/О С/П (1 мин 28 с) «910224» С/П (1 с) «π»  
Переключатель в положении Г

## Структура программы

- 00—04 : ввод и подготовка данных для организации цикла;  
05 : формирование первой цифры в Р6;

06—13 : вычисление второй цифры;

14—24 : построение искомого числа в РВ;

25—30 : проверка кратности числа 504 (7, 9 и 8);

31—37 : индикация л, если число удовлетворяет условию, и возврат к началу цикла генерирования нового числа в противном случае.

Замечание. Согласно программе, если первая цифра больше 4, то сумма крайних цифр числа 13 (команды 06—09, 11—13), если меньше или равна 4, то эта сумма четыре, команды 06—09, 10, 13.

В действительности, когда первая цифра 4, существуют две возможности для последней цифры: 0 и 9. В связи с этим пришлось независимо от программы убедиться, что число 410229 не удовлетворяет условию.

Предусмотрите в программе полный анализ случая равенства четырем первой цифры искомого числа.

Нетрудно заметить, что число альтернатив сведется к пяти, если, учитывая четность последней цифры (число ведь делится на 8), давать ей значения 0, 2, 4, 6, 8, а вычислять первую цифру. Преобразуйте программу.

## 2.3. ЭКСПЕРИМЕНТЫ С ДВУЗНАЧНЫМИ ЧИСЛАМИ

Задачи, приведенные в этом параграфе, относятся к переборному типу. Предварительное ограничение множества альтернатив позволяет запрограммировать решение для ПМК. Ограничение состоит в том, что рассматриваются не все двузначные числа, а только те, у которых вторая цифра не меньше первой. Это позволяет вдвое сократить количество анализируемых чисел, значительно уменьшить время решения.

Найти все двузначные числа, сумма квадратов цифр которых делится на 13.

Задача решается с помощью двух циклов: внешнего — по множеству значений первой цифры (от 1 до 9) и внутреннего — по значениям второй цифры, не меньшим очередного значения первой цифры. Например, если первая цифра 6, то программа генерирует для анализа числа 66, 67, 68, 69.

### Программа СУММА КВАДРАТОВ КРАТНА 13

Распределение памяти

P4 и P5 — очередные первая и вторая цифры искомого числа;  
P1 и P2 — счетчики первой и второй цифр.

|    |       |    |    |                 |    |    |       |    |
|----|-------|----|----|-----------------|----|----|-------|----|
| 00 | CX    | 0Г | 15 | FL1             | 5L | 29 | 0     | 00 |
| 01 | 9     | 09 | 16 | 05              | 05 | 30 | ×     | 12 |
| 02 | X→P1  | 41 | 17 | F π             | 20 | 31 | F sin | 1C |
| 03 | 0     | 00 | 18 | C/P             | 50 | 32 | FX=0  | 5E |
| 04 | X→P4  | 44 | 19 | Π→X4            | 64 | 33 | 40    | 40 |
| 05 | KΠ→X4 | 44 | 20 | FX <sup>2</sup> | 22 | 34 | Π→X4  | 64 |
| 06 | Π→X1  | 61 | 21 | Π→X5            | 65 | 35 | BΠ    | 0C |
| 07 | X→P2  | 42 | 22 | FX <sup>2</sup> | 22 | 36 | 1     | 01 |
| 08 | Π→X4  | 64 | 23 | +               | 10 | 37 | Π→X5  | 65 |
| 09 | 1     | 01 | 24 | 1               | 01 | 38 | +     | 10 |
| 10 | —     | 11 | 25 | 3               | 03 | 39 | C/P   | 50 |
| 11 | X→P5  | 45 | 26 | :               | 13 | 40 | FL2   | 58 |
| 12 | KΠ→X5 | 45 | 27 | 1               | 01 | 41 | 12    | 12 |
| 13 | PΠ    | 53 | 28 | 8               | 08 | 42 | B/O   | 52 |
| 14 | 19    | 19 |    |                 |    |    |       |    |

Инструкция. В/О С/П. Переключатель в положении Г. После каждого останова зафиксируйте индицируемое число и, нажав С/П, продолжите вычисления. Индикация л означает, что перебор завершен, других чисел, удовлетворяющих условию, нет.

Решение задачи. В/О С/П (37 с) «15» С/П (20 с) «18» С/П (23 с) «23» С/П (1 мин 55 с) «46» С/П (7 с) «47» С/П и т. д. Процесс вычислений занимает меньше 3 мин.

Рассмотрим структуру основной части программы:

- 00—04 : подготовка данных для организации внешнего цикла;  
05—12 : подготовка данных для внутреннего цикла;  
13, 14 : обращение к подпрограмме;  
15—18 : проверка окончания внешнего цикла, по выходе из него индицируется число л, означающее, что задача решена;  
19—42 : подпрограмма проверки выполнения условия задачи для очередного двузначного числа.

Программа составлена с расчетом, чтобы, меняя условия задачи, нужно было изменять только подпрограмму.

Адрес алг СУММА КВАДРАТОВ  
команды нач nat i, j, A, B, цел R

```
00—04    для i от 1 до 9
          нц
05—12    для j от i до 9
          нц
          A := 10*i + j;  B := i**2 + j**2;  R := B - [B/13]*13
          если R = 0
          то выв A
          все
          кц
          кон
```

## ЗАДАНИЯ

1. Найти все двузначные числа, которые равны сумме квадратов своих цифр. Ответ: таких чисел нет.

2. Найти все двузначные числа, для которых:

1) Сумма кубов цифр делится на 13. Ответ: 14, 27, 28, 34, 49, 57, 58, 67, 68.

2) Сумма четвертых степеней делится на 13. Ответ: таких чисел нет.

3) Сумма четвертых степеней делится на 41. Ответ: 13, 26, 39.

4) Сумма пятых степеней кратна 11. Ответ: 12, 16, 17, 18, 23, 24, 25, 29, 36, 37, 38, 46, 47, 48, 56, 57, 58, 69, 79, 89.

3. Найти все двузначные числа, которые на 33 меньше суммы первой цифры и квадрата второй. Ответ: 17, других чисел нет.

4. Задумайте целое число от 1 до 15 включительно и назовите, в какую из четырех групп оно попало:

| Группа | 1          | 2             | 3          | 4             |
|--------|------------|---------------|------------|---------------|
| Числа  | 0, 1, 2, 3 | 0, 1, 2, 3, 8 | 0, 1, 4, 5 | 0, 2, 4, 6, 8 |

ПМК отгадывает задуманное. Составьте программу для решения задачи.

## Алгоритм. Пусть

$x_i = \begin{cases} 1, & \text{если задуманное число входит в } i\text{-ю группу,} \\ 0 & \text{в противном случае } (i=1, 2, 3, 4). \end{cases}$

Вы «сообщаете» ПМК число  $A=x_1x_2x_3x_4$ , цифры которого — единицы и нули.

Задуманное число равно разности между 15 и десятичным значением двоичного числа  $A$ :  $15 - (x_1 \cdot 2^3 + x_2 \cdot 2^2 + x_3 \cdot 2 + x_4)$ . Например ваше сообщение  $A = 1010$ , т. е. задуманное число входит в первую и третью группы и не входит в остальные.

Вы задумали:  $15 - (1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2 + 0) = 5$ .

Нетрудно убедиться, что 5 — единственное число, которое удовлетворяет сообщению 1010.

## ОТВЕТЫ

### Подпрограмма для задания 3

|    |      |    |    |                 |    |    |        |    |
|----|------|----|----|-----------------|----|----|--------|----|
| 19 | ПП   | 53 | 28 | X→P4            | 44 | 37 | ВП     | 0C |
| 20 | 26   | 26 | 29 | ↔               | 14 | 38 | 1      | 01 |
| 21 | ПП   | 53 | 30 | X→P5            | 45 | 39 | P→X5   | 65 |
| 22 | 26   | 26 | 31 | FX <sup>2</sup> | 22 | 40 | +      | 10 |
| 23 | FL2  | 58 | 32 | +               | 10 | 41 | —      | 11 |
| 24 | 12   | 12 | 33 | 3               | 03 | 42 | FX = 0 | 5E |
| 25 | B/O  | 52 | 34 | 3               | 03 | 43 | 25     | 25 |
| 26 | P→X4 | 64 | 35 | —               | 11 | 44 | BX     | 0  |
| 27 | P→X5 | 65 | 36 | P→X4            | 64 | 45 | C/P    | 50 |

### Программа для задачи 4

|    |      |    |    |      |       |    |     |     |
|----|------|----|----|------|-------|----|-----|-----|
| 00 | X→ПД | 4Г | 08 | X→П0 | 40    | 16 | +   | 10  |
| 10 | F0   | 25 | 09 | 9    | 09    | 17 | FL0 | 5Г  |
| 02 | X→ПС | 4С | 10 | X→П4 | 44    | 18 | 13  | 13  |
| 03 | F0   | 25 | 11 | 0    | 00    | 19 | 1   | 01  |
| 04 | X→ПВ | 4L | 12 | B↑   | 0E    | 20 | 5   | 05  |
| 05 | ↔    | 14 | 13 | 2    | 02    | 21 | ↔   | 14  |
| 06 | X→ПА | 4— | 14 | ×    | 12    | 22 | —   | 11  |
| 07 |      | 4  | 04 | 15   | KП→X4 | G4 | 23  | C/P |
|    |      |    |    |      |       |    |     | 50  |

Инструкция.  $x_1$  B↑  $x_2$  B↑  $x_3$  B↑  $x_4$  B/O C/P.

Контрольные примеры. 1 B↑ 0 B↑ 1 B↑ 0 B/O C/P (12 с) «5», 0 B↑ B↑ 1 B↑ 0 B/O C/P (12 с) «13».

## 2.4. ПМК ОТГАДЫВАЕТ ЦИФРЫ

Человек задумывает четырехзначное число. Число может начинаться с нуля, в нем могут быть и одинаковые цифры. Микрокалькулятор пытается его угадать — высвечивает свое четырехзначное число, а человек в ответ вводит информацию о количестве цифр задуманного числа, содержащихся в числе на индикаторе ПМК. ПМК анализирует информацию, называет следующее четырехзначное число и т. д. От ПМК требуется отгадать задуманное число (точнее, его цифры) с помощью меньшего числа ходов.

В отличие от известных игр здесь не существенно совпадение

позиций цифр в задуманном и предлагаемом числах. Главное — вхождение задуманных цифр в состав числа, генерируемого ПМК. Образно выражаясь, существен «материал», общий цифровой состав, а не позиция, занимаемая цифрой в числе. Это значительно упрощает задачу.

Условимся об обозначениях:  $m$  — четырехзначное число, называемое ПМК;  $l$  — количество цифр задуманного числа, содержащихся в  $m$ .

Предлагаем следующий алгоритм игры:

1. ПМК индицирует нуль:  $m=0000$ .
2. Человек сообщает количество  $l$  цифр задуманного числа, содержащихся в  $m$ .
3. Если  $l=4$ , то задача решена, в противном случае — следующее указание.

4. Микрокалькулятор оставляет первые  $l$  цифр в составе пробного числа неизменными, а все остальные — свободные цифры увеличивает на 1. Получается очередное значение  $m$ .

5. Индикация  $m$ . Возврат к указанию 2.

Каждое значение  $m$  является ситуацией в игре. Задуманное человеком число — целевая ситуация. Тогда количество несовпадающих цифр в текущей и целевой ситуациях можно интерпретировать как метрику в пространстве Хемминга (см. ч. II, гл. 1). В основе алгоритма лежит последовательное приближение к цели в этом пространстве. По структуре решения задача близка к задачам «Бочонок», «Игра на дорожке», рассмотренных ранее. Есть, однако, принципиальное отличие. Теперь целевая ситуация не известна решающему задачу, приближение к ней происходит не прямым сравнением ПМК двух ситуаций, а по мере получения информации извне. Первые  $l$  цифр, которые ПМК сохраняет в числе, — это инвариант, обеспечивающий приближение к «невидимой» цели.

Пример решения задачи по алгоритму.

| Номер хода | 1    | 2    | 3    | 4    | 5    | 6    |
|------------|------|------|------|------|------|------|
| ПМК        | 0000 | 1111 | 1222 | 1233 | 1234 | 1235 |
| Человек    | 0    | 1    | 2    | 3    | 3    | 4    |

Задумано 1532. Цифровой состав у полученного и задуманного числа одинаков.

Нетрудно заметить, что число ходов, необходимое для решения задачи, на единицу больше наибольшей цифры в задуманном числе: у нас она равна пяти, ходов шесть. Максимальное число ходов, необходимых для решения задачи, 10.

# Программа ПМК ОТГАДЫВАЕТ ЦИФРЫ

Распределение памяти

- P0 — счетчик, управляющий циклами распределения свободных цифр в регистры памяти и формирования очередного значения  $m$ ;  
 P1 — P4 — цифры, входящие в состав  $m$ ;  
 P5 — значение, принимаемое свободными цифрами в данном ходе; вначале нуль, в каждом следующем ходе увеличивается на 1;  
 P6 — для построения  $m$ .

|    |      |    |    |       |    |    |       |    |
|----|------|----|----|-------|----|----|-------|----|
| 00 | CX   | 0Г | 10 | KХ→P6 | L6 | 19 | 0     | 00 |
| 01 | X→P5 | 45 | 11 | FL0   | 5Г | 20 | ×     | 12 |
| 02 | X→P6 | 46 | 12 | 10    | 10 | 21 | KП→Х6 | Г6 |
| 03 | 4    | 04 | 13 | KП→Х5 | Г5 | 22 | +     | 10 |
| 04 | ↔    | 14 | 14 | 4     | 04 | 23 | FL0   | 5Г |
| 05 | —    | 11 | 15 | X→P0  | 40 | 24 | 18    | 18 |
| 06 | X→P0 | 40 | 16 | CX    | 0Г | 25 | C/P   | 50 |
| 07 | FX≠0 | 57 | 17 | X→P6  | 46 | 26 | БП    | 51 |
| 08 | 25   | 25 | 18 | 1     | 01 | 27 | 02    | 02 |
| 09 | P→Х5 | 65 |    |       |    |    |       |    |

Инструкция. В/О С/П (15 с) «0».

Контрольный пример. Задумано 1532. В/О С/П (15 с) «0», С/П (15 с) «1111», 1 С/П (15 с) «1222», 2 С/П (15 с) «1233», 3 С/П «1234», 3 С/П «1235», 4 С/П (2 с) «0» — задача решена.

## Структура программы

- 00, 01 : формирование начального значения  $m=0$ , оно сохраняется в Р5.  
 02—06 : вычисляется и запоминается в Р0 количество свободных цифр; начальное значение 4, конечное — 0.  
 08, 25 : индикация 0 свидетельствует об отсутствии свободных цифр — задача решена.  
 07, 09—12 : цикл формирования содержимого регистров для хранения свободных цифр.  
 13 : увеличение значения, принимаемого свободной цифрой на единицу.  
 14, 15 : формирование счетчика вычисления очередного значения  $m$ .  
 16—25 : нахождение и индикация  $m$ .  
 26, 27 : возврат к команде 02 для организации нового цикла вычислений.

## ЗАДАНИЯ

1. Программу «ПМК отгадывает цифры» можно сделать пригодной для трехзначных и двузначных чисел. Внесите в нее изменения, чтобы она решала эти задачи. Количество цифр, определяющих значение числа, используется только в командах 03 и 14. Подумайте, как преобразовать программу, чтобы она могла быть использована для отгадывания также пятизначных, шестизначных чисел.

2. В задаче «ПМК отгадывает цифры» решение строится на основе перебора в ограниченном программой множестве вариантов. Однако для решения ряда задач существуют более эффективные алгоритмы поиска неизвестного чи-

сла в диалоге человек — ПМК. Читатель, наверно, помнит детскую игру «Горячо-холодно». Играющему завязывают глаза, и онощупью должен найти товарища, который притаился в комнате и все время остается на месте. Поиском управляет ведущий с помощью двух сигналов: «горячо», «холодно». О приближении к объекту поиска ведущий сигнализирует восклицанием «горячо!», удалении — «холодно!» Будем считать, что «прятается» число, а ищет его ПМК — «заявлять» машине «глаза», конечно, нет необходимости. Если человек управляет поиском с помощью цифр 0 и 1, то приходим к задаче «Отгадывание числа». Дадим ей четкую формулировку.

Человек задумывает целое число от  $a=0$  до  $b=99999999$  — в пределах диапазона разрядной сетки индикатора. Калькулятор «отгадывает» задуманное, используя числа-гипотезы. Если предлагаемое ПМК число совпадает с задуманным, то задача решена, игра окончена. Если число больше задуманного, то человек вводит в машину 0, в противном случае — 1. От ПМК требуется решить задачу наименьшим числом ходов. Составьте программу, воспользовавшись алгоритмом:

1. Интервал  $(a; b)$  делится пополам в точке  $c = (a+b)/2$ ; значение  $c$  индицируется в качестве очередной гипотезы.

2. Если человек ввел в машину нуль (0 С/П), то новым правым концом интервала является  $c(P\rightarrow XC X\rightarrow PB)$ ; левый конец сохраняется. В противном случае  $c$  является левым концом интервала  $(P\rightarrow XC X\rightarrow PA)$ ; сохраняется правый конец.

3. Возврат к указанию 1.

Микрокалькулятор как бы берет цель в вилку, и с каждым ходом уменьшается расстояние до нее. Задача внешне и по существу близка к решению уравнения методом половинного деления (см. ч. 1). Есть и различие: при решении уравнения ПМК самостоятельно вычисляет «нужную половину», в нашем же случае, если продолжить аналогию, на нее указывает человек.

Минимальное число ходов, с помощью которых может быть решена любая задача рассматриваемого типа, равно  $\lceil \log_2 10^8 \rceil + 1 = 27$ . Квадратные скобки означают отделение целой части числа, стоящего в скобках, например,  $[8, 5] = 8$ .

Запишем алгоритм на УАЯ:

```

алг ОТГАДЫВАНИЕ ЧИСЛА (цел, a, b, c, n)
    арг a, b
    рез c, n
    нач цел m, лит L
        n := 0; L := «задуманное число меньше предложенного»
        пока L ≠ «задача решена»
            нц
                m := (a + b)/2; n := n + 1; вывод m; ввод L
                выбор
                    при L := «задуманное число меньше предложенного» b := m
                    при L := «задуманное число больше предложенного» a := m
                все
            кц
            c := m
    кон

```

Причесание.  $m$  — число-гипотеза, предлагаемое ПМК,  $L$  — высказывание человека о соотношении задуманного и предложенного чисел, символьная переменная, принимающая одно из трех возможных значений: «задуманное число меньше предложенного», «задуманное число больше предложенного», «задача решена»;  $n$  — счетчик числа гипотез, потребовавшихся для решения задачи.

3. В задаче «Отгадывание числа-2» поменяем партнеров местами: ПМК «задумывает», человек отгадывает. Постройте программу для этого случая. (Теперь ПМК управляет: вспомните «горячо!», «холодно!». Жаль, что нет у него эмоций!)

## 2.5. БЫКИ И КОРОВЫ

В отличие от игры «ПМК отгадывает цифры» (см. § 2.4) в игре «Быки и коровы» важно совпадение в двух числах не только цифр, но и их позиций. Игра соответственно усложняется.

Как обычно, в игре участвуют человек и ПМК. Человек задумывает двузначное натуральное число в пределах 01—99, ПМК отгадывает задуманное, строя числа-гипотезы. Цифра в составе числа, генерируемого ПМК, называется «быком», если она совпадает с цифрой в задуманном числе по значению и занимаемой позиции. Цифра называется «коровой», если она совпадает с цифрой в задуманном числе, но занимает другую позицию.

Игра протекает так. ПМК индицирует свое число. В ответ человек называет количество «быков» и «коров», содержащихся в нем. ПМК на основе полученной информации формирует очередную гипотезу и т. д., пока в гипотезе не будет два «быка».

Не исключено, что из двух цифр в гипотезе ПМК одна является «быком», вторая — «коровой». Например, задумано 31, гипотеза 11. Цифра 1 во второй позиции гипотезы является «быком», в первой — «коровой». Требуя между цифрами задуманного и предложенного чисел взаимно однозначного соответствия, мы отдаём предпочтение «быкам» и утверждаем, что в числе один «бык». Обозначим  $B$ ,  $K$  — количество «быков» и «коров», содержащихся в генерируемом ПМК числе. В основе программы лежит следующий алгоритм игры.

1. Первая гипотеза ПМК: 10.

2. Если в гипотезе два « $B$ », т. е.  $B=2$ , то задача решена.

3. Если  $K=2$ , то переставить цифры в гипотезе местами; после перестановки  $B=2$  задача решена.

4. Если  $K=1$ ,  $B=0$ , то переставить цифры местами. В гипотезе  $B=1$ , но неизвестна его позиция. Перейти к указанию 6.

5. Если  $K=0$ ,  $B=0$ , то для построения очередной гипотезы в предшествующей увеличить каждую цифру на 1. Возврат к указанию 2.

6. В числе  $B=1$  найти его позицию. Сделать пробный ход: увеличить вторую цифру гипотезы на 1. Возможны три случая: если получилось  $B=2$ , то задача решена. Если по-прежнему  $B=1$ , то первая цифра есть «бык». Перейти к указанию 7. Если после пробного хода  $B=0$ , то вторая цифра числа, предшествовавшего пробному ходу, есть «бык». Перейти к указанию 8.

7. Сохраняя первую цифру, увеличить вторую последовательность на 1, пока в числе, генерируемом ПМК, не окажется  $B=2$ . Задача решена.

8. Сохраняя вторую цифру до ее увеличения на 1 в пробном ходе, прибавлять к первой по 1, пока не окажется  $B=2$ . Задача решена.

Например, задумано 21. Первая гипотеза 10 (указание 1) — в числе одна «корова», т. е.  $K=1$ ,  $B=0$ . Указания 2 и 3 пропускаются: не выполнены содержащиеся в них условия. Согласно указанию 4 переставляем цифры —  $B=1$  в числе 01. В соответствии с указанием 6 строим число 02, оказалось  $B=0$ . Обращаемся к указанию 8, получим число 11, т. е.  $B=1$ . Согласно указанию 8 строим число 21 ( $B=2$ ). Задача решена, задумано было 21.

Нетрудно заметить, что число гипотез, необходимых для отгадывания задуманного числа, не более 10. При полном систематическом переборе для решения задачи потребовалось бы в худшем случае 99 ходов. Таков выигрыш от диалоговой организации игры.

В соответствии с алгоритмом построена программа для ПМК.

### Программа БЫКИ и КОРОВЫ

#### Распределение памяти

P4, P5 — первая и вторая цифры гипотезы ПМК, начальные значения соответственно 1 и 0;  
 +P7, P8, P9 — параметры косвенного обращения к подпрограммам проверки окончания (61), формирования и индикации гипотезы ПМК (68), перестановки цифр в гипотезе (73);  
 PA, PB — значения  $B$  и  $K$ ;  
 +PRD — константа для подпрограммы построения гипотез — 10.

|    |        |    |    |        |    |    |        |    |
|----|--------|----|----|--------|----|----|--------|----|
| 00 | CX     | 0Г | 26 | 37     | 37 | 52 | 41     | 41 |
| 01 | X→P5   | 45 | 27 | P→XA   | 6— | 53 | KP→X4  | Г4 |
| 02 | 1      | 01 | 28 | P→XB   | 6L | 54 | П→X5   | 65 |
| 03 | X→P4   | 44 | 29 | +      | 10 | 55 | KПП8   | —8 |
| 04 | ↔      | 14 | 30 | FX = 0 | 5E | 56 | С/П    | 50 |
| 05 | KПП8   | —8 | 31 | 37     | 37 | 57 | X→PA   | 4— |
| 06 | С/П    | 50 | 32 | KП→X4  | Г4 | 58 | KПП7   | —7 |
| 07 | X→PA   | 4— | 33 | KП→X5  | Г5 | 59 | БП     | 51 |
| 08 | ↔      | 14 | 34 | P→X5   | 65 | 60 | 53     | 53 |
| 09 | X→PB   | 4L | 35 | БП     | 51 | 61 | P→XA   | 6— |
| 10 | KПП7   | —7 | 36 | 05     | 05 | 62 | 2      | 02 |
| 11 | P→XB   | 6L | 37 | P→X5   | 65 | 63 | —      | 11 |
| 12 | 2      | 02 | 38 | 1      | 01 | 64 | FX = 0 | 5E |
| 13 | —      | 11 | 39 | +      | 10 | 65 | 72     | 72 |
| 14 | FX = 0 | 5E | 40 | KПП8   | —8 | 66 | Fπ     | 20 |
| 15 | 20     | 20 | 41 | С/П    | 50 | 67 | С/П    | 50 |
| 16 | KПП9   | —9 | 42 | X→PA   | 4— | 68 | P→X4   | 64 |
| 17 | KПП8   | —8 | 43 | KПП7   | —7 | 69 | P→ХД   | 6Г |
| 18 | БП     | 51 | 44 | 1      | 01 | 70 | Х      | 12 |
| 19 | 66     | 66 | 45 | +      | 10 | 71 | +      | 10 |
| 20 | 1      | 01 | 46 | FX = 0 | 5E | 72 | В/О    | 52 |
| 21 | +      | 10 | 47 | 53     | 53 | 73 | P→X4   | 64 |
| 22 | FX = 0 | 5E | 48 | KП→X5  | Г5 | 74 | P→X5   | 65 |
| 23 | 27     | 27 | 49 | P→X5   | 65 | 75 | X→P4   | 44 |
| 24 | KПП9   | —9 | 50 | KПП8   | —8 | 76 | ↔      | 14 |
| 25 | БП     | 51 | 51 | БП     | 51 | 77 | X→P5   | 45 |
|    |        |    |    |        |    | 78 | В/О    | 52 |

| Номер хода | 1          | 2          | 3          | 4          |
|------------|------------|------------|------------|------------|
| Человек    | B/O C/P    | 0 B↑ C/P   | 0 B↑ C/P   | 1 B↑ 0 C/P |
| Время, с   | 4          | 12         | 12         | 12         |
| ПМК        | 10         | 21         | 32         | 24         |
| Номер хода | 5          | 6,7        |            | 8          |
| Человек    | 1 B↑ 0 C/P | 0 B↑ 1 C/P | 0 B↑ 2 C/P |            |
| Время, с   | 12         | 12         |            | 3          |
| ПМК        | 33         | 43,53      |            | π          |

Комментарий. Ход 1: сравнив числа 10 и 53, человек обнаруживает, что  $K=0$ ,  $B=0$ , и во втором ходе вводит 0 B↑ C/P.

Ход 5: человек информирует машину, что после прибавления ко второй цифре единицы  $B=0$ , ПМК начинает увеличивать первую цифру на единицу до тех пор, пока не получит  $B=2$ .

### Структура программы

00—06: построение и индикация начальной гипотезы 10. Ответ человека в этом, как и во всех других случаях, вводится  $K B↑B C/P$ .

07—09: в РА и РВ заносятся значения  $B$  и  $K$ , окончание проверяется в команде 10. Пусть  $K=2$ ; после перестановки местами цифр в гипотезе  $B=2$ .

11—19: построение числа и останов с индикацией  $π$ .

20—26:  $K=1$ , после перестановки цифр в гипотезе  $B=1$ , переход к команде 37 и далее, где анализируется случай  $B=1$ .

27—36:  $K=B=0$ , каждая цифра увеличивается на 1, цикл повторяется, начиная с команды 05.

37—60: наиболее трудный случай  $B=1$ . Вторая цифра гипотезы увеличивается на 1, образовавшееся число индицируется (команды 37—41); если в выведенном на индикатор числе значение  $B$  не уменьшилось по сравнению с преды-

дущим, то первая цифра «бык» (команды 42—52). Остается вторую цифру последовательно увеличить на 1, пока не придем к  $B=2$ . Пусть после увеличения второй цифры в индицируемой гипотезе на единицу  $B=0$ , следовательно, вторая цифра была «быком», ее нужно сохранить в первоначальном виде и, прибавляя к первой цифре по 1, получим  $B=2$  (команды 53—60).

61—67: после получения от человека информации о  $B$  и  $K$  в очередной гипотезе, ПМК решает вопрос о завершении решения по признаку  $B=2$  (подпрограмма проверки окончания).

68—72: каждый ход машины строится из двух цифр  $B$  и  $K$  и индицируется двузначным числом  $10 \cdot B + K$  (подпрограмма формирования числа).

73—78: в некоторых случаях в зависимости от сигнала человека необходимо поменять местами цифры в числе-гипотезе (подпрограмма перестановки цифр).

алг БЫКИ И КОРОВЫ (нат  $L$ )

рез  $L$

нач нат  $M$ , цел  $c_1, c_2, n, d, K, B$

$c_1 := 1; c_2 := 0; M := 10; n := 1$

пока  $n = 1$

нц

выв  $M$ ; ввод  $B, K$

выбор

при  $B = 2: n := 0$

при  $(K = 2) \vee ((K = 1) \wedge (B = 0)) : d := c_1; c_1 := c_2; c_2 := d$

иначе  $c_2 := c_2 + 1; M := 10 * c_1 + c_2$

выв  $M$ ; ввод  $B, K$

выбор

при  $B = 2: n := 0$

при  $B = 1: \text{пока } B \neq 2$

нц

$c_2 := c_2 + 1; M := 10 * c_1 + c_2$

выв  $M$ ; ввод  $B, K$

кц

$n := 0$

иначе  $c_2 := c_2 - 1$

пока  $B \neq 2$

нц

$c_1 := c_1 + 1; M := 10 * c_1 + c_2$

выв  $M$ ; ввод  $B, K$

кц

$n := 0$

все

все

кц

$L := M$

кон

Примечание.  $L$  — задуманное число,  $M$  — гипотеза,  $c_1$  и  $c_2$  — ее цифры,  $n$  — параметр выхода из цикла.

## ЗАДАНИЯ

1. Вы задумали 12, сыграйте с ПМК. Обратите внимание на то, что вторая и третья гипотезы ПМК одинаковы. Чем это объясняется?

2. Переставьте партнеров: ПМК «задумывает», человек отгадывает. Составьте программу.

3. В командах 11—19 анализируется случай  $K=2$ . Это возможно только в том случае, если задумано число 1. Действительно, число 10, индицируемое ПМК в первом ходе, содержит две «коровы». В других случаях вариант  $K=3$  не встречается: как только в игре оказывается  $K=1$ , ПМК приводит число к  $B=1$  перестановкой цифр.

Исключив по договоренности число 1 из репертуара человека, мы получаем возможность сократить программу на девять команд: команда 20 станет 11 и т. д. Внесите в программу необходимые изменения.

## 2.6. ЛОГИКА

— Сыграем? — Илюша снял крышку с длинной плоской коробки, на которой написано: Игра ЛОГИКА. — Это легко, девушка, ты поймешь. Я расставлю цветные фишечки в ряд, ты отгадываешь, в каком порядке они стоят. Потом ты загадываешь, я отгадываю. Кому потребуется меньше вопросов, тот выиграл.

Игра уж очень напоминала игру «Быки и Коровы», я согласился. Выиграв, Илюша предложил: «Давай, составим программу, может, тебе поможет микрокалькулятор».

Имеются фишечки восьми цветов, закодируем их цифрами 0—7. Человек на своей стороне игрового поля ставит в ряд в специальные отверстия пять цветных фишечек в любой последовательности. ПМК в противоположной стороне поля делает ходы — индицирует гипотезы в виде пятизначных чисел, приближающих, «по его мнению», к задуманному (в реальной игре выставляется ряд цветных фишечек).

В ответ человек ставит в том же ряду набор из пяти черных и белых штырьков, возможны пробелы. Белый штырек означает, что ПМК угадал цвет фишечки в соответствующей позиции, черный — фишечка такого цвета входит в задуманный набор, но в другой позиции, пробел — такого цвета в наборе нет.

Картинка кодируется пятизначным числом, состоящим из двоек, единиц и нулей. Двойка соответствует белому штырьку, единица — черному, нуль — пробелу.

Используя полученную информацию, ПМК строит очередное число-гипотезу (на игровом поле для расстановки фишечек происходит сдвиг вперед на один ряд) и т. д. ПМК должен угадать цветовую последовательность фишечек, используя для этого по возможности меньше пробных ходов. Можно предложить следующий алгоритм игры.

Начальный ход ПМК 00000. Человек сравнивает очередной ход ПМК с задуманным числом и вводит подсказку — пятизначное число, состоящее из 0 и 2. Двойка означает, что соответствующая цифра угадана ПМК, нуль — не угадана. Ориентируясь

на полученную информацию, ПМК сохраняет свои цифры в тех позициях, которые соответствуют двойкам, в остальных цифры увеличиваются на 1. Если задача не решена, то процедура повторяется.

В подсказке-числе, вводимом человеком, единицы исключены. Дело в том, что все неотгаданные цифры равны, и понятие совпадения в позициях теряет смысл. Это упрощает структуру программы. Использованную стратегию будем называть (разумеется, условно) методом повторения цифр.

Нетрудно заметить, что максимальное число ходов, необходимых ПМК для решения задачи, — восемь (если хотя бы одна из цифр задуманного числа 7).

### Программа ЛОГИКА

|    |        |    |    |        |    |    |        |    |
|----|--------|----|----|--------|----|----|--------|----|
| 00 | X→П0   | 40 | 21 | 1      | 01 | 41 | 0      | 00 |
| 01 | CX     | 0Г | 22 | —      | 11 | 42 | ×      | 12 |
| 02 | X→ПС   | 4C | 23 | X→ПА   | 4— | 43 | KП→ХВ↑ | ГЕ |
| 03 | X→П6   | 46 | 24 | 1      | 01 | 44 | +      | 10 |
| 04 | KХ→ПВ↑ | LE | 25 | 0      | 00 | 45 | FL0    | 5Г |
| 05 | FL0    | 5Г | 26 | ×      | 12 | 46 | 40     | 40 |
| 06 | 01     | 01 | 27 | —      | 11 | 47 | P→ХС   | 6С |
| 07 | C/P    | 50 | 28 | FX = 0 | 5E | 48 | ↔      | 14 |
| 08 | X→ПА   | 4— | 29 | 34     | 34 | 49 | —      | 11 |
| 09 | 5      | 05 | 30 | KП→ХВ↑ | ГЕ | 50 | FX = 0 | 5E |
| 10 | X→П0   | 40 | 31 | 1      | 01 | 51 | 56     | 56 |
| 11 | P→ХА   | 6— | 32 | +      | 10 | 52 | Fп     | 20 |
| 12 | 1      | 01 | 33 | KХ→ПВ↑ | LE | 53 | C/P    | 50 |
| 13 | 0      | 00 | 34 | FL0    | 5Г | 54 | P→Х6   | 66 |
| 14 | :      | 13 | 35 | 11     | 11 | 55 | C/P    | 50 |
| 15 | 1      | 01 | 36 | KП→Х6  | Г6 | 56 | FBX    | 0  |
| 16 | +      | 10 | 37 | 5      | 05 | 57 | X→ПС   | 4C |
| 17 | X→П7   | 47 | 38 | X→П0   | 40 | 58 | БП     | 51 |
| 18 | KП→Х7  | Г7 | 39 | CX     | 0Г | 59 | 07     | 07 |
| 19 | P→ХА   | 6— | 40 | 1      | 01 |    |        |    |
| 20 | P→Х7   | 67 |    |        |    |    |        |    |

Инструкция. 5 В/О С/П (7 с) « ». При остановах индицируются пятизначные числа — ходы ПМК, прочитываются они справа налево. Вводятся пятизначные числа из 0, 1 или 2. 2 означает, что цифра, выставленная ПМК в соответствующей позиции, совпадает с задуманной, 0 — не совпадает (цифра 1 при данной стратегии игры исключена). Нажать С/П для продолжения вычислений.

Время, необходимое для генерирования и высвечивания одного хода ПМК, 1 мин. Индикация π означает, что задача решена. После нажатия С/П на индикаторе число ходов, использованных ПМК для решения задачи.

Контрольный пример. Задумано 15 231.

5 В/О С/П (7 с) «0» С/П (50 с) «11111» 20002 С/П (45 с) «12221» 20202 С/П (45 с) «13231» 22202 С/П (45 с) «14231».

Обратите внимание на то, что подсказка вводится в ПМК справа налево: не совпадают в числе, генерируемом ПМК и задуманном, вторые цифры, поэтому вместо 20222 вводить надо 22202.

Дальнейшее развитие игры: 22202 С/П (45 с) «15231» 22222 С/П (45 с) «π» — задача решена. После нажатия С/П на индикаторе «6» — число ходов, сделанных ПМК для решения задачи. Впрочем, его можно предсказать заранее, посмотрев на задуманное число.

## Структура программы

- 00—08 : в Р0 организованы пять повторений цикла по числу цифр (команды 00, 09, 10, 37, 38). В Р1—Р5 заносятся нули, формируя первый ход ПМК (команды 01—07). Одновременно засыпаются нули в РС для проверки окончания и в Р6 в качестве начального значения числа ходов ПМК (команды 01—03). При остановах (команда 07) индицируется очередной ход ПМК. Далее подсказка человека вводится в обратной последовательности — справа налево (см. контрольный пример). Нажатием С/П продолжаются вычисления. После команды 08 подсказка хранится в РА.
- 09—27 : в РА последовательно строятся целая часть от деления числа-подсказки на 10, затем целая часть от деления предыдущей целой части на 10 и т. д. В РХ образуются остатки. Если остаток равен нулю, то цифра в соответствующей позиции числа ПМК увеличивается на 1, в противном случае она не изменяется.
- 28—33 : повторение цикла (команды 09—33) пять раз.
- 34, 35 : в Р1—Р5 хранение информации, необходимой для формирования хода ПМК.
- 37—46 : построение хода ПМК.
- 47—57 : проверка окончания — совпадают ли два последних числа. Число π на индикаторе — признак окончания.
- 58—59 : повторение цикла вычислений в противном случае.

— С твоей программой, дедушка, ничего не остается, как проигрывать, — сказал Илюша, выиграв соревнование с крупным счетом. — Да уже после второй партии можно предсказать, как будет играть машинка.

— Туповата, — вынужден был я признать. — Впрочем, знаешь что — поменяйтесь ролями: пусть машинка «задумывает», а ты отгадываешь. Тогда и меня освободишь, и получишь безотказного честного партнера.

— Такая логика меня устраивает. Давай составим программу, — согласился внук.

Как обычно, распределим память ПМК. Пусть цифры «задуманного» случайного пятизначного числа 0, 1, ..., 7 хранятся в регистрах 9, А, В, С, Д.

Остановимся на самой простой идее. Ты вводишь по очереди цифры предполагаемого числа. ПМК сравнивает очередную цифру со всеми «задуманными». Совпадение цифр и их позиций обозначается числом 2, вхождение предполагаемой цифры в задуманное число в другой позиции — единица, отсутствие этой цифры в задуманном числе — нуль. На основании полученных ре-

зультатов строится пятизначное число. Если в числе пять 2, то задача решена.

## Программа ЛОГИКА. МАШИНА-ПАРТНЕР

|    |       |    |    |        |    |    |      |    |
|----|-------|----|----|--------|----|----|------|----|
| 00 | X→P8  | 48 | 24 | X→P5   | 45 | 48 | 2    | 02 |
| 01 | 8     | 08 | 25 | C/P    | 50 | 49 | X→P7 | 47 |
| 02 | X→P6  | 46 | 26 | X→P8   | 48 | 50 | БП   | 51 |
| 03 | 5     | 05 | 27 | KP→X5  | Г5 | 51 | 56   | 56 |
| 04 | X→P0  | 40 | 28 | 5      | 05 | 52 | 1    | 01 |
| 05 | P→X8  | 68 | 29 | X→P1   | 41 | 53 | X→P7 | 47 |
| 06 | ВП    | 0C | 30 | 0      | 00 | 54 | FL0  | 5Г |
| 07 | 9     | 09 | 31 | X→P4   | 44 | 55 | 38   | 38 |
| 08 | F cos | ИГ | 32 | 8      | 08 | 56 | P→X4 | 64 |
| 09 | F cos | -1 | 33 | X→P6   | 46 | 57 | 1    | 01 |
| 10 | Fπ    | 20 | 34 | 5      | 05 | 58 | 0    | 00 |
| 11 | :     | 13 | 35 | X→P0   | 40 | 59 | ×    | 12 |
| 12 | X→P8  | 48 | 36 | 0      | 00 | 60 | P→X7 | 67 |
| 13 | 7     | 07 | 37 | X→P7   | 47 | 61 | +    | 10 |
| 14 | ×     | 12 | 38 | P→X8   | 68 | 62 | X→P4 | 44 |
| 15 | 1     | 01 | 39 | KP→X6  | Г6 | 63 | Fπ   | 20 |
| 16 | +     | 10 | 40 | —      | 11 | 64 | С/П  | 50 |
| 17 | X→P2  | 42 | 41 | FX = 0 | 5E | 65 | X→P8 | 48 |
| 18 | KP→X2 | Г2 | 42 | 54     | 54 | 66 | FL1  | 5L |
| 19 | P→X2  | 62 | 43 | P→X1   | 61 | 67 | 32   | 32 |
| 20 | KX→P6 | L6 | 44 | P→X0   | 60 | 68 | P→X4 | 64 |
| 21 | FL0   | 5Г | 45 | —      | 11 | 69 | БП   | 51 |
| 22 | 05    | 05 | 46 | FX = 0 | 5E | 70 | 25   | 25 |
| 23 | 0     | 00 | 47 | 52     | 52 |    |      |    |

**Инструкция.** Наберите произвольное неотрицательное число  $A < 10^5$  В/О С/П (переключатель в положении Р). Через 45 с ПМК останавливается. Это значит, что машина «задумала» число и предлагает вам ввести первую цифру гипотезы.

Наберите свою цифру и, нажав С/П, включите продолжение вычислений. Через 15 с индицируются восемь знаков числа π — сообщение, что ПМК готов ко вводу следующей цифры. После ее набора и нажатия С/П ждите очередной индикации π и т. д. Когда введены все цифры данной гипотезы, на экране появляется пятизначное число. Если ответ ПМК «22222», то задача решена. Команда P→X5 выводит на индикатор число ходов-гипотез, которые вы сделали, чтобы решить задачу.

**Контрольный пример.** Переключатель углов в положении Р, 9,75 В/О С/П (45 с) «0».

5 С/П (15 с) «π», 1 С/П «π», 0 С/П «π», 4 С/П «π», 6 С/П «π» С/П (3 с) «20201». Человек предложил 51046, ПМК проинформировал 20201: цифры 5 и 0 угаданы, цифра 6 входит в другой позиции, цифры 1 и 4 вовсе не входят в задуманное число.

5 С/П «π», 6 С/П «π», 0 С/П «π», 3 С/П «π», 2 С/П «π», С/П «22200» — первые три цифры верные.

Последовательно вводим: 5, 6, 0, 7, 5 С/П «22202».

5, 6, 0, 5, 5 С/П «22222» — задача решена.

P→X5 «4». Для решения задачи потребовалось четыре хода.

## Структура программы

Программа состоит из двух вложенных циклов: внешнего по числу цифр в гипотезе и внутреннего по числу цифр в задуманном числе. При каждом прохождении внешнего цикла (команды

32—67) внутренний повторяется пять раз (команды 38—55); первый цикл управляет счетчиком Р1, второй — Р0. Кроме того, имеется цикл по числу вводов гипотез, содержащий первые два. Обращение к нему управляет человеком (команды 25—70). Выход из цикла происходит, когда индицируется число 22222. Рассмотрим структуру программы подробнее.

00—22 : генерируются цифры случайного пятизначного числа в восьмеричной системе счисления, они запоминаются в Р9—РД. Формула для вычислений:  $x_{n+1} = 7 \arccos(\cos 10^9 x_n)$ , где  $x_0 = A$  — произвольно, но не отрицательно и меньше 10;

23, 24, 27 : в Р5 подсчет числа гипотез. Начальное значение Р5 нуль. С каждой новой гипотезой к нему прибавляется единица;

25, 26 : ПМК останавливается для индикации — вначале индицируется 0, при следующих остановах — пятизначные числа. Человек вводит первую цифру очередной гипотезы, после нажатия С/П она запоминается в Р8;

28—31 : подготовка данных для внешнего цикла;

32—37 : подготовка данных для внутреннего цикла;

56—64 : подготовка числа для индикации, хранение его в Р4; после останова с индикацией π вводится и запоминается в Р8 очередная цифра данной гипотезы.

### ЗАДАНИЯ

1. Для программ составьте соответствующие алгоритмы на УАЯ.

2. Составьте программу для модификации игры, в которой после ввода человеком очередной цифры своей гипотезы ПМК — вместо «нейтрального» π, как это было ранее, информирует сигналами 2, 1, 0 о типе совпадения.

Имейте в виду — найденная ПМК цифра для построения пятизначного числа из 2, 1, 0 хранится в Р7.

Станет ли игра легче для человека, если учесть, что с помощью одного хода можно ввести не более пяти цифр и что ПМК в любом случае после проверки всех цифр выдаст на экран пятизначное число из двоек, единиц, нулей? Проверьте экспериментально, уменьшится ли в этом случае число гипотез, необходимых для решения задачи.

3. Чтобы сократить программу и время вычисления ПМК, человек набирает отдельные цифры, которые последовательно сравниваются ПМК с цифрами задуманного числа. Для ввода одного числа требуется пять остановов ПМК и нажатий С/П. Преобразуйте программу так, чтобы человек имел возможность ввести гипотезу единственным пятизначным числом.

4. ПМК информирует человека о близости его гипотезы к задуманному числу. Оставив за ПМК роль партнера, задумывающего число, предложим ему выдавать свою информацию на экран двузначными числами: первая цифра — количество «двоек», т. е. совпадений цифр в позициях, вторая — «единиц», т. е. вхождений цифр в задуманное число в других позициях. Теперь игра усложнилась, но, по-видимому, стала интересней. Составьте программу для ПМК и запишите соответствующий алгоритм на УАЯ.

### ОТВЕТЫ

| 4. Программа МАШИНА-ПАРТНЕР |                     | (индикация двузначным числом) |                        |
|-----------------------------|---------------------|-------------------------------|------------------------|
| 00                          | X→P8                | 48                            | 30 0 00 59             |
| 01                          | 8                   | 08                            | 31 X→P4 44 60 FX=0 5E  |
| 02                          | X→P6                | 46                            | 32 X→P6 46 61 72 72    |
| 03                          | 5                   | 05                            | 33 1 01 62 П→X1 61     |
| 04                          | X→P0                | 40                            | 34 4 04 63 П→X0 60     |
| 05                          | P1→X8               | 68                            | 35 X→P2 42 64 — 11     |
| 06                          | BП                  | 0C                            | 36 5 05 65 FX=0 5E     |
| 07                          | 9                   | 09                            | 37 X→P0 40 66 70 70    |
| 08                          | F cos               | 1Г                            | 38 П→X3 63 67 КП→X4 Г4 |
| 09                          | F cos <sup>-1</sup> | 1—                            | 39 1 01 68 БП 51       |
| 10                          | Fπ                  | 20                            | 40 0 00 69 78 78       |
| 11                          | :                   | 13                            | 41 : 13 70 1 01        |
| 12                          | X→P8                | 48                            | 42 1 01 71 X→P7 47     |
| 13                          | 7                   | 07                            | 43 + 10 72 FL0 5Г      |
| 14                          | ×                   | 12                            | 44 X→P3 43 73 57 57    |
| 15                          | 1                   | 01                            | 45 КП→X3 Г3 74 П→X6 66 |
| 16                          | +                   | 10                            | 46 FO 25 75 П→X7 67    |
| 17                          | X→P2                | 42                            | 47 П→X3 63 76 + 10     |
| 18                          | KП→X2               | Г2                            | 48 — 11 77 X→P6 46     |
| 19                          | P→X2                | 62                            | 49 1 01 78 FL1 5L      |
| 20                          | KХ→P6               | L6                            | 50 0 00 79 33 33       |
| 21                          | FL0                 | 5Г                            | 51 × 12 80 П→X4 64     |
| 22                          | 05                  | 05                            | 52 FBX 0 81 1 01       |
| 23                          | 0                   | 00                            | 53 — 11 82 0 00        |
| 24                          | X→P5                | 45                            | 54 X→P8 48 83 × 12     |
| 25                          | C/П                 | 50                            | 55 0 00 84 П→X6 66     |
| 26                          | X→P3                | 43                            | 56 X→P7 47 85 + 10     |
| 27                          | KП→X5               | Г5                            | 57 П→X8 68 86 БП 51    |
| 28                          | 5                   | 05                            | 58 КП→X2 Г2 87 25 25   |
| 29                          | X→P1                | 41                            |                        |

**Инструкция.** Наберите любое неотрицательное число, меньшее 10 В/О С/П. Через 45 с на индикаторе появляется 0. Введите свою первую гипотезу пятизначным числом; примерно через 2 мин на индикаторе будет двузначное число: первая цифра — количество цифр в вашей гипотезе, которые входят в задуманное число и стоят в той же позиции; вторая — количество цифр, входящих в задуманное число, но в другой позиции. Наберите следующую гипотезу, нажав С/П, продолжите вычисления, ждите индикации следующего двузначного числа и т. д. Индикация числа 50 свидетельствует о завершении решения: ваша последняя гипотеза совпадает с задуманным числом.

Нажмите П→X5, и ПМК сообщит, сколько ходов вам потребовалось для решения задачи.

**Контрольный пример.** 9,75 В/О С/П (45 с) «0», 51046 С/П (2 мин) «21», 56032 С/П (2 мин) «30», 56045 С/П «40», 56065 С/П «41», 56055 С/П «50». Задача решена, задумано 56055. П→X5 «5» — на решение задачи потребовалось пять ходов.

### Структура программы

00—22 : генерирование случайного пятизначного числа;

23—27 : нуль заносится в Р5 (счетчик числа ходов человека), после останова пятизначным числом вводится гипотеза, она запоминается в Р3, и к счетчику ходов прибавляется 1;

28—87 : внешний цикл формирования и индикации двузначного числа; подготовка данных (команды 28—32);

33—79 : внутренний цикл накопления в Р4 числа «двоек» и в Р6 — «единиц» по мере сравнения каждой цифры гипотезы со всеми цифрами задуманного числа, подготовка данных (команды 33—37).

## 2.7. ЛОВЛЯ БЛОХИ (АЛГОРИТМ ПРОТИВ СЛУЧАЯ)

Задели собаку блохи, совсем жизни нет. Пришла она к человеку: «Избавь меня от блох, буду служить тебе верой и правдой». Человек подумал и вывел блохолова. Прекрасное получилось насекомое, далеко видело, могло часами, притаившись, выжидать добычу, но вот беда — несообразительное. И тогда человек снабдил блохолова инструкцией, она не увеличила скорости, но привела в порядок движение. Чтобы проверить, хороша ли инструкция, человек описал хаотичные прыжки блохи и ходы блохолова числами и составил программу для ПМК. Эта программа имитировала поведение блохолова, поэтому подобные программы называют имитационными.

Сформулируем условие задачи. Квадратная доска  $n \times n$  ( $n \leq 9$ ) разбита на клетки. В одной из них находится блоха, она беспорядочно перепрыгивает с одной клетки на другую. На доске притаился блохолов, он может перемещаться только на одну клетку — вправо, влево, вверх, вниз. Требуется составить алгоритм и на его основе программу, позволяющую блохолову поймать добычу.

Присвоим каждой клетке координаты: абсцисса — номер столбца, ордината — номер строки, на пересечении которых находится клетка. За расстояние между двумя клетками  $(a, b)$  и  $(c, d)$  примем величину  $|c-a| + |d-b|$ . Обозначим: блоха —  $A$ , блохолов —  $D$ .

Делая ход,  $D$  выбирает такое из четырех направлений, чтобы расстояние до  $A$  уменьшилось.  $A$  делает случайные прыжки, координаты занимаемой клетки вычисляются по формуле  $x_{n+1} = (n/\pi) \arccos(\cos 10^\circ x_n)$ , где  $x_0 < 10$  произвольно.  $D$  начинает,

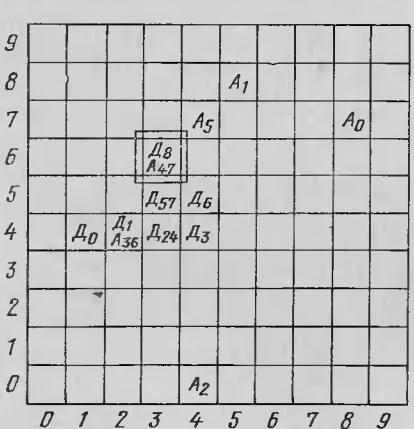
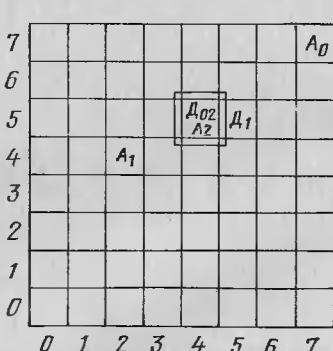


Рис. 10



далее ходы чередуются. После каждого хода проверяется окончание игры: совпадают ли координаты  $A$  и  $D$ .

## Программа ЛОВЛЯ БЛОХИ

|    |        |    |    |                 |    |    | СХ                  | 0Г |
|----|--------|----|----|-----------------|----|----|---------------------|----|
| 00 | X→П3   | 43 | 30 | KП→X6           | Г6 | 60 | 4                   | 04 |
| 01 | F0     | 25 | 31 | БП              | 51 | 61 | X→П0                | 40 |
| 02 | X→П1   | 41 | 32 | 39              | 39 | 62 | П→Х1                | 61 |
| 03 | F0     | 25 | 33 | FX ≠ 0          | 57 | 63 | 9                   | 09 |
| 04 | X→П6   | 46 | 34 | 58              | 58 | 64 |                     |    |
| 05 | ↔      | 14 | 35 | П→X6            | 66 | 65 | F10 <sup>x</sup>    | 15 |
| 06 | X→П5   | 45 | 36 | 1               | 01 | 66 | ×                   | 12 |
| 07 | СХ     | 0Г | 37 | —               | 11 | 67 | F cos               | 1Г |
| 08 | X→П4   | 44 | 38 | X→П6            | 46 | 68 | F cos <sup>-1</sup> | 1— |
| 09 | П→X5   | 65 | 39 | П→X5            | 65 | 69 | Fπ                  | 20 |
| 10 | П→X1   | 61 | 40 | 1               | 01 | 70 | :                   | 13 |
| 11 | —      | 11 | 41 | 0               | 00 | 71 | П→Х9                | 69 |
| 12 | FX < 0 | 5C | 42 | ×               | 12 | 72 | ×                   | 12 |
| 13 | 17     | 17 | 43 | П→X6            | 66 | 73 | X→П1                | 41 |
| 14 | KП→X5  | Г5 | 44 | +               | 10 | 74 | 1                   | 01 |
| 15 | БП     | 51 | 45 | C/P             | 50 | 75 | +                   | 10 |
| 16 | 39     | 39 | 46 | KП→X4           | Г4 | 76 | X→П2                | 42 |
| 17 | FX ≠ 0 | 57 | 47 | П→X5            | 65 | 77 | KП→Х2               | Г2 |
| 18 | 25     | 25 | 48 | П→X1            | 61 | 78 | П→Х2                | 62 |
| 19 | П→X5   | 65 | 49 | —               | 11 | 79 | KХ→П0               | L0 |
| 20 | 1      | 01 | 50 | FX <sup>2</sup> | 22 | 80 | FL0                 | 5Г |
| 21 | —      | 11 | 51 | П→X6            | 66 | 81 | 63                  | 63 |
| 22 | X→П5   | 45 | 52 | П→X3            | 63 | 82 | 1                   | 01 |
| 23 | БП     | 51 | 53 | —               | 11 | 83 | 0                   | 00 |
| 24 | 39     | 39 | 54 | FX <sup>2</sup> | 22 | 84 | ×                   | 12 |
| 25 | П→X6   | 66 | 55 | +               | 10 | 85 | П→Х3                | 63 |
| 26 | П→X3   | 63 | 56 | FX = 0          | 5E | 86 | +                   | 10 |
| 27 | —      | 11 | 57 | 60              | 60 | 87 | C/P                 | 50 |
| 28 | FX < 0 | 5C | 58 | Fπ              | 20 | 88 | БП                  | 51 |
| 29 | 33     | 33 | 59 | C/P             | 50 | 89 | 09                  | 09 |

Инструкция. Переключатель углов на Р. Нумерация сторон квадрата начинается с нуля, поэтому в Р9 вводится  $n-1$ , где  $n$  — число единиц в строке. Через стек вводятся координаты начальных позиций  $D$  ( $a, b$ ) и  $A$  ( $c, d$ ):  $a$  В↑  $b$  В↑  $c$  В↑  $d$  В/O C/P.

Через 5 с на индикаторе появляется двузначное число, описывающее позицию  $D$  после хода: первая цифра — номер столбца (абсцисса), вторая — номер строки (ордината) занятой клетки. Если число однозначно, значит первая координата нуль.

Через 25 с после нажатия С/P на индикаторе выводятся координаты текущей клетки, занятой  $A$  после прыжка, С/П. Далее ходы чередуются. Сигналом, что задача решена, является появление на индикаторе восьми знаков л. Из Р4 можно вызвать число ходов, потребовавшихся  $D$  для решения задачи.

Контрольные примеры. Индексы при буквах  $A$  и  $D$  означают номера ходов  $A_0$  и  $D_0$  — начальные позиции.

$D_0$  (1; 4),  $A_0$  (8; 7) (рис. 10, а).

9 X→П9 1 В↑ 4 В↑ 8 В↑ 7 В/O С/P (7 с) «24» —  $D_1$ , С/П (25 с) «58» —  $A_1$ , С/П (5 с) «34» —  $D_2$ , С/П (25 с) «40» —  $A_2$ , С/П «44» —  $D_3$ , С/П «24» —  $A_3$  и т. д.

Индикация л означает, что задача решена. П→Х4 «8» — на восьмом ходу охотник поймал добычу в клетке (3; 6).

$D_0$  (4; 5),  $A_0$  (7; 7),  $n=8$  (рис. 10, б).

7 X→П9 4 В↑ 5 В↑ 7 В↑ В/O С/P.

На рисунке отмечена динамика перемещений. П $\rightarrow$ X4 «2» — задача решена за два хода. В отличие от предыдущего случая теперь блоха сама сократила время охоты, прыгнув навстречу.

На этот раз, не разбирая структуру программы подробно, отметим назначение отдельных фрагментов. Ходы  $D$  для случаев, когда он находится левее, правее, ниже, выше  $A$  (команды 14, 19—22, 30, 35—38). По вычисленным координатам строится и индицируется двузначным числом (команды 39—46) очередной ход  $D$ ; одновременно к накопителю числа ходов прибавляется 1 (команды 47—59), проверяется окончание после хода  $D$ , формируется и выводится случайный «прыжок»  $A$  (команды 60—87). Далее процедура повторяется (команды 88, 89).

## 2.8. РЕЙТИНГ

Рейтинг, или индивидуальный коэффициент, есть число, характеризующее силу игры. Одна из наиболее популярных шкал отсчета построена американским математиком Эло для шахматистов (табл. 5). Выступающему в турнире высшего класса шахматисту впервые присваивается начальный рейтинг 2200. Участие в

Таблица 5

| $ K_{ct}-K_t $ ,<br>$K_b-K_m$ | $n_{ож\ b}$ | $n_{ож\ m}$ | $ K_{ct}-K_t $ ,<br>$K_b-K_m$ | $n_{ож\ b}$ | $n_{ож\ m}$ |
|-------------------------------|-------------|-------------|-------------------------------|-------------|-------------|
| 0—3                           | 0,50        | 0,50        | 198—206                       | 0,76        | 0,24        |
| 4—10                          | 0,51        | 0,49        | 207—215                       | 0,77        | 0,23        |
| 11—17                         | 0,52        | 0,48        | 216—225                       | 0,78        | 0,22        |
| 18—25                         | 0,53        | 0,47        | 226—235                       | 0,79        | 0,21        |
| 26—32                         | 0,54        | 0,46        | 236—245                       | 0,80        | 0,20        |
| 33—39                         | 0,55        | 0,45        | 246—256                       | 0,81        | 0,19        |
| 40—46                         | 0,56        | 0,44        | 257—267                       | 0,82        | 0,18        |
| 47—53                         | 0,57        | 0,43        | 268—278                       | 0,83        | 0,17        |
| 54—61                         | 0,58        | 0,42        | 279—290                       | 0,84        | 0,16        |
| 62—68                         | 0,59        | 0,41        | 291—302                       | 0,85        | 0,15        |
| 69—76                         | 0,60        | 0,40        | 303—315                       | 0,86        | 0,14        |
| 77—83                         | 0,61        | 0,39        | 316—328                       | 0,87        | 0,13        |
| 84—91                         | 0,62        | 0,38        | 329—344                       | 0,88        | 0,12        |
| 92—98                         | 0,63        | 0,37        | 345—357                       | 0,89        | 0,11        |
| 99—106                        | 0,64        | 0,36        | 358—374                       | 0,90        | 0,10        |
| 107—113                       | 0,65        | 0,35        | 375—391                       | 0,91        | 0,09        |
| 114—121                       | 0,66        | 0,34        | 392—411                       | 0,92        | 0,08        |
| 122—129-                      | 0,67        | 0,33        | 412—432                       | 0,93        | 0,07        |
| 130—137                       | 0,68        | 0,32        | 433—456                       | 0,94        | 0,06        |
| 138—145                       | 0,69        | 0,31        | 457—484                       | 0,95        | 0,05        |
| 146—153                       | 0,70        | 0,30        | 485—517                       | 0,96        | 0,04        |
| 154—162                       | 0,71        | 0,29        | 518—559                       | 0,97        | 0,03        |
| 163—170                       | 0,72        | 0,28        | 560—619                       | 0,98        | 0,02        |
| 171—179                       | 0,73        | 0,27        | 620—735                       | 0,99        | 0,01        |
| 180—188                       | 0,74        | 0,26        | 736 и выше                    | 1,00        | 0,00        |
| 189—197                       | 0,75        | 0,25        |                               |             |             |

турнире влияет на рейтинг его участников: в случае успеха он увеличивается, неудачи — уменьшаются.

Новый рейтинг участников турнира (матча) вычисляется по формуле

$$K_n = K_{st} + 10(N - N_{ож}), \quad (1)$$

где  $K_{st}$ ,  $K_n$  — соответственно старый и новый рейтинги данного участника турнира;  $N$  — число фактически набранных им очков;  $N_{ож}$  — ожидаемое от шахматиста в турнире число очков. Наибольшие трудности вызывает вычисление  $N_{ож}$ .

Рассмотрим простейший случай: играют двое в матче из  $m$  партий. Дано:  $K_{st1}$ ,  $K_{st2}$  — рейтинги первого и второго участников;  $N_1$ ,  $N_2$  — число фактически набранных игроками очков;  $m$  — число сыгранных партий. Условимся, что  $K_b$  и  $K_m$  — большее и меньшее из  $K_{st1}$  и  $K_{st2}$ ;  $n_{ож\ b}$  и  $n_{ож\ m}$  — число ожидаемых очков в одной партии для игроков с большим и меньшим рейтингом (оно находится по таблице Эло), требуется вычислить  $K_{n1}$  и  $K_{n2}$ .

Алгоритм вычисления новых рейтингов

1. Вычислить  $K_b - K_m$ .
2. По таблице Эло найти  $n_{ож\ b}$  и  $n_{ож\ m}$  (контроль:  $n_{ож\ b} + n_{ож\ m} = m$ ).
3. Вычислить  $N_{ож\ b}$  и  $N_{ож\ m}$  по формулам  $N_{ож\ b} = m n_{ож\ b}$ ,  $N_{ож\ m} = m n_{ож\ m}$  (контроль:  $N_{ож\ b} + N_{ож\ m} = m$ ).
4. Результаты округлить до целого числа.
5. Вычислить по формуле (1)  $K_{n1}$  и  $K_{n2}$ .

Пример. 1. Исходные данные Московского матча на звание чемпиона мира по шахматам между А. Карповым и Г. Каспаровым в 1985 г.:  $K_{st1}=2717$ ,  $K_{st2}=2700$ ,  $m=24$ . Выполняем указания алгоритма:

$$1) 17. 2) 0,52; 0,48. 3) 12,48; 11,52. 4) 12; 12. 5) K_{n1} = 2717 + 10(11,5 - 12) = 2712, K_{n2} = 2700 + 10(13,5 - 12) = 2715.$$

Существует и несколько отличная методика вычисления рейтингов, более удачная, если в турнире участвуют много людей. Согласно ей вычисляется среднее арифметическое рейтингов всех участников турнира, и это число условно принимается за рейтинг турнира  $K_t$ . Мы этим как бы считаем, что каждый шахматист играет с одним и тем же «средним» противником. Для игрока с рейтингом  $K$  находится  $K - K_t$  и далее предыдущая процедура повторяется, с тем отличием, что для данного игрока по таблице Эло выбирается  $n_{ож\ b}$  или  $n_{ож\ m}$  в зависимости от условий  $K > K_t$  или  $K < K_t$ . Воспользуемся этой методикой.  $K_t = (2717 + 2700)/2 = 2708,5$

Первый игрок — Карпов:  $2717 - 2708,5 = 8,5$ ;  $n_{ож\ b} = 0,51$  (табл. 5);  $N_{ож\ b} = 0,51 \cdot 24 = 12$  (после округления);  $K_{n1} = 2717 + 10(11,5 - 12) = 2712$ .

Второй игрок — Каспаров:  $2708 - 2700 = 8,5$ ;  $n_{ож\ m} = 0,49$ ;  $N_{ож\ m} = 12$  (после округления);  $K_{n2} = 2700 + 10(13,5 - 12) = 2715$ .

Результаты, как видим, совпадают с предыдущими.

**Пример 2.** В турнире играют четверо, каждый с каждым по одной партии. Результаты турнира сведены в таблицу (табл. 6).

Таблица 6

| Номер игрока | 1   | 2   | 3   | 4 | $N$ |
|--------------|-----|-----|-----|---|-----|
| 1            | —   | 1   | 0,5 | 1 | 2,5 |
| 2            | 0   | —   | 0,5 | 1 | 1,5 |
| 3            | 0,5 | 0,5 | —   | 0 | 1   |
| 4            | 0   | 0   | 1   | — | 1   |

Известно:  $n=4$ ,  $K_{ct1}=2250$ ;  $K_{ct2}=2380$ ;  $K_{ct3}=2200$ ;  $K_{ct4}=2290$ ;  $N_1=2,5$ ;  $N_2=1,5$ ,  $N_3=1$ ,  $N_4=1$ ;  $m=3$  (каждый играет три партии);  $K_t=(2250+2380+2200+2290)/4=2280$ .

Выполняем указания алгоритма по турнирной методике;

1)  $|K_{ct1}-K_t|=30$ , 2)  $n_{ojk}=0,46$  (взято из таблицы  $n_{ojk}$ , так как  $K_{ct1} < K_t$ ), 3)  $N_{ojk}=m n_{ojk}=0,46 \cdot 3=1,38$ , 4)  $N_{ojk} \approx 1$  — округлено до целого, 5)  $K_{h1}=2250+10(2,5-1)=2265$  — рейтинг первого игрока после окончания матча.

Аналогично вычисляются рейтинги остальных игроков.

Ответы:  $K_{h2}=2375$ ,  $K_{h3}=2200$ ,  $K_{h4}=2280$ .

#### Программа РЕЙТИНГ

|    |      |    |    |                  |    |    |      |    |
|----|------|----|----|------------------|----|----|------|----|
| 00 | X→П0 | 40 | 17 | F0               | 25 | 34 | 7    | 07 |
| 01 | X→П1 | 41 | 18 | X→ПС             | 4C | 35 | —    | 11 |
| 02 | 0    | 00 | 19 | П→ХД             | 6Г | 36 | П→ХВ | 6L |
| 03 | X→ПД | 4Г | 20 | —                | 11 | 37 | ↔    | 14 |
| 04 | C/P  | 50 | 21 | FX <sup>2</sup>  | 22 | 38 | —    | 11 |
| 05 | П→ХД | 6Г | 22 | F V <sup>-</sup> | 21 | 39 | —    | 01 |
| 06 | +    | 10 | 23 | C/P              | 50 | 40 | 0    | 00 |
| 07 | X→ПД | 4Г | 24 | П→XA             | 6— | 41 | ×    | 12 |
| 08 | FL0  | 5Г | 25 | ×                | 12 | 42 | П→ХС | 6C |
| 09 | 04   | 04 | 26 | B↑               | 0E | 43 | +    | 10 |
| 10 | П→Х1 | 61 | 27 | СХ               | 0Г | 44 | C/P  | 50 |
| 11 | :    | 13 | 28 | BП               | 0C | 45 | FL1  | 5L |
| 12 | X→ПД | 4Г | 29 | 7                | 07 | 46 | 13   | 13 |
| 13 | C/P  | 50 | 30 | +                | 10 | 47 | Fп   | 20 |
| 14 | X→ПА | 4— | 31 | B↑               | 0E | 48 | C/P  | 50 |
| 15 | —F0  | 25 | 32 | СХ               | 0Г | —  | —    | —  |
| 16 | X→ПВ | 4L | 33 | BП               | 0C | —  | —    | —  |

#### Инструкция

1. Вводится  $n$  — число игроков в турнире (для матча  $n=2$ ) В/О С/П.
2. Индицируется 0. Ввести рейтинг первого игрока С/П, после останова рейтинг второго игрока С/П и т. д. Когда все рейтинги введены, индицируется

$K_t$  (рейтинг «среднего» игрока), его необходимо запомнить для последующего вычисления  $n_{ojk}$  по таблице Эло.

3. Ввод:  $K_{ct}$   $B\uparrow N$   $B\uparrow m$  С/П ( $K_{ct}$  — известный еще до начала игры рейтинг данного игрока;  $N$  — число очков, фактически набранных им в соревновании;  $m$  — количество партий, которое сыграл участник соревнования). Индицируется  $|K_{ct}-K_t|$ . С помощью этого числа и известного значения  $K_t$  по таблице находится  $n_{ojk}$ , если  $K_{ct}>K_t$ , и  $n_{ojk}$  в противном случае.

4. Ввод:  $n_{ojk}$  С/П. Индицируется значение  $K_p$  данного игрока.

5. Нажатие С/П — процедура повторяется, начиная с п. 3, однако уже для следующего игрока и т. д.

Индикация π означает, что задача решена.

Для контроля решим по программе пример, рассмотренный выше:

4 В/О С/П «0», 2250 С/П 2380 С/П 2200 С/П 2290 С/П.  $\langle K_t=2280 \rangle$ . 2250  $B\uparrow 2.5$   $B\uparrow 3$  С/П «30». Находим по таблице Эло интервал для числа 30: 26—32. Так как 2250  $< K_t=2280$ , то берем в таблице число 0,46, а не 0,54. 0,46 С/П,  $\langle K_h=2265 \rangle$  — ответ совпадает с полученным ранее. Далее С/П, после останова вводятся данные второго игрока 2380  $B\uparrow 1.5$   $B\uparrow 3$  С/П, вся процедура повторяется для второго игрока и т. д. После получения всех значений  $K_h$  ПМК индикацией π информирует о завершении решения.

#### Структура программы

00—03 : ввод начальных данных.

04—09 : вычисление суммы рейтингов всех участников, ее сохранение в РД.

10—13 : вычисление среднего арифметического всех рейтингов участников  $K_t$  и его запись в РД.

14—18 : запись  $m$  в РА,  $N$  в РВ,  $K_{ct}$  в РС.

19—23 : вычисление значения  $|K_{ct}-K_t|$ .

24, 25 : вычисление  $N_{ojk}$ .

26—35 : округление  $N_{ojk}$  до целого числа.

36—44 : вычисление и индикация  $K_h$  очередного игрока.

45, 46 : повторение цикла для следующего игрока;  $K_{ct}$   $B\uparrow N$   $B\uparrow m$  С/П и т. д.

47, 48 : индикация π — задача решена.

Задача решалась в диалоговом режиме человек — ПМК. Однако распределение функций между компонентами системы здесь несколько отлично от рассмотренных ранее задач диалогового типа,

Человек вводил поочередно в ПМК данные всех игроков ПМК, вычислял по ним значения  $|K_{ct}-K_t|$ . Человек по выведенному ПМК результату находил в таблице Эло значения  $n_{ojk}$  для игрока. ПМК, на основе введенного человеком значения  $n_{ojk}$ , вычислял и индицировал его рейтинг после окончания турнира.

Человек избавлен от утомительных вычислений, особенно при большом числе играющих и сыгранных партий, а от ПМК не требуется запоминать таблицу, для которой не хватает емкости памяти. Если в предшествующих задачах человек вначале уменьшал число вариантов, предоставив ПМК «доводку» решения в доступной ему ограниченной области, то в данном случае действия человека и ПМК чередуются на протяжении всего решения задачи.

Алгоритм, записанный на УАЯ, поможет лучше разобраться в программе для ПМК или составить программу для компьютера.

Адрес команды алг РЕЙТИНГ (нат  $n, m$ , nat таб  $K_{\text{ст}}[1:n], K_{\text{н}}[1:n], N[1:n]$   
арг  $n, m, K_{\text{ст}}, N, \mathcal{E}_1, \mathcal{E}_2$   
рез  $K_{\text{н}}$   
нач nat  $i, j$ , цел  $N_{\text{ож}}$ ,  $B$ ,вещ  $K_{\text{т}}, A, n_{\text{ож}}$   
ком таблица  $\mathcal{E}_1$ —первые числа столбца 1 таблицы Эло,  $\mathcal{E}_2$ —элементы столбца 2 таблицы Эло  
 $K_t := 0$   
для  $i$  от 1 до  $n$   
нц  
     $K_t := K_t + K_{\text{ст}}$   
кц  
     $K_t := K_t/n$   
для  $i$  от 1 до  $n$   
нц  
         $A := |K_{\text{ст}}[i] - K_t|; B := 1$   
        для  $j$  от 1 до 50  
            нц  
                если ( $A \geq \mathcal{E}_1[j] \wedge A < \mathcal{E}_1[j+1]$ )  
                    то  $B := 0$ ; если  $K_{\text{ст}}[i] < K_t$   
                        то  $n_{\text{ож}} := 1 - \mathcal{E}_2[j]$   
                        иначе  $n_{\text{ож}} := \mathcal{E}_2[j]$   
                все  
                 $j := 50$   
            все  
        кц  
        если  $B = 1$   
            то если  $K_{\text{ст}}[i] < K_t$   
                то  $n_{\text{ож}} := 0$   
                иначе  $n_{\text{ож}} := 1$   
            все  
        все  
         $A := m * n_{\text{ож}}; N_{\text{ож}} := A$   
        если  $A - N_{\text{ож}} > 0,5$   
            то  $N_{\text{ож}} := N_{\text{ож}} + 1$   
    все  
     $K_{\text{н}}[i] := K_{\text{ст}}[i] + 10 * (N[i] - N_{\text{ож}})$   
кц  
кон

## ПОСЛЕСЛОВИЕ К ГЛАВЕ

В задачах «Шестизначное число», «Совместная расшифровка чисел», «Эксперименты с двузначными числами» человек вначале уменьшает количество возможных вариантов (ограничивает дерево целей), предоставляя ПМК уточнять решение в доступной ему ограниченной области. Партнеры, объединенные единой целью, как бы включаются в процесс последовательно: человек начинает, ПМК завершает. При этом за человеком остается интеллектуальная работа, требующая догадки. На долю ПМК выпадает рутинная часть — вычислительные процедуры. Избавле-

ние от утомительных вычислений позволяет человеку сконцентрировать усилия на творческой части задачи.

В задачах «ПМК отгадывает цифры», «Быки и коровы», «Логика» и их модификациях ходы участников чередуются: человек каждый раз информирует ПМК об эффективности его действий либо сам получает от него такие сведения. При этом роль партнера, считающего информацию, скрытую от другого, пассивна.

Чтобы раскрыть познавательные возможности каждого участника диалога, мы передаем роль «отгадчика» то человеку, то ПМК, сравнивая их возможности.

Подводя итоги, можно сказать, что ПМК с трудом справляется с задачей, его действия достаточно примитивны (см. «Быки и коровы», «Логика» в варианте, когда машина отгадывает и др.). Эксперименты показывают, что человек после нескольких партий теряет интерес, легко находит контригру, быстро разгадав алгоритм партнера. Напротив, в роли «загадчика» игра ПМК вызывает интерес.

В задаче «Ловля блохи» ПМК, вооруженный простым алгоритмом, «преследует» противника, делающего беспорядочные прыжки. Игра со случаем имеет важное значение для практической деятельности людей.

Вспомним, что в задаче «Бочонок» (см. гл. 1) роль «случая» принадлежит человеку, задающему начальное расположение булылок и количество поворотов бочонка. Наконец, в задаче «Рейтинг» действия человека и машины согласованы в процессе активного решения задачи.

В последующих главах будут рассмотрены другие формы диалога человек — ПМК.

— Мудрая женщина, не считаешь ли ты, что дело владыки ловить блох, отгадывать цифры, побеждать быков и другую скотину? — спросил царь. Еще одна такая победа, и я останусь без войска. Не помню уже, когда воевал.

— О великий из великих! — молвила Шахразада. — Поверженный сосед еще может подняться, противник же, ставший другом, — твой на всю жизнь. Впереди нас ждет интереснейшая игра «Магические квадраты». Мы испытаем с тобой искусственное разумное существо.

— Солдата? — спросил царь.  
И тут наступило утро.

## Глава 3

### ПРОДОЛЖЕНИЕ ДИАЛОГА

#### 3.1. МАГИЧЕСКИЕ КВАДРАТЫ

На гравюре выдающегося математика Альберта Дюрера «Меланхolia» сидит женщина с задумчивым, отрешенным видом, окруженнная геометрическими телами и физическими приборами. Над ее головой — магический квадрат, внизу — год создания гравюры — 1514:

|    |    |    |    |
|----|----|----|----|
| 16 | 3  | 2  | 13 |
| 5  | 10 | 11 | 8  |
| 9  | 6  | 7  | 12 |
| 4  | 15 | 14 | 1  |

Квадрат из  $n \times n$  клеток называется магическим, если сумма натуральных чисел по всем горизонталям, вертикалям и обеим диагоналям есть величина постоянная (константа магического квадрата). Доказано, что для нечетного  $n$  можно построить магический квадрат, причем константа будет равна  $0,5n(n^2+1)$ .

Начнем, как обычно, с алгоритма. Ходом будем считать заполнение свободной клетки квадрата некоторым числом. Клетка считается заблокированной, если ближняя к ней клетка по диагонали в направлении справа вверх уже занята числом.

1. Проставить в верхней строке среднего столбца 1.
2. К предыдущему числу прибавить 1, сумма фиксируется (запоминается).

3. Если использованная в последнем ходе клетка заблокирована либо занимает правый верхний угол квадрата, то в очередном ходе фиксированное число вписать в клетку, находящуюся под данной в том же столбце (она свободна). Возврат к указанию 2.

4. Если клетка, использованная в последнем ходе, в верхней строке квадрата, то в очередном ходе заполнить клетку первой строки соседнего справа столбца. Возврат к указанию 2.

5. Если занятая в последнем ходе клетка в правом столбце, то в очередном ходе заполнить клетку первого столбца ближеи сверху строки. Возврат к указанию 2.

6. Занять фиксированным числом клетку, ближнюю к данной по диагонали в направлении справа вверх. Возврат к указанию 2.

Пример. Построение магического квадрата  $n=3$ .

| $n$ | 1 | 2 | 3 |
|-----|---|---|---|
| 3   | 8 | 1 | 6 |
| 2   | 3 | 5 | 7 |
| 1   | 4 | 9 | 2 |

1: проставляется 1. 2: запоминается 2. 3: условие не выполнено. 4: условие выполнено, делается ход 2. 2: запоминается 3. 3: условие не выполнено, не выполнено условие указания 4. Согласно указанию 5 делается ход 3. 2: запоминается 4. В соответствии с указанием 3 — клетка, занятая тройкой, заблокирована единицей — делается ход 4 и т. д.

Особенно нагляден алгоритм на УАЯ, в котором ходы записываются с помощью координат клеток — номеров столбцов и строк:

```

алг МАГИЧЕСКИЙ КВАДРАТ (нат  $n$ , nat таб Т [1 :  $n$ , 1 :  $n$ ])
арг  $n$ 
рез Т
нач nat  $k$ ,  $i$ ,  $j$ ,  $m$ 
 $i := (n+1)/2$ ;  $j := n$ ; Т [ $i$ ,  $j$ ] := 1
для  $k$  от 2 до  $n**2$ 
иц
    ввод  $m$ 
        выбор
            при  $m = 3$ :  $j := j - 1$ 
            при  $m = 4$ :  $i := i + 1$ ;  $j := 1$ 
            при  $m = 5$ :  $i := 1$ ;  $j := j + 1$ 
            иначе  $i := i + 1$ ;  $j := j + 1$ 
    все
    Т [ $i$ ,  $j$ ] :=  $k$ 
кц
кон

```

Примечание.  $m$  — номер указания алгоритма, условие которого выполнено для данного значения  $k$ , перед каждым ходом вводится в алгоритм.

Функции между ПМК и человеком в диалоге распределяются следующим образом. ПМК формирует очередное число, вычисляет координаты клетки (столбец и строку), в которую оно проставляется. Человек после каждой индикации записывает полученное ПМК число в указанную клетку, сигнализирует о блокировании занятой в очередном ходе клетки или о правом верхнем угле, вводя нуль, если условие выполнено, или другое число в противном случае. Нажимает клавиши В/О С/П для запуска счета, а для продолжения вычислений после индикации клавишу С/П. За человеком остается также решение вопроса об окончании игры. Информацию он получает визуально, вводит, нажимая клавиши. Выбор клетки для выполнения хода связан с определенными размышлениями. Здесь человека успешно заменяет ПМК. Напротив, алгоритмическое описание «блока» и «правого верхнего угла» связано со специальными вычислительными процедурами для ПМК, с ними лучше справляется человек. Так в решении задачи человек и ПМК образуют союз по дополнению.

#### Программа МАГИЧЕСКИЙ КВАДРАТ

Распределение памяти  
Р4, Р5 — абсцисса (столбец) и ордината (строка) клетки, заполненной ПМК в очередном ходе;

|    |                                                               |    |    |       |    |    |       |    |
|----|---------------------------------------------------------------|----|----|-------|----|----|-------|----|
| P6 | — фиксированное число, проставляемое в клетке в данном ходе;  |    |    |       |    |    |       |    |
| P7 | — константа $n$ — число клеток в столбцах и строках квадрата. |    |    |       |    |    |       |    |
| 00 | X→П7                                                          | 47 | 18 | KП→X6 | Г6 | 35 | FX=0  | 5E |
| 01 | X→П5                                                          | 45 | 19 | ↔     | 14 | 36 | 42    | 42 |
| 02 | 1                                                             | 01 | 20 | FX≠0  | 57 | 37 | 1     | 01 |
| 03 | X→П6                                                          | 46 | 21 | 46    | 46 | 38 | X→П4  | 44 |
| 04 | +                                                             | 10 | 22 | П→X7  | 67 | 39 | KП→X5 | Г5 |
| 05 | 2                                                             | 02 | 23 | П→X5  | 65 | 40 | БП    | 51 |
| 06 | :                                                             | 13 | 24 | —     | 11 | 41 | 08    | 08 |
| 07 | X→П4                                                          | 44 | 25 | FX=0  | 5E | 42 | KП→X4 | Г4 |
| 08 | П→X6                                                          | 66 | 26 | 32    | 32 | 43 | KП→X5 | Г5 |
| 09 | ВП                                                            | 0C | 27 | 1     | 01 | 44 | БП    | 51 |
| 10 | 2                                                             | 02 | 28 | X→П5  | 45 | 45 | 08    | 08 |
| 11 | П→X4                                                          | 64 | 29 | KП→X4 | Г4 | 46 | П→X5  | 65 |
| 12 | ВП                                                            | 0C | 30 | БП    | 51 | 47 | 1     | 01 |
| 13 | 1                                                             | 01 | 31 | 08    | 08 | 48 | —     | 11 |
| 14 | +                                                             | 10 | 32 | П→X7  | 67 | 49 | X→П5  | 45 |
| 15 | П→X5                                                          | 65 | 33 | П→X4  | 64 | 50 | БП    | 51 |
| 16 | +                                                             | 10 | 34 | —     | 11 | 51 | 08    | 08 |
| 17 | C/P                                                           | 50 |    |       |    |    |       |    |

**Инструкция.** Начальный ввод:  $n$  В/О С/П, где  $n$  — число клеток в столбце или строке квадрата. После появления на индикаторе четырех- или трехзначного числа (см. индикацию ходов ПМК) запишите результаты в клетки квадрата. Нажмите 0 С/П, если вы видите, что клетка заблокирована или примыкает к правому верхнему углу. Во всех остальных случаях достаточно нажать С/П. Решение заканчивается, когда заполнены все клетки квадрата.

Ходы ПМК индицируются в виде четырехзначных чисел. Число, образованное первыми двумя цифрами, проставляется в клетке, остальные две цифры — абсцисса (столбец) и ордината (строка) занимаемой клетки. Если индицируемое число трехзначно, то первая цифра 0, и в клетку записывается однозначное число.

**Контрольный пример.**  $n=3$ . 3 В/О С/П (5 с) «123» С/П (9 с) «231» С/П (9 с) «312» 0 С/П (6 с) «411» — блок, С/П (9 с) «522» С/П (9 с) «633» 0 С/П (6 с) «732» — угол и т. д.

### Структура программы

- 00—07 : ввод данных для построения первого хода в регистры памяти;
- 08—17 : формирование и индикация очередного хода;
- 18, 27—31 : вычисление компоненты для хода из клеток верхней строки, за исключением правой угловой;
- 19—26 : проверка — блок или правая верхняя клетка;
- 18, 37—41 : вычисление компонентов для хода после заполнения клетки правого столбца, за исключением верхней угловой;
- 42—51 : вычисление компонентов для ходов в остальных случаях.

### ЗАДАНИЯ

1. Предоставьте ПМК решение вопроса об окончании игры. Воспользуйтесь тем, что игра заканчивается индикацией числа  $n^2$ . После каждой индикации сравните полученное число с  $n^2$ . Если они равны, то индикация π означает

окончание. В противном случае процесс построения магического квадрата продолжается. Внесите в программу соответствующие изменения.

2. Исключение человека из контура решения задачи. Требуется передать автомата непродуктивную работу сложения за ситуациями «блок» и «правый верхний угол». Точным признаком наличия этих ситуаций является кратность числа, проставленного в данной клетке, числу  $n$ . Внесите соответствующие изменения в программу.

**З а м е ч а н и е.** Выполнив задания 1, 2, мы фактически исключаем человека из контура решения задачи, не считая, разумеется, нажатия клавиши и записи данных в клетки квадрата. Обидно, что малая емкость памяти ПМК не дает возможности освободить человека от этой не требующей «ума» работы.

3. Оптимизация программы. В программе независимо рассмотрены случаи, встречающиеся при решении задачи: клетка расположена в верхней строке квадрата, в правом столбце. В то же время можно объединить все случаи общим аналитическим условием и построить на этой основе единый алгоритм.

Первый ход: число 1, координаты  $((n+1)/2; n)$ . Далее проставляемое число и обе его координаты увеличиваются на 1. Если какая-либо координата выходит за пределы квадрата, то ее «возвращают», вычитая  $n$ . В процессе решения задачи это повторяется  $n$  раз. Координаты  $(n+1)$ -го числа получаются так: абсцисса  $n$ -го числа сохраняется, ордината уменьшается на единицу. Затем предыдущая процедура повторяется для очередной  $n$ -ки и т. д.

Составьте основанный на этом алгоритме программу для ПМК.

**З а м е ч а н и е.** Алгоритм, использованный в предыдущей программе, ориентирован на то, что с полем игры работает человек, который без дополнительных указаний распознает верхнюю строку, правый столбец и т. д. Эти топологические характеристики не требуют специальных вычислений. ПМК «не видит», но может вычислять. Предложенный алгоритм имеет чисто вычислительный характер и поэтому приспособлен к возможностям автомата.

### Программа для задания 3

| <i>n</i> | В/О   | С/П |    |                 |    |    |       |    |
|----------|-------|-----|----|-----------------|----|----|-------|----|
| 00       | X→П7  | 47  | 20 | П→X6            | 66 | 39 | KП→X5 | Г5 |
| 01       | X→П0  | 40  | 21 | П→X7            | 67 | 40 | П→X7  | 67 |
| 02       | X→П5  | 45  | 22 | FX <sup>a</sup> | 22 | 41 | П→X4  | 64 |
| 03       | 1     | 01  | 23 | —               | 11 | 42 | —     | 11 |
| 04       | X→П6  | 46  | 24 | FX=0            | 5E | 43 | FX<0  | 5C |
| 05       | +     | 10  | 25 | 28              | 28 | 44 | 49    | 49 |
| 06       | 2     | 02  | 26 | F π             | 20 | 45 | /—/   | 0L |
| 07       | :     | 13  | 27 | C/P             | 50 | 46 | X→П4  | 44 |
| 08       | X→П4  | 44  | 28 | FL0             | 5Г | 47 | БП    | 51 |
| 09       | П→X6  | 66  | 29 | 38              | 38 | 48 | 09    | 09 |
| 10       | ВП    | 0C  | 30 | П→X5            | 65 | 49 | П→X7  | 67 |
| 11       | 02    | 02  | 31 | 1               | 01 | 50 | П→X5  | 65 |
| 12       | П→X4  | 64  | 32 | —               | 11 | 51 | —     | 11 |
| 13       | ВП    | 0C  | 33 | X→П5            | 45 | 52 | FX<0  | 5C |
| 14       | 1     | 01  | 34 | П→X7            | 67 | 53 | 56    | 56 |
| 15       | +     | 10  | 35 | X→П0            | 40 | 54 | /—/   | 0L |
| 16       | П→X5  | 65  | 36 | БП              | 51 | 55 | X→П5  | 45 |
| 17       | +     | 10  | 37 | 09              | 09 | 56 | БП    | 51 |
| 18       | C/P   | 50  | 38 | KП→X4           | Г4 | 57 | 09    | 09 |
| 19       | KП→X6 | Г6  |    |                 |    |    |       |    |

**Контрольный пример.** 5 В/О С/П (5 с) «135» С/П (12 с) «241» С/П «352» С/П «413» С/П «524» С/П (7 с) «623» С/П (12 с) «734» и т. д. Индикация π означает, что задача решена, последний ход вследствие его очевидности не индицируется.

В алгоритме на УАЯ предусмотрен вывод всей таблицы без индикации отдельных ходов:

```

Адрес алг МАГИЧЕСКИЕ КВАДРАТЫ (нат n, nat таб МАК [1:n, 1:n])
команды
    арг n
    рез МАК
нач nat j, k, i
j := (n+1)/2; k := n; МАК[j, k] := 1
для i от 2 до n2
нц
    если i = [i/n]*n = 1
        то k := k - 1
        иначе j := j + 1; k := k + 1
        выбор
            при j > n: j := j - n
            при k > n: k := k - n
        все
    МАК[j, k] := i
кц
кон

```

## 3.2. КРЕСТИКИ-НОЛИКИ

Играют человек и микрокалькулятор. Клетки пронумерованы «по улитке», т. е. по часовой стрелке, как показано в таблице:

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 8 | 9 | 4 |
| 7 | 6 | 5 |

Ходы ПМК обозначим крестиками, человека — ноликами. Сделать ход — значит указать номер занимаемой клетки. Ходы человека вводятся набором номера на клавиатуре, ходы ПМК выводятся на индикатор. ПМК начинает и выигрывает либо, если противник не допустил ошибок, сводит игру вничью. Человек записывает ходы игроков на игровом поле. Предлагаем следующий алгоритм.

1. ПМК занимает центральную позицию — высвечивается 9. Вводится ход человека  $l_1$ .

2. Из номера клетки, занятой противником в очередном ходе, ПМК вычитает 1. Если разность равна нулю, то она заменяется 8. Результат выдается на индикатор в качестве очередного хода ПМК: второго, третьего и т. д. Вводится ответный ход человека.

3. Если номер поля, занятого человеком, не симметричен относительно центра предшествующему полю, занятому ПМК, то человек проиграл, о чем ПМК сообщает, индицируя ЕГГОГ.

4. Если человек сделал четвертый ход, то появляется сообщение л — игра закончилась вничью.

5. Проверяется четность номера клетки, занятой человеком первым ходом (рис. 11).

6. Если  $l_1$  четно, то ход ПМК вычисляется как  $l_1 - 2$ . Если разность равна 0, она заменяется на 8. Так формируется третий ход ПМК.

Независимо от ответного третьего хода человека появляется сообщение ЕГГОГ: машина перекрыла сразу две линии и выиграла на третьем ходе.

7. Если  $l_1$  нечетно, возвратиться к указанию 2.

Заняв первым ходом центральную клетку, ПМК в дальнейшем каждым своим ходом заставляет противника выполнять симметричные относительно центра ходы, лишая его возможности создавать угрозы себе. ПМК настроен на выигрыш, как бы не заботясь о своем проигрыше.

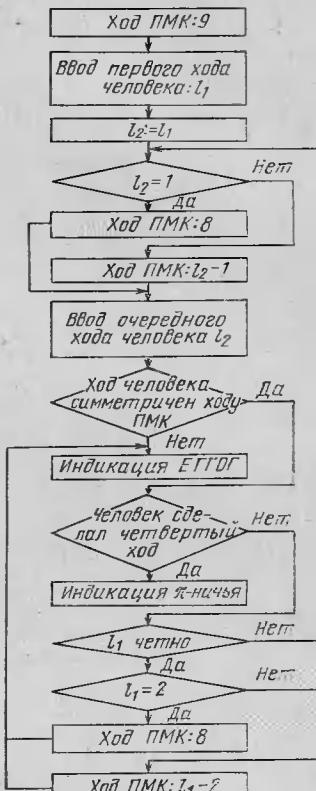


Рис. 11

## Программа КРЕСТИКИ-НОЛИКИ

### Распределение памяти

P1 — номер клетки, занятой человеком в первом ходе;

P2 — номера клеток в последующих ходах человека, вначале содержимое P2 равно содержимому P1.

|    |       |    |    |       |    |    |                 |    |
|----|-------|----|----|-------|----|----|-----------------|----|
| 00 | СХ    | 0Г | 14 | П→Х1  | 61 | 28 | FX=0            | 5E |
| 01 | 9     | 09 | 15 | 2     | 02 | 29 | 31              | 31 |
| 02 | C/P   | 50 | 16 | —     | 11 | 30 | 8               | 08 |
| 03 | X→П1  | 41 | 17 | ПП    | 53 | 31 | C/P             | 50 |
| 04 | X→П2  | 42 | 18 | 28    | 28 | 32 | X→П2            | 42 |
| 05 | ПП    | 53 | 19 | К—    | 27 | 33 | —               | 11 |
| 06 | 26    | 26 | 20 | ПП    | 53 | 34 | FX <sup>a</sup> | 22 |
| 07 | П→Х1  | 61 | 21 | 26    | 26 | 35 | FV              | 21 |
| 08 | 9     | 09 | 22 | ПП    | 53 | 36 | 4               | 04 |
| 09 | 0     | 00 | 23 | 26    | 26 | 37 | —               | 11 |
| 10 | X     | 12 | 24 | F π   | 20 | 38 | FX=0            | 5E |
| 11 | F sin | 1C | 25 | С/П   | 50 | 39 | 19              | 19 |
| 12 | FX=0  | 5E | 26 | КП→Х2 | Г2 | 40 | B/O             | 52 |
| 13 | 20    | 20 | 27 | П→Х2  | 62 |    |                 |    |

**Инструкция.** Переключатель углов в положении Г. Ввод: В/О С/П, индицируется 9 — номер клетки, занятой ПМК в первом ходе. После каждого вывода ПМК своего хода наберите номер клетки, которую вы занимаете ответным ходом, и, нажав С/П, запустите счет. Через 5—7 с на индикаторе появляется следующий ход ПМК и т. д. Появление на индикаторе ЕГГОГ означает, что вы проиграли — либо оказались в ситуации, когда ПМК перекрыл две линии, либо упустили возможность занять клетку, симметричную относительно центра той клетки, что занял ПМК. Появление π означает ничейный исход игры.

**Контрольные игры.** 1. В/О С/П (2 с) «9».

| Номер клетки, занятой человеком | Время вычисления хода ПМК, с | Ход ПМК   |
|---------------------------------|------------------------------|-----------|
| 7 С/П                           | 3                            | 6         |
| 2 С/П                           | 7                            | 1         |
| 5 С/П                           | 5                            | 4         |
| 8 С/П                           | 3                            | π — ничья |

2. В/О С/П (2с) «9».

| Номер клетки, занятой человеком | Время вычисления хода ПМК, с | Ход ПМК             |
|---------------------------------|------------------------------|---------------------|
| 4 С/П                           | 3                            | 3                   |
| 7 С/П                           | 7                            | 2                   |
| 1 С/П                           | 7                            | ЕГГОГ — ПМК выиграл |

### Структура программы

- 00—04 : генерация первого хода ПМК, ввод первого хода противника, запоминается в Р1 и Р2.
- 05, 06 : обращение к подпрограмме формирования второго хода ПМК.
- 07—13 : проверка на четность номера клетки, занятой человеком в первом ходе.
- 14—19 : если номер клетки четный, ПМК объявляет о выигрыше.
- 20—25 : если номер клетки нечетный, ПМК на четвертом ходе объявляет ничью, индицируя π.
- 26—31 : генерация очередного хода ПМК; ввод ответного хода человека (подпрограмма).
- 32—38 : проверка симметричности клетки, занятой человеком, клетке, занятой перед этим ПМК (подпрограмма). Если клетки симметричны, игра продолжается (40).
- 39 : если клетки несимметричны, переход к индикации проигрыша человека.

```

Адреса команд для одного цикла алгоритма
01, 02
03, 04
05, 06, 26
27—31
32—39
07—13
14—18
28—40
19
      k := l2 — 1
      если k = 0
      то k := 8
      все
      вывод k; ввод l2
      если |k — l2| ≠ 4
      то C := «Вы проиграли»; i := 3
      иначе если l1/2 = [l1/2]
      то k := l1 — 2; если k = 0
      то k := 8
      все
      C := «вы проиграли»; i := 3
      все
      кон

```

### ЗАДАНИЕ

В командах по адресу 28—30 предусмотрена замена хода ПМК  $k=0$  ходом  $k=8$ . Условимся, что при индикации 0 ПМК занимает клетку 8. Исключите из программы команды 28—30.

### 3.3. КРЕСТИКИ-НОЛИКИ, ВАРИАЦИЯ

Как и раньше, играют человек и микрокалькулятор. В отличие от известной игры, крестик и нолик не закреплены за игроками: каждый, делая ход, может выбрать любой символ. Выигрывает тот, кто зачеркивает три одинаковых символа на горизонтали, вертикали или диагонали. ПМК начинает, занимая первым ходом центральную клетку 9: ставит крестик или нолик

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 5 | 9 | 4 |
| 6 | 7 | 8 |

Если после очередного хода противника возникла выигрышная для ПМК ситуация, то ПМК информирует о своей победе крестиком или ноликом. В противном случае ПМК занимает клетку, центрально-симметричную позиции, занятой человеком, и копирует его символ. Ходы человека вводятся, а ответы ПМК индицируются двузначными числами: первая цифра — номер клетки, вторая — выбранный символ, т. е. крестик (1) или нолик (0). Если человек первым или вторым ходом занял угловую клетку,

то в худшем случае ПМК выигрывает на третьем ходу, в противном случае — на четвертом.

### Программа КРЕСТИКИ-НОЛИКИ, вариация

|    |      |    |    |       |    |    |      |    |
|----|------|----|----|-------|----|----|------|----|
| 00 | X→П0 | 40 | 24 | П→ХД  | 6Г | 48 | П→ХД | 6Г |
| 01 | 9    | 09 | 25 | С/П   | 50 | 49 | —    | 11 |
| 02 | 0    | 00 | 26 | 1     | 01 | 50 | FX=0 | 5Е |
| 03 | +    | 10 | 27 | 0     | 00 | 51 | 54   | 54 |
| 04 | С/П  | 50 | 28 | :     | 13 | 52 | П→ХД | 6Г |
| 05 | КППВ | —L | 29 | X→ПС  | 4C | 53 | С/П  | 50 |
| 06 | ПП   | 53 | 30 | КП→ХС | ГС | 54 | В/О  | 52 |
| 07 | 55   | 55 | 31 | F0    | 25 | 55 | П→ХС | 6C |
| 08 | FX=0 | 5E | 32 | П→ХС  | 6C | 56 | —    | 01 |
| 09 | 14   | 14 | 33 | —     | 11 | 57 | —    | 11 |
| 10 | П→ХА | 6— | 34 | 1     | 01 | 58 | П→ХС | 6C |
| 11 | С/П  | 50 | 35 | 0     | 00 | 59 | 3    | 03 |
| 12 | БП   | 51 | 36 | ×     | 12 | 60 | —    | 11 |
| 13 | 23   | 23 | 37 | X→ПД  | 4Г | 61 | ×    | 12 |
| 14 | П→ХА | 6— | 38 | 9     | 09 | 62 | П→ХС | 6C |
| 15 | С/П  | 50 | 39 | П→ХС  | 6C | 63 | 6    | 06 |
| 16 | КППВ | —L | 40 | —     | 11 | 64 | —    | 11 |
| 17 | ПП   | 53 | 41 | 1     | 01 | 65 | ×    | 12 |
| 18 | 55   | 55 | 42 | 0     | 00 | 66 | П→ХС | 6C |
| 19 | FX≠0 | 57 | 43 | ×     | 12 | 67 | 8    | 08 |
| 20 | 24   | 24 | 44 | П→ХД  | 6Г | 68 | —    | 11 |
| 21 | П→ХА | 6— | 45 | +     | 10 | 69 | ×    | 12 |
| 22 | С/П  | 50 | 46 | X→ПА  | 4— | 70 | В/О  | 52 |
| 23 | КППВ | —L | 47 | П→Х0  | 60 |    |      |    |

**Инструкция.** Перед началом первой игры набрать 26 X→ПВ, в последующих играх можно этого не делать. Выбираем произвольный символ: крестик или нолик и вводим его код, 1 или 0. Запускаем счет, нажимая клавиши В/О С/П. Через несколько секунд на индикаторе двузначное число, начинающееся с 9, — первый ход ПМК. В ответ вводим также двузначное число. Каждый свой ход завершаем нажатием клавиши С/П. Появляющиеся на индикаторе 0 или 1 означают победу ПМК.

**Контрольный пример.** 26 X→ПВ 1 В/О С/П (12 с) «91», 70 С/П (7 с) «20», 40 С/П (17 с) «50», 60 С/П (16 с) «0» — ПМК выиграл. См. таблицу слева, где индексы помогают восстановить последовательность ходов:

|                |                |                |  |   |   |   |
|----------------|----------------|----------------|--|---|---|---|
|                | M <sub>2</sub> |                |  | 0 |   |   |
| M <sub>3</sub> | M <sub>1</sub> | Ч <sub>2</sub> |  | 0 | 1 | 0 |
| Ч <sub>3</sub> | Ч <sub>1</sub> | M <sub>4</sub> |  | 0 | 0 | 0 |

После того, как ПМК установил в центре квадрата 1 (крестик), человек записал в клетку 7 нолик, в ответ ПМК ставит нолик в клетку 2 и т. д.

Поясним назначение отдельных фрагментов программы. Для этого воспользуемся обозначениями:  $l$ ,  $m$  — ходы, введенные человеком или индицируемые ПМК;  $N_l$ ,  $N_m$  — номера клеток, занятых в данном ходе, первые цифры  $l$  и  $m$ ;  $K_l$ ,  $K_m$  — коды ходов, вторые цифры  $l$  и  $m$  ( $K_l, K_m = 0, 1$ ):  $K_{m1}$  — начальный код ПМК.

### Структура программы

00—04 : индикация первого значения  $m$  и после останова ввод ответного  $l$ .  
 05—11 : индикация второго значения  $m$ , ввод ответного  $l$ .  
 12, 13, 23—25 : ПМК выиграл на третьем ходе.  
 14, 15 : индикация значения  $m$ , если ПМК не выиграл.  
 16—22 : индикация третьего значения  $m$ , ввод третьего значения  $l$ .  
 23—25 — ПМК выиграл на четвертом ходе.  
 26—54 : подпрограмма, обращение к которой выполняется командами КППВ. Вычисление с помощью  $l$  двух цифр: первая — номер занятой клетки, хранится в РС (команды 26—30), вторая — код, хранится в РД (команды 31—37). Вычисление номеров клетки  $N_m$ , симметричной относительно центра (команды 38—40), построение ответного значения  $m$  в РА (команды 41—46), проверка равенства  $K_l = K_{m1}$  (команды 47—50); в случае его выполнения из РД выводится сигнал о выигрыше ПМК (команды 52, 53), в противном случае выход из подпрограммы (команда 54).  
 55—70 : подпрограмма проверки, является ли угловой клетка, занятая последним ходом человека, обращение к подпрограмме производится командами ПП 55.

Адрес команды алг КРЕСТИКИ-НОЛИКИ (лит А)  
 рез А  
 нач nat  $i$ ,  $l$ ,  $m$ ,  $N_l$ ,  $N_m$  цел  $K_l$ ,  $K_m$ ,  $n$   
 $n := 0$ ; ввод  $K_m$ ;  $K_{m1} := K_m$ ;  $m := 90 + K_m$ ; выв  $m$   
 для  $i$  от 2 до 4  
 нц  
 05, ввод  $l$ ;  $N_l := [l/10]$ ;  $K_l := l - N_l * 10$ ;  $N_m := 9 - N_l$ ;  $K_m := K_l$   
 26—46  $m := N_m * 10 + K_m$   
 06, 07, если  $(N_l - 1)*(N_l - 3)*(N_l - 6)*(N_l - 8) = 0$   
 55—70 то  $n := 1$   
 все  
 выбор  
 47—50 при  $i = 2$ : если  $K_l = K_{m1}$   
 52, 53 то выв  $K_m$ ;  $i := 4$   
 10, 11 иначе выв  $m$   
 все  
 52, 53 при  $i = 3$ : если  $(K_l = K_{m1}) \vee (n = 1)$   
 10, 11 то выв  $K_m$ ;  $i := 4$   
 иначе выв  $m$   
 все  
 52, 53 иначе выв  $K_m$   
 все  
 кон  
 A := «Вы проиграли»

**Примечание.** Команды ввод и выв необходимы для организации игры в диалоговом режиме.

## ЗАДАНИЯ

1. Перестройте программу для ПМК, используя нумерацию игрового поля из § 3.2.

**Указание.** Проверка использования человеком угловой клетки упрощается, для этого достаточно нечетность  $N_i$ , однако несколько усложняется процедура вычисления центрально-симметричных клеток. Можно воспользоваться следующим алгоритмом: для клеток 1—4 симметричные получаются прибавлением 4, для 5—8 — вычитанием 4.

2. Пусть человек вместе с ходом вводит также признак использования угловой клетки, например 1 — угол, 2 — не угол. В этом случае отпадает необходимость в командах 55—70. Внесите изменения в программу.

3. «Мини-шашки». На доске  $4 \times 4$  играют в шашки-поддавки человек и ПМК. Выигрывает, как обычно, тот, у кого на доске не осталось фигур. Первый ход за человеком. Сформулируйте алгоритм и составьте программу для ПМК. Начальная ситуация приведена на рис. 12.

**Указание.** Клетки на этот раз удобнее задавать номерами столбцов и строк. Например, клетка (2; 4) находится на пересечении второго столбца и четвертой строки (отмечена на рисунке крестиком).

4. Игра Хейна\*. Игровым полем является квадратная доска, разделенная на одинаковые клетки. На каждой клетке стоит по шашке. Играют двое, делая ходы поочередно. Сделать ход — значит взять любое число шашек из одного вертикального или горизонтального ряда. Брать шашки можно только подряд, не перепрыгивая через пустые клетки. Если между шашками в строке или столбце есть пробел, играющий не имеет права взять все оставшиеся шашки за один ход. Побеждает тот из соперников, кто взял последнюю шашку.

Если доска имеет нечетное число клеток, то выигрывает начинаящий игру. Для этого он первым ходом берет центральную шашку, а каждым следующим — столько шашек, сколько взял соперник, снимая их с клеток, симметричных относительно центра. Если доска имеет четное число клеток, то первый ход следует уступить сопернику. Далее игра продолжается, как описано выше.

Составьте программу беспроигрышной игры для ПМК, ограничиваясь игровым полем  $8 \times 8$ .

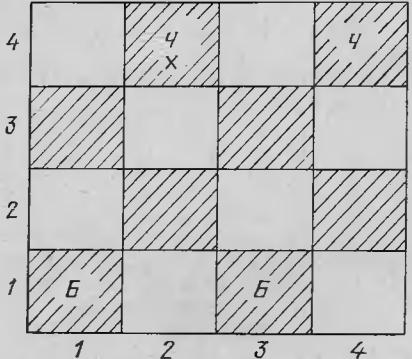


Рис. 12

\* Владыкина Л. И., Гнатенко Т. П., Касаткин В. Н. Из опыта использования игр при ознакомлении учащихся с основами теории алгоритмов и программирования//Логические и игровые задачи на ЭВМ. — Курск, 1976. — С. 219.

## 3.4. ТРИГЕКС

Игровое поле изображено на рис. 13,а. В игре участвуют человек и ПМК. Противники ходят поочередно. Сделать ход — значит «вписать» в занумерованную точку игрового поля код: для ПМК — крестик, для человека — нолик. Побеждает тот, кто первым займет три позиции вдоль одного из девяти возможных прямых направлений. ПМК отдано преимущество первого хода.

Для удобства описания рассмотрим два треугольника: внешний 1—9—4—3—7—6 и внутренний 2—5—8. Ходы ПМК и человека обозначены  $p_i$  и  $q_i$ , где  $i$  — номер хода. Схема алгоритма игры для ПМК приведена на рис. 14.

| Программа ТРИГЕКС |        |    |    |        |    |    |        |    |
|-------------------|--------|----|----|--------|----|----|--------|----|
| 00                | 1      | 01 | 28 | ×      | 12 | 56 | C/П    | 50 |
| 01                | 8      | 08 | 29 | F cos  | 1Г | 57 | 5      | 05 |
| 02                | 0      | 00 | 30 | ×      | 12 | 58 | 6      | 06 |
| 03                | X→ПА   | 4— | 31 | 3      | 03 | 59 | Fπ     | 20 |
| 04                | 2      | 02 | 32 | ×      | 12 | 60 | C/П    | 50 |
| 05                | C/П    | 50 | 33 | +      | 10 | 61 | P→XB   | 6L |
| 06                | X→ПВ   | 4L | 34 | Fπ     | 20 | 62 | 4      | 04 |
| 07                | 5      | 05 | 35 | C/П    | 50 | 63 | —      | 11 |
| 08                | —      | 11 | 36 | P→XB   | 6L | 64 | FX = 0 | 5E |
| 09                | FX = 0 | 5E | 37 | 1      | 01 | 65 | 70     | 70 |
| 10                | 16     | 16 | 38 | —      | 11 | 66 | 9      | 09 |
| 11                | 8      | 08 | 39 | P→XB   | 6L | 67 | 5      | 05 |
| 12                | 7      | 07 | 40 | 6      | 06 | 68 | Fπ     | 20 |
| 13                | C/П    | 50 | 41 | —      | 11 | 69 | C/П    | 50 |
| 14                | БП     | 51 | 42 | X→ПС   | 4C | 70 | 3      | 03 |
| 15                | 23     | 23 | 43 | ×      | 12 | 71 | —      | 11 |
| 16                | 3      | 03 | 44 | P→XB   | 6L | 72 | FX = 0 | 5E |
| 17                | —      | 11 | 45 | 9      | 09 | 73 | 78     | 78 |
| 18                | FX = 0 | 5E | 46 | —      | 11 | 74 | 8      | 08 |
| 19                | 36     | 36 | 47 | ×      | 12 | 75 | 5      | 05 |
| 20                | 5      | 05 | 48 | FX = 0 | 5E | 76 | Fπ     | 20 |
| 21                | 1      | 01 | 49 | 61     | 61 | 77 | C/П    | 50 |
| 22                | C/П    | 50 | 50 | P→XC   | 6C | 78 | 8      | 08 |
| 23                | P→XA   | 6— | 51 | FX = 0 | 5E | 79 | 6      | 06 |
| 24                | ×      | 12 | 52 | 57     | 57 | 80 | Fπ     | 20 |
| 25                | F cos  | 1Г | 53 | 5      | 05 | 81 | C/П    | 50 |
| 26                | P→XA   | 6— | 54 | 9      | 09 |    |        |    |
| 27                | P→XB   | 6L | 55 | Fπ     | 20 |    |        |    |

**Инструкция СХ В/О С/П.** Переключатель углов в положении Г. ПМК начинает. Игра завершается в четыре хода, если человек в ответ на первый ход ПМК занял позицию на любой из сторон внутреннего треугольника, и в три хода, если на стороне внешнего треугольника. После запуска индицируется 2 — номер вершины, которую занимает ПМК. Вторая индикация — это или двузначное число, составленное из номеров позиций, занятых ПМК во втором и третьем ходах, или π, что означает победу ПМК, поставившего противника в положение «двойного боя», т. е. создав возможность перекрытия сразу двух линий. В первом случае игра продолжается и завершается после следующего, четвертого хода ПМК. В этом случае нажатием клавиши  $\leftrightarrow$  выводим на индикатор последние ходы ПМК. После каждого останова человек вводит свой ход и нажимает клавиши  $B \uparrow$  С/П.

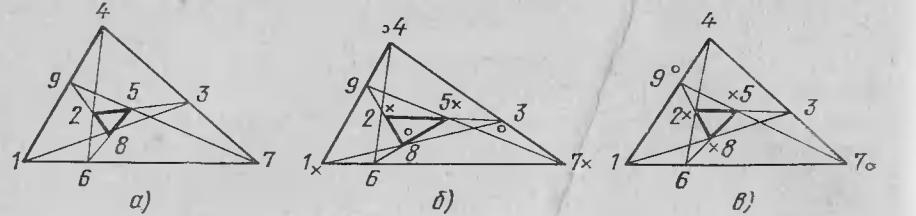


Рис. 13

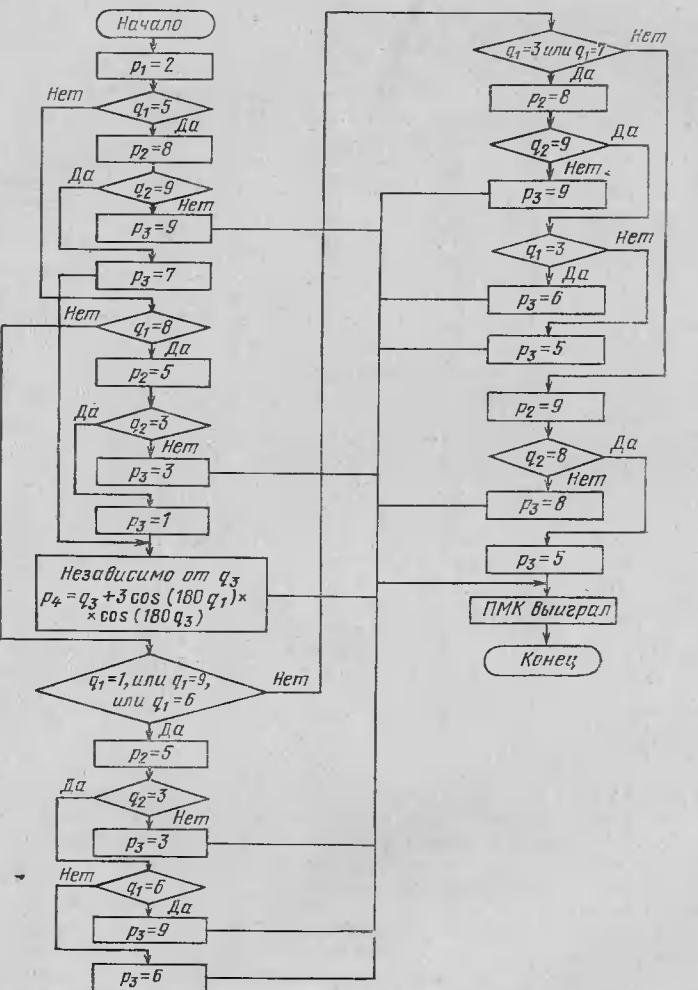


Рис. 14

Есть одна особенность в выводе двузначного числа. Между двумя ходами ПМК предполагается ход человека. Он не выводится на индикатор, так как легко угадывается по рисунку, сопровождающему игру. Это единственно возможный в сложившейся ситуации ход человека. Предполагается, таким образом, что человек не делает ошибок в игре и сдается лишь в безвыходной для него ситуации «двойного боя».

#### Контрольные примеры

1. СХ В/О С/П. Переключатель в положении Г. (2 с) «2» 8 В↑ С/П (4 с) «51». Последняя индикация расшифровывается так: ПМК занял позицию 5, человек под угрозой проигрыша вынужден ходить 3 (рис. 13, б). После этого ПМК ходит 1, «закрывая» свой проигрыш. Пусть следующий ход человека 4: 4 В↑ С/П (5 с) «п» — ПМК выиграл. Нажимаем клавишу ↔ на индикаторе 7. Этот ход ПМК исключает выигрыш человека, одновременно создает двойной бой на линиях 7—5—9 и 7—6—1. Для выигрыша ПМК потребовалось четыре хода.

2. СХ В/О С/П (2 с) «2» 7 В↑ С/П (11 с) «п» — ПМК выиграл. Нажимаем ↔ (на индикаторе 85), ПМК делает ход 8, человек защищается от поражения ходом 9 (см. рис. 13, в), ПМК ходом 5 избегает проигрыша и одновременно создает безвыходную ситуацию для человека — двойной бой по линиям 2—5—3 и 5—8—6.

#### Структура программы

- 00—03 : В РА заносится константа 180 для построения четвертого хода ПМК.
- 04, 05 : индикация первого хода ПМК : 2.
- 06 : ввод ответного хода человека, нажатием В↑ С/П включается продолжение вычислений, первый ход человека запоминается в РВ.
- 07—15 : человек занял позицию на контуре внутреннего треугольника. Если это позиция 5, то на индикатор выводятся второй и третий ходы ПМК: 8 и 7 (между ними предполагается вынужденный для ухода от проигрыша ответный ход человека).
- 16—22 : если человек занял позицию 8, то два следующих хода ПМК: 5 и 1.
- 23—35 : формирование и вывод четвертого хода ПМК.
- 36—81 : разработаны варианты, когда человек, вступив в игру, занял позицию на контуре внешнего треугольника. Точнее, анализируются возникшие ситуации (команды 36—60), когда этот ход относится к позициям 1—9—6; ответ на занятие позиции 6 (команды 53—56), 1 или 9 (команды 57—60). Если человек начинает игру ходом в позицию 4, то выполняются команды 61—69, в позицию 7 — команды 70—77, в позицию 3 — команды 78—81.

Адрес алг ТРИГЕКС (лит Т)  
команды рез Т

```
нач nat p1, p2, p3, p4, q1, q2, q3
p1 := 2; выв p; ввод q1
выбор
```

06—15, при q1=5 p2:=8; выв p2; ввод q2; если q2 ≠ 9  
23—35 to p3:=9; выв p3

16-35

```

иначе  $p_3 := 7$ ; вывод  $p$ 
ввод  $q_3$ 
 $p_4 := q_3 + 3 *$ 
 $* \cos(180 * q_1) *$ 
 $* \cos(180 * q_3)$ 
выв  $p_4$ 

все
при  $q_1 = 8$ :  $p_2 := 5$ ; вывод  $p_2$ ; ввод  $q_2$ ; если  $q_2 \neq 3$ 
то  $p_3 := 3$ ; вывод  $p_3$ 
иначе  $p_3 := 1$ ; вывод  $p_3$ 
ввод  $q_3$ 
 $p_4 := q_3 + 3 *$ 
 $* \cos(180 * q_1) *$ 
 $* \cos(180 * q_3)$ 
выв  $p_4$ 

```

36-60

```

все
при  $q_1 = 1 \vee q_1 = 9 \vee q_1 = 6$ :
 $p_2 := 5$ ; вывод  $p_2$ ; ввод  $q_2$ ; если  $q_2 \neq 3$ 
то  $p_3 := 3$ 
иначе если  $q_1 = 6$ 
то  $p_3 := 9$ 
иначе  $p_3 := 6$ 
все

```

70-81

```

при  $q_1 = 3 \vee q_1 = 7$ :  $p_2 := 8$ ;
выв  $p_2$ ; ввод  $q_2$ ; если  $q_2 \neq 9$ 
то  $p_3 := 9$ 
иначе если  $q_1 = 7$ 
то  $p_3 := 5$ 
иначе  $p_3 := 6$ 
все

```

61-69

```

иначе  $p_2 := 9$ ; вывод  $p_2$ ; ввод;  $q_2$  если  $q_2 \neq 8$ 
то  $p_3 = 8$ 
иначе  $p_3 := 5$ 
все
выв  $p_3$ 

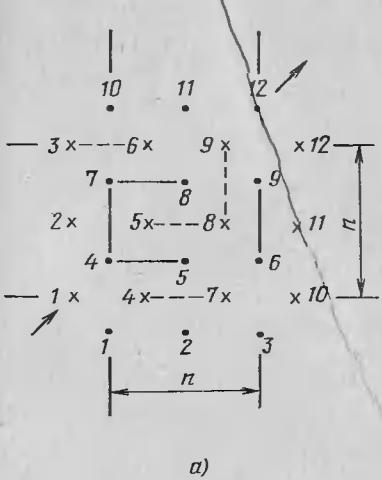
```

все  
Т: «вы проиграли»  
кон

### 3.5. БРИДЖ-ИТ

Игра придумана в начале 50-х годов нашего века американцем Гейлом. Полем для игры служат два «симметрично вложенных» друг в друга прямоугольника, например «точечного» и «крестового», размером  $n(n+1)$  (рис. 15, а, б).

ПМК и человек делают ходы поочередно. Элементы поля ПМК — точки, поля противника — крестики. Сделать ход — значит соединить отрезком прямой по горизонтали или вертикали два смежных элемента своего поля, не пересекая ранее проведенных отрезков противника. Выигрывает тот, кто первый построит ломаную, соединяющую две противоположные вершины своего прямоугольника.



a)

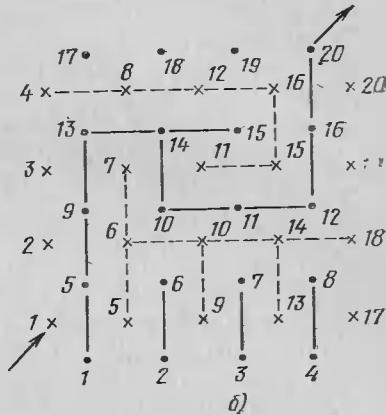


Рис. 15

Требуется сформулировать алгоритм игры для начинающего и построить на основе алгоритма программу.

**Замечание.** Игра не может закончиться вничью \*.

Точка называется *изолированной*, если ее невозможно соединить ни с одной другой точкой, например точка  $5\times$  на рис. 15, а. Расстояние между смежными точками (крестиками) по горизонтали или вертикали равно единице. Отрезок  $1\times - 12\times$  назовем *главной диагональю фигуры*. Она соединяет «напрямую» крайнюю слева точку второго ряда с противоположной вершиной прямоугольника ПМК. Главная диагональ делит фигуру на две части (плоскость — на две полуплоскости): левую верхнюю относительно диагонали и правую нижнюю.

Отрезок, проведенный ПМК, называется параллельно смещенным относительно отрезка, проведенного противником в предыдущем ходе, если он параллелен ему и смещен относительно него либо влево вверх, либо вправо вниз на  $1/2$  единицы. Например,  $6\times, 9\times$  параллельно смещены относительно  $8\times, 9\times$  вправо вниз;  $7\times, 8\times$  — относительно  $8\times, 5\times$  влево вверх.

Предлагается следующий алгоритм.

- Ход ПМК: нижняя левая вершина прямоугольника ПМК (точечного) вертикально соединяется с главной диагональю,  $1, n+1$  (см. нумерацию игровых полей).

- Ход противника ПМК.

- Если ход противника относится к крайней левой (правой) вертикали его прямоугольника, например  $10\times, 11\times$ , то ПМК от-

\* Глушанков Е., Певзнер П. Переключательная игра Шеннона//Квант. — 1980. — № 9. — С. 14—21.

вечает произвольным ходом, приближающим его к результативной вершине — вправо вверх, из любой неизолированной точки (см. рис. 15,а). Такое «продвижение по краю» для противника крайне невыгодно: ПМК, имея преимущество в один ход, беспрепятственно продвигается к своей противоположной вершине, причем может в любой момент пресечь продвижение противника. Поэтому при «разумной» игре такие ходы человек делать не станет, и они не рассматриваются в данной программе для ПМК.

4. Если человек сделал горизонтальный ход и конец отрезка принадлежит главной диагонали, то ПМК делает ход перпендикулярно отрезку, продолжающему данный, но по другую сторону диагонали. Например, на ход человека 8, 5 (точка с номером пять лежит на главной диагонали) ПМК отвечает 7, 4, т. е. перпендикуляр к отрезку 2×, 5×, который продолжает данный влево от диагонали.

5. В остальных случаях отрезок ПМК параллельно смещен относительно отрезка противника. Причем в вертикальных ходах направление смещения совпадает с названием полуплоскости, в горизонтальных — противоположно ему. Например, на ход человека 8, 9 ПМК отвечает 6, 9 (отрезок 8, 9 находится в правой нижней полуплоскости, соответственно смещение будет вправо вниз), на горизонтальный ход 3×, 6×, ответ ПМК 7, 8 (вправо вниз, т. е. противоположно названию левой верхней полуплоскости, где находится отрезок человека), на 7×, 4× ответ 5, 4 и т. д. После выполнения хода ПМК возврат к указанию 2.

Вопрос об окончании решает человек, когда позиция его становится безнадежной, поскольку ПМК, следующий алгоритму, выигрывает.

Проанализируем алгоритм с позиции возможностей ПМК. Прежде всего отметим, что он сформулирован в расчете на человека. Понятия «горизонтальный ход», «параллельное смещение» и другие, связанные со зрительным восприятием, ПМК недоступны. Для построения программы необходимо описать их в виде математических формул. Этим мы и займемся.

Возможны пять типов ходов человека: горизонтальные с концом отрезка на главной диагонали (1); горизонтальные в левой верхней полуплоскости, концы не на главной диагонали (2); горизонтальные в правой нижней полуплоскости, концы не на главной диагонали (3); вертикальные в левой полуплоскости (4); вертикальные в правой полуплоскости (5).

Распознавать ходы можно, например, так:

$$|a-b| = \begin{cases} n, & \text{если ход горизонтальный}, \\ 1, & \text{если ход вертикальный.} \end{cases}$$

Существенен порядок ввода в ПМК номеров точек, соединяемых человеком:  $a \rightarrow b$ , где  $b$  — номер точки, ближней к диагонали.

Если остаток  $r$  от деления  $b$  на  $(n+1)$  — единица, то точка с этим номером находится на главной диагонали.

Полуплоскость является верхней (левой) в двух случаях: ход человека горизонтальный и  $a < b$  или вертикальный и  $a > b$ . Полуплоскость правая (нижняя), если в горизонтальном ходе  $a > b$  или в вертикальном  $a < b$ .

Теперь можно перевести процесс распознавания типа хода человека с наглядно-образного языка на символический язык математики:

$$|a-b|=n, r=1; \quad (1)$$

$$|a-b|=n, r \neq 1, a < b; \quad (2)$$

$$|a-b|=n, r \neq 1, a > b; \quad (3)$$

$$|a-b|=1, a > b; \quad (4)$$

$$|a-b|=1, a < b. \quad (5)$$

По этим формулам вычисляется ответный ход ПМК, нумерация формул согласована с типами ходов:  $x$  и  $y$  — номера отрезка, построенного человеком:

$$x = \begin{cases} b, & \text{если } a < b \text{ (левая полуплоскость)}, \\ b-1, & \text{если } a > b \text{ (правая полуплоскость)} \end{cases}, \quad y = b+n; \quad (1)$$

$$x = b + (n-1)(r-1), \quad y = x-1; \quad (2)$$

$$x = \begin{cases} b, & \text{если } r=0, \\ b-(n-1)(n-r+1), & \text{если } r \neq 0, \end{cases}, \quad y = x+1; \quad (3)$$

$$x = \begin{cases} nb - (n^2-1)[b/n], & \text{если } a > b, \\ nb - (n^2-1)[b/n] - (n-1), & \text{если } a < b; \end{cases} \quad (4)$$

$$y = \begin{cases} x+n, & \text{если } a > b, \\ x-n, & \text{если } a < b, \end{cases} \quad (5)$$

Наиболее удобной формой описания игры является алгоритм на УАЯ.

### Программа БРИДЖ-ИТ

#### Распределение памяти

|                               |                                                                                  |                        |                                                                                                                |
|-------------------------------|----------------------------------------------------------------------------------|------------------------|----------------------------------------------------------------------------------------------------------------|
| P1<br>+P2<br>+P3<br>+P4<br>P5 | $b$<br>$n^2-1$<br>$n-1$<br>$n+1$<br>$r$<br>(остаток от деления<br>$b$ на $n+1$ ) | P6<br>+P8<br>P9<br>+PC | $r-1$<br>$n$<br>$a-b$<br>Одна из точек ( $x$ ) очередного хода, генерируемого ПМК, начальное значение<br>$n+1$ |
|-------------------------------|----------------------------------------------------------------------------------|------------------------|----------------------------------------------------------------------------------------------------------------|

|    |                 |    |    |       |    |      |          |
|----|-----------------|----|----|-------|----|------|----------|
| 1  | 01              | 33 | 36 | 36    | 66 | X→ПС | 4С       |
| 01 | C/П             | 50 | 34 | П→Х1  | 61 | 67   | 1 01     |
| 02 | П→ХС            | 6C | 35 | БП    | 51 | 68   | + 10     |
| 03 | C/П             | 50 | 36 | 39    | 39 | 69   | В/О 52   |
| 04 | X→П1            | 41 | 37 | КП→Х1 | Г1 | 70   | П→Х1 61  |
| 05 | —               | 11 | 38 | П→Х1  | 61 | 71   | П→Х8 68  |
| 06 | X→П9            | 49 | 39 | Х→ПС  | 4C | 72   | : 13     |
| 07 | П→Х1            | 61 | 40 | П→Х8  | 68 | 73   | X→П7 47  |
| 08 | П→Х4            | 64 | 41 | +     | 10 | 74   | КП→Х7 Г7 |
| 09 | :               | 13 | 42 | В/О   | 52 | 75   | П→Х7 67  |
| 10 | X→П7            | 47 | 43 | П→Х9  | 69 | 76   | П→Х2 62  |
| 11 | КП→Х7           | Г7 | 44 | FX<0  | 5C | 77   | ×        |
| 12 | П→Х1            | 61 | 45 | 55    | 55 | 78   | П→Х1 61  |
| 13 | П→Х4            | 64 | 46 | П→Х1  | 61 | 79   | П→Х8 68  |
| 14 | П→Х7            | 67 | 47 | П→Х6  | 66 | 80   | ×        |
| 15 | ×               | 12 | 48 | П→Х3  | 63 | 81   | ↔ 14     |
| 16 | —               | 11 | 49 | ×     | 12 | 82   | — 11     |
| 17 | X→П5            | 45 | 50 | +     | 10 | 83   | X→ПС 4C  |
| 18 | П→Х9            | 69 | 51 | Х→ПС  | 4C | 84   | П→Х9 69  |
| 19 | FX <sup>2</sup> | 22 | 52 | 1     | 01 | 85   | FX≥0 59  |
| 20 | FV              | 21 | 53 | —     | 11 | 86   | 91 91    |
| 21 | П→Х8            | 68 | 54 | В/О   | 52 | 87   | ↔ 14     |
| 22 | —               | 11 | 55 | П→Х5  | 65 | 88   | П→Х8 68  |
| 23 | FX=0            | 5E | 56 | FX≠0  | 57 | 89   | +        |
| 24 | 70              | 70 | 57 | 61    | 61 | 90   | В/О 52   |
| 25 | ↔               | 14 | 58 | П→Х4  | 64 | 91   | ↔ 14     |
| 26 | 1               | 01 | 59 | П→Х5  | 65 | 92   | П→Х3 63  |
| 27 | —               | 11 | 60 | —     | 11 | 93   | — 11     |
| 28 | X→П6            | 46 | 61 | П→Х1  | 61 | 94   | X→ПС 4C  |
| 29 | FX=0            | 5E | 62 | ↔     | 14 | 95   | П→Х8 68  |
| 30 | 43              | 43 | 63 | П→Х3  | 63 | 96   | — 11     |
| 31 | П→Х9            | 69 | 64 | ×     | 12 | 97   | В/О 52   |
| 32 | FX<0            | 5C | 65 | —     | 11 |      |          |

Замечание. В ПМК «Электроника МК-52» оператор В/О не возвращает к команде 01. В этом случае команды 42, 54, 69, 90, 97 записываются С/П 50 и каждый раз после С/П для индикации первой координаты хода ПМК нажимают БП 02 С/П, например 5 В↑ 6 С/П «2» БП 02 С/П «6» и т. д.

Инструкция. Начальный ввод:  $n$  X→П8,  $n^2-1$  X→П2,  $n-1$  X→П3,  $n+1$  X→П4 X→ПС. Для данного  $n$ , кроме  $n+1$  X→ПС, вводятся один раз независимо от числа сыгранных партий. В начале каждой партии следует нажать клавиши: СХ В/О С/П (1 с) «1», С/П « $n+1$ ». Ввод хода противника и включение вычислений:  $a$  В↑  $b$  С/П, где  $b$  — ближний к диагонали конец отрезка,  $a$  — дальний. При останове индицируется значение  $y$ , после нажатия С/П — значение  $x$ ;  $x, y$  — номера отрезка, проведенного ПМК. Затем снова выводится ход человека и т. д., пока человек не попадет в безнадежную для него ситуацию и не остановит игру, ПМК всегда выигрывает.

Может случиться, что ПМК выведет на индикатор отрицательное число. В этом случае к числам в двух последовательных индикациях, т. е. к значениям  $x$  и  $y$ , необходимо прибавить  $a$ . Например, индицируется  $-3$  и после С/П 1; в предшествующем ходе человека  $a=15$ . В этом случае отрезок, генерируемый ПМК, соединяет точки 12 и 16.

Контрольный пример.  $n=4$ , см. рис. 15.6.

4 X→П8, 15 X→П2, 3 X→П3, 5 X→П4 X→ПС СХ В/О С/П «1», С/П «5» 5 В↑ 6 С/П «2», С/П «6» 9 В↑ 10 С/П «3», С/П «7» 13 В↑ 14 С/П «4», С/П «8» 18 В↑ 14 С/П «12», С/П «11» 14 В↑ 10 С/П «11», С/П «10» 10 В↑ 6 С/П «9», С/П «5» 7 В↑ 6 С/П «13», С/П «9» 4 В↑ 8 С/П «13» С/П «14» 8 В↑ 12 С/П «14» С/П «15»

12 В↑ 16 С/П «20» С/П «16» 15 В↑ 16 С/П «-3» С/П «1». Каждому результату прибавляется 15, получаем «12» и «1», 15 В↑ СС С/П «14» С/П «10» — человек проиграл.

## Структура программы

- 00—03 : индикация очередного хода ПМК начиная с первого:  $1, n+1$ . После каждого останова нажать С/П для продолжения вычислений. Далее, после ввода данных о ходе человека значение  $b$  запоминается в Р1,  $a-b$  — в Р9.
- 07—17 : вычисление остатка от деления  $b$  на  $n+1$ , запоминается в Р5.
- 18—23 : вычисление разности  $|a-b|-n$ .
- 25—28 : если разность равна нулю ( $|a-b|=n$ ), то ход человека горизонтальный; для анализа этого случая подсчитывается разность  $r-1$ , запоминается в Р6.
- 31—42 : если  $r-1=0$  ( $r=1$ ), то конец отрезка  $b$  находится на главной диагонали; ответный ход ПМК вычисляется по формулам (1).
- 30, 43—69 : горизонтальный ход человека при  $r \neq 1$  и соответствующие ответные действия ПМК согласно формулам (2), (3), точнее, когда отрезок, проведенный человеком, находится в левой (команды 43—54) и в правой полуплоскости (команды 55—69).
- 70—97 : вертикальные ходы человека — сначала вычисляется целая часть от деления  $b$  на  $n$ , входящая в формулы ответных ходов ПМК (команды 70—74), затем значение  $x$  для варианта левой полуплоскости, формула (4) (команды 75—83), сохраняется в РС и, наконец, значение  $x$  для правой полуплоскости и соответствующие значения  $y$  (команды 84—97).
- После построения очередного хода ПМК команда В/О каждый раз выводит на индикацию хода.
- Тип хода ПМК алг БРИДЖ-ИТ (нат  $n$ , лит А)
- |            |                                      |
|------------|--------------------------------------|
| формулы    | арг $n$                              |
| Связь с    | рез $A$                              |
| программой | нач нат $a, b, d, x, y$ , цел $k, r$ |
- ком  $b$  — номер ближайшего к диагонали конца отрезка, проведенного противником ПМК в очередном ходе,  $a$  — номер противоположного конца отрезка;  $x, y$  — номера точек, соединенных ПМК в ответном ходе,  $k$  (вводится противником) равен нулю, если игра продолжается, и единице в противном случае
- Ход 1 ПМК  
 $k := 0$ ; выв 1,  $n+1$   
 пока  $k = 0$   
 иц  
 ввод  $a, b, k$   
 если  $k = 1$   
 то  $A :=$  «Вы проиграли»  
 иначе  $r := b - (n+1) * [b/(n+1)]$

выбор

```

при ( $|a - b| = n$ )  $\wedge$  ( $r = 1$ ):
  если  $a < 0$ 
    то  $x := b$ 
    иначе  $x := b - 1$ 
  все
   $y := b + n$ 
при ( $|a - b| = n$ )  $\wedge$  ( $r \neq 1$ )  $\wedge$  ( $a < b$ ):
   $x := b + (n - 1) * (r - 1)$ ;  $y := x - 1$ 
при ( $|a - b| = n$ )  $\wedge$  ( $r \neq 1$ )  $\wedge$  ( $a' > b$ ):
  если  $r = 0$ 
    то  $x := b$ 
    иначе  $x := b - (n - 1) * (n - r + 1)$ 
  все
   $y := x + 1$ 
при  $|a - b| = 1$ :  $d := n * b - (n * 2 - 1) * b / n$ 
  если  $a > b$ 
    то  $x := d$ ;  $y := x + n$ 
    иначе  $x := d - (n - 1)$ 
     $y := x - n$ 
  все
  все
  все
  кон

```

Ход ПМК  
1-го типа, (1)Ход ПМК  
2-го типа, (2)Ход ПМК  
3-го типа, (3)Ходы 4-го и  
5-го типа,  
(4) и (5)

## 3.6. ИГРА «5»

Игра напоминает известную игру С. Ллойда «15». Клетки прямоугольника  $3 \times 2$  занумерованы следующим образом:

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 6 | 5 | 4 |
|   |   |   |

Пять клеток заняты фишками, шестая (левая нижняя) свободна. Фишкам соответствуют номера от 1 до 5. Естественной считается конфигурация, в которой номера фишек и занимаемых ими клеток равны:  $\Phi_1 \Phi_2 \Phi_3$ . В этом случае говорят, что данная позиция совмещена с естественной. Требуется совместить все фишки или сделать заключение, что задачу решить невозможно. Это бывает в том случае, если три фишк уже совмещены, а две оставшиеся — нет.

Программа строится на основе диалога человек — ПМК. Фрагменты программы соответствуют отдельным этапам совмещения. В начале каждого этапа человек информирует ПМК о том, какую фишку требуется установить на место и какую клетку она сейчас занимает. Ход кодируется двузначным числом, первая цифра которого номер фишк, вторая — номер клетки. Например, число 24 означает, что фишк  $\Phi_2$  (с номером два) находится в клетке K4, а фишки с меньшими номерами (в нашем случае  $\Phi_1$ ) уже поставлены на место. Кроме того, за человеком остается решение об окончании игры. ПМК на основе полученных данных выполняет и индицирует последовательность необходимых ходов. На индикатор выводится многозначное число,

состоящее из номеров клеток в той последовательности, в какой делаются ходы. Первая цифра — номер клетки, с которой фишка отправляется в первом ходе, вторая — номер следующей клетки и т. д. Если задача не имеет решения, индицируется ЕГГОГ.

Нетрудно заметить, что каждый компонент в системе выполняет ту функцию, которая ему больше подходит. Человек видит, считывает номера фишек и клеток, и это не требует труда. С той же легкостью мы распознаем ситуацию, когда задача решена. ПМК для выполнения перечисленных функций требовалась бы сложная вычислительная процедура, так как единственное, что он умеет, — это вычислять. ПМК справляется с расчетом ходов, для чего человеку требуются значительные умственные усилия. Диалоговый режим позволяет машине обходиться без «внутренней» модели игры. Она «не помнит» расположения фишек на доске, поскольку внешний мир машины сосредоточен в человеке: только он воспринимает ситуацию и перемещает фишки. Автомат работает примерно так: если введено 12, то вывести на экран 1256, точнее, число извлеченное из регистра, и т. д. Это известный из биологии условный рефлекс: в ответ на некоторый стимул (ввод с клавиатуры) следует определенная реакция — извлечение содержимого очередного регистра.

Интересно, что человек также не помнит ситуацию, каждый раз считывая информацию с игрового поля, которое является как бы его внешней памятью. Однако сняв зрительную копию, человек сопоставляет ей свою внутреннюю модель нумерации клеток (в явном виде нумерация поля не требуется) и на их основе синтезирует двузначное число — ход.

Можно ли исключить человека из контура игры? Конечно можно, если научить ПМК следить за динамикой перемещения фишек. Эта идея использована в алгоритме на УАЯ, который не трудно закодировать в программу для ЭВМ.

Выразительных средств достаточно и у ПМК, но, как это часто бывает, препятствием является ограниченность адресуемой памяти. ПМК больше нуждается в участии человека, чем ЭВМ высокого класса, т. е. является более диалоговым. Диалог, таким образом, не только удобный, но часто и единственно возможный способ участия ПМК в решении задач.

Теперь перейдем к составлению программы.

## Программа ИГРЫ-5

## Распределение памяти

|     |          |        |                                             |
|-----|----------|--------|---------------------------------------------|
| +P1 | 1256     | +P6    | 543256                                      |
| +P2 | 12345612 | +P7    | 12543216                                    |
| +P3 | 54325612 | +P8    | 12345216                                    |
| +P4 | 5216     | PA, PB | Номера фишк и позиции, которую она занимает |
| +P5 | 523456   |        |                                             |

|    |        |    |    |        |    |    |        |    |
|----|--------|----|----|--------|----|----|--------|----|
| 00 | ВП     | 0С | 29 | П→Х2   | 62 | 58 | 1      | 01 |
| 01 | 1      | 01 | 30 | С/П    | 50 | 59 | —      | 11 |
| 02 | / — /  | 0L | 31 | 5      | 05 | 60 | FX = 0 | 5E |
| 03 | X→PA   | 4— | 32 | 6      | 06 | 61 | 65     | 65 |
| 04 | KП→XA  | Г— | 33 | БП     | 51 | 62 | П→X6   | 66 |
| 05 | FО     | 25 | 34 | 82     | 82 | 63 | БП     | 51 |
| 06 | П→XA   | 6— | 35 | 1      | 01 | 64 | 82     | 82 |
| 07 | —      | 11 | 36 | —      | 11 | 65 | П→X7   | 67 |
| 08 | 1      | 01 | 37 | FX = 0 | 5E | 66 | С/П    | 50 |
| 09 | 0      | 00 | 38 | 43     | 43 | 67 | БП     | 51 |
| 10 | X      | 12 | 39 | П→X3   | 63 | 68 | 55     | 55 |
| 11 | X→PВ   | 4L | 40 | С/П    | 50 | 69 | 1      | 01 |
| 12 | П→XA   | 6— | 41 | БП     | 51 | 70 | —      | 11 |
| 13 | 1      | 01 | 42 | 31     | 31 | 71 | FX = 0 | 5E |
| 14 | —      | 11 | 43 | П→X4   | 64 | 72 | 85     | 85 |
| 15 | FX = 0 | 5E | 44 | БП     | 51 | 73 | П→XB   | 6L |
| 16 | 46     | 46 | 45 | 82     | 82 | 74 | 4      | 04 |
| 17 | П→XB   | 6L | 46 | 1      | 01 | 75 | —      | 11 |
| 18 | 2      | 02 | 47 | —      | 11 | 76 | FX = 0 | 5E |
| 19 | —      | 11 | 48 | FX = 0 | 5E | 77 | 81     | 81 |
| 20 | FX = 0 | 5E | 49 | 69     | 69 | 78 | П→X8   | 68 |
| 21 | 25     | 25 | 50 | П→XB   | 6L | 79 | БП     | 51 |
| 22 | П→X1   | 61 | 51 | 3      | 03 | 80 | 82     | 82 |
| 23 | БП     | 51 | 52 | —      | 11 | 81 | П→X7   | 67 |
| 24 | 82     | 82 | 53 | FX = 0 | 5E | 82 | С/П    | 50 |
| 25 | 1      | 01 | 54 | 58     | 58 | 83 | БП     | 51 |
| 26 | —      | 11 | 55 | П→X5   | 65 | 84 | 00     | 00 |
| 27 | FX = 0 | 5E | 56 | БП     | 51 | 85 | K—     | 27 |
| 28 | 35     | 35 | 57 | —      | 82 | 82 |        |    |

**Инструкция.** Введите двузначное число  $A$ , где первая цифра — наименьший номер фишки, которая еще не стоит на причитающемся в естественной последовательности месте, вторая — номер позиции, которую она занимает. Примерно через 10 с после нажатия клавиш В/О С/П на индикаторе появится многозначное число. Его цифры, считая слева, соответствуют номерам позиций, с которых перемещаются фишки в последовательных ходах. Переставив фишки, вводите следующее двузначное число и т. д., пока конфигурация не станет естественной. Если окажется, что клетка К6 не пуста, то после нажатия С/П на индикаторе появится продолжение ходов.

Игра идет до тех пор, пока не получится заданная конфигурация, или до тех пор, пока ПМК не выдаст ЕГГОГ, означающий, что задача не имеет решения. (Не забудьте в начале первой игры ввести данные в Р1—Р8, см. с. 181.)

**Контрольный пример.** Исходная конфигурация:  $\begin{array}{|c|c|c|} \hline 5 & 3 & 2 \\ \hline | & 4 & | \\ \hline 1 & & 1 \\ \hline \end{array}$  Ввод начальных

данных в Р1—Р8: 1256 X→P1 ... 12345216 X→P8. Требуется поставить на место фишку Ф1, занимающую четвертую позицию: 14 В/О С/П (8 с) «54325612». После выполнения ходов возникает конфигурация:  $\begin{array}{|c|c|c|} \hline 1 & | & 3 \\ \hline | & 4 & | \\ \hline 5 & | & 2 \\ \hline \end{array}$ . Клетка К6 еще не освободилась, значит последовательность ходов не завершена. Нажимаем С/П (2 с) «56». Получим конфигурацию:  $\begin{array}{|c|c|c|} \hline 1 & 4 & 3 \\ \hline | & 5 & | \\ \hline 2 & & \\ \hline \end{array}$

на место фишку Ф2: 24 С/П (10 с) «543256», получим конфигурацию:  $\begin{array}{|c|c|c|} \hline 1 & & 2 \\ \hline | & 5 & | \\ \hline \end{array}$

Далее 34 С/П (10 с) «12345216», 45 С/П (13 с) «ЕГГОГ» — задача не имеет решения.

## Структура программы

00—11 : вычисление номера фишки, которая должна попасть на место в данной серии ходов (запоминается в РА), и номера ее позиции (запоминается в РВ).  
 12—45 : перемещение фишки Ф1 из клетки К2 (команды 12—24), К3 (25—34), К4 (35—42), К5 (43—45).  
 46—68 : перемещение фишки Ф2 из клетки К3 (команды 46—57), К4 (команды 58—64), К5 (команды 65—68).  
 69—81 : перемещение фишки Ф3 из клетки К4 (команды 69—80), К5 (команда 81).  
 82 : индикация последовательностей ходов на каждом этапе.  
 72, 85 : если окажется, что Ф4 стоит не на соответствующем месте, то задача не имеет решения.  
 83, 84 : если после установления фишки Ф4 задача еще не решена, то управление переходит к началу программы.

алг ИГРА-5 (нат  $n$ , таб  $\Phi [1 : n]$ )  
 арг  $n$ ,  $\Phi$   
 рез  $\Phi$   
 нач нат  $i, j, k$ , фиш,  $A$   
 для  $i$ , от 1 до  $n - 1$   
 нц  
 выбор  
 при  $(i = 4) \wedge (\Phi[i] = i)$ : выв «задача решена»  
 при  $i = 4$ : выв «нет решения»  
 при  $\Phi[i] = i$ :  $i := i$   
 иначе для  $j$  от  $i$  до  $n$   
 нц  
 если  $\Phi[j] = i$   
 то фиш :=  $i$ ;  $k := j$ ;  $j := n$   
 все  
 кц  
 выбор  
 при фиш = 1: выбор  
 при  $k = 2$ :  $A := 1256$   
 при  $k = 3$ :  $A := 1234561256$   
 при  $k = 4$ :  $A := 5432561256$   
 иначе  $A := 5216$   
 все  
 $\Phi[1] := 1$   
 при фиш = 2: выбор  
 при  $k = 3$ :  $A := 523456$   
 при  $k = 4$ :  $A := 543256$   
 иначе  $A := 12543216523456$   
 все  
 $\Phi[2] := 2$   
 иначе  
 выбор  
 при  $k = 4$ :  $A := 12345216$   
 иначе  $A := 12543216$   
 все  
 $\Phi[3] := 3$   
 все  
 выв  $A$   
 все  
 кон

## ЗАДАНИЕ

В программе предусмотрены шесть безусловных переходов к команде 82. Сократите программу на несколько шагов, используя косвенное обращение.

### 3.7. РОБОТ НАХОДИТ ПУТЬ К ЦЕЛИ

Древние греки сложили легенду о Тесее, который отправился на Крит, чтобы убить чудовище, прятавшееся в лабиринте. Дочь царя Крита, Ариадна, дала герою клубок ниток. Путеводная нить помогла Тесею выполнить задание и выбраться из лабиринта.

В век электроники мифическое сказание обрело новый смысл. Подвиги Тесея повторяют роботы, а нитью Ариадны для ориентации в сложных производственных и технических «лабиринтах» служат программы.

При первом (поисковом) решении робот идет вверх от исходной позиции. Если на пути встретилось препятствие, то он смещается на одну клетку вправо и продолжает движение вверх в смежном столбце. Выйдя к горизонтали, на которой находится цель, робот поворачивает вправо и движется прямо к цели. Если в первом решении на пути встречалось препятствие, то решая задачу повторно, робот сразу же переходит к смежному справа столбцу и движется к цели беспрепятственно.

В одной из клеток квадратной площадки  $9 \times 9$  стоит робот (рис. 16). Он должен попасть в клетку, расположенную выше и правее первой. Каждое перемещение робота на одну клетку по горизонтали или вертикали будем называть ходом. На пути может встретиться одно препятствие (например, гора) в любом месте, за исключением горизонтали, на которой находится цель. Робот должен обойти препятствие и прибыть к цели. При повторном решении из той же исходной клетки робот идет другим путем, свободным от препятствий. (На рис. 16 точками отмечена траектория робота при первом решении задачи, крестиками — при повторном.) Действиями робота управляет программа, которую требуется составить.

Обозначения:  $P$ ,  $\Pi$ ,  $\mathcal{C}$  — двузначные числа, кодирующие соответственно позицию робота, препятствия, целевую клетку. Первая цифра числа — абсцисса клетки (номер столбца), вторая — ордината (номер строки). При ходе вверх прибавляется 1 к ординате, при ходе вправо — к абсциссе.

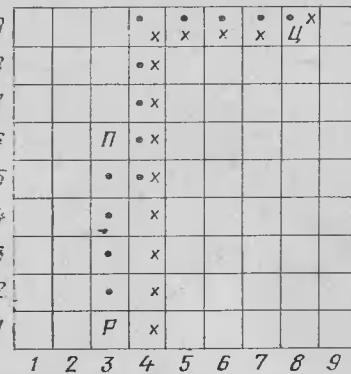


Рис. 16

## Программа РОБОТ

| Распределение памяти                                                               |                       |    |    |                       |    |    |                       |    |  |
|------------------------------------------------------------------------------------|-----------------------|----|----|-----------------------|----|----|-----------------------|----|--|
| $P^4$ — хранение копии начального содержимого РА;                                  |                       |    |    |                       |    |    |                       |    |  |
| $+PA, PB, PC$ — код начальной позиции робота ( $P$ ), препятствия ( $P$ ), целевой |                       |    |    |                       |    |    |                       |    |  |
| позиции ( $C$ ) (двузначные числа);                                                |                       |    |    |                       |    |    |                       |    |  |
| $+RD$ — параметр (для первого решения равен 1, для второго — 0)                    |                       |    |    |                       |    |    |                       |    |  |
| 00                                                                                 | $\Pi \rightarrow XA$  | 6— | 18 | 9                     | 09 | 35 | $X$                   | 12 |  |
| 01                                                                                 | $X \rightarrow \Pi 4$ | 44 | 19 | +                     | 10 | 36 | $F \sin$              | 1C |  |
| 02                                                                                 | $\Pi \rightarrow XD$  | 6Г | 20 | $X \rightarrow \Pi 4$ | 44 | 37 | $FX = 0$              | 5E |  |
| 03                                                                                 | $FX = 0$              | 5E | 21 | СХ                    | 0Г | 38 | 11                    | 11 |  |
| 04                                                                                 | 11                    | 11 | 22 | $X \rightarrow PD$    | 4Г | 39 | $\Pi \rightarrow X9$  | 69 |  |
| 05                                                                                 | $\Pi \rightarrow X4$  | 64 | 23 | $\Pi \rightarrow X4$  | 64 | 40 | $FX \neq 0$           | 57 |  |
| 06                                                                                 | 1                     | 01 | 24 | $C/P$                 | 50 | 41 | 50                    | 50 |  |
| 07                                                                                 | 0                     | 00 | 25 | $\Pi \rightarrow XC$  | 6C | 42 | $\Pi \rightarrow X4$  | 64 |  |
| 08                                                                                 | +                     | 10 | 26 | $\Pi \rightarrow X4$  | 64 | 43 | 1                     | 01 |  |
| 09                                                                                 | $X \rightarrow \Pi 4$ | 44 | 27 | —                     | 11 | 44 | 0                     | 00 |  |
| 10                                                                                 | $C/P$                 | 50 | 28 | $X \rightarrow \Pi 9$ | 49 | 45 | +                     | 10 |  |
| 11                                                                                 | $KP \rightarrow X4$   | Г4 | 29 | 1                     | 01 | 46 | $X \rightarrow \Pi 4$ | 44 |  |
| 12                                                                                 | $\Pi \rightarrow X4$  | 64 | 30 | 0                     | 00 | 47 | $C/P$                 | 50 |  |
| 13                                                                                 | $\Pi \rightarrow XB$  | 6L | 31 | :                     | 13 | 48 | FП                    | 51 |  |
| 14                                                                                 | —                     | 11 | 32 | 1                     | 01 | 49 | 25                    | 25 |  |
| 15                                                                                 | $FX = 0$              | 5E | 33 | 8                     | 08 | 50 | Fп                    | 20 |  |
| 16                                                                                 | 23                    | 23 | 34 | 0                     | 00 | 51 | $C/P$                 | 50 |  |
| 17                                                                                 | $\Pi \rightarrow X4$  | 64 |    |                       |    |    |                       |    |  |

Инструкция. 1-е решение:  $P \rightarrow PA$ ,  $\Pi \rightarrow PB$ ,  $C \rightarrow PC$ , 1  $X \rightarrow PD$  В/О С/П. 2-е решение: В/О С/П. В обоих случаях переключатель углов в положении Г. Ходы робота индицируются двузначными числами: первая цифра — абсцисса занимаемой позиции, вторая — ордината. После каждого сстанова нажатием С/П запускаем счет. Индикация π означает, что задача решена.

Контрольный пример (рис. 16).

1-е решение: 31  $X \rightarrow PA$ , 36  $X \rightarrow PB$ , 89  $X \rightarrow PC$ , 1  $X \rightarrow PD$  В/О С/П (4 с) «32» С/П (8 с) «33» (8 с) «34» (8 с) «35» (10 с) «45» (8 с) «46» (8 с) «47» (8 с) «48» (8 с) «49» (9 с) «59» (8 с) «69» (8 с) «79» (8 с) «89» (6 с) «π» — задача решена.

2-е решение: В/О С/П (3 с) «41» (2 с) «42» (8 с) «43» и т. д. — беспрепятственное движение к цели (после каждой индикации, исключая π, нажмите С/П).

## Структура программы

- 00—01 : для повторного решения в  $P^4$  записывается копия содержимого РА.
- 02—11 : проверка содержимого РД: 0 (повторное решение) направляет робота вправо, позиция индицируется (команды 05—10), иначе (первое решение), робот движется вверх (команда 11). В последнем случае ход вверх пробный и пока не индицируется.
- 12—24 : проверка наличия препятствия (команды 12—15) — если оно есть, то пробный ход отклоняется, из предыдущей позиции делается ход вправо (команды 17—20), в РД засыпается нуль, фиксирующий наличие препятствия (команды 21, 22); если препятствия нет, то проба одоб-

ряется (команда 16), а ход индицируется (команды 23, 24).

25—38: проверяется, вышел ли робот к цели по горизонтали — если разность  $\Pi - P$  кратна 10, т. е. вторые цифры (ординаты)  $\Pi$  и  $P$  совпадали, то цель достигнута по горизонтали (команды 38, 11 и т. д.).

39—51: робот вышел к горизонтали, на которой находится цель. Проверяется, совпадает ли его очередная позиция с целевой (команды 39, 40) — если да, то индикация  $\pi$  означает, что задача решена (команды 41, 50, 51), в противном случае робот делает ход вправо, индицируется его позиция (команды 42—47), далее процедура повторяется (команды 48, 49, 25 и т. д.).

Адреса алг РОБОТ (нат  $P, \Pi, \Pi$ , лит А)  
команд арг  $P, \Pi, \Pi$   
рэз А

нач nat  $P_1$ , цел  $r, M$ , вещ  $q$

$M := 1$

пока  $M \geq 0$

нц

$P_1 := P$

$r := 1$

если  $M = 0$

то  $P_1 = P_1 + 10$ ; выв  $P_1$

все

пока  $r > 0$

нц

$P_1 := P_1 + 1$

если  $P_1 = \Pi$

то  $P_1 = P_1 + 9$

все

выв  $P_1$

$q := (\Pi - P_1)/10$ ;  $r := \Pi - P_1 - [q]*10$

кц

пока  $\Pi \neq P_1$

нц

$P_1 := P_1 + 10$ ; выв  $P_1$

кц

$M := M - 1$

$A := \text{«задача решена»}$

кон

Может показаться, что алгоритм и программа заведомо усложнены. Казалось бы, достаточно сравнить абсциссы, проверить, находятся ли позиции робота и препятствия на одной вертикали, чтобы убедиться в том, что можно подниматься вверх из исходной позиции или позиции, смещенной относительно нее вправо. К сожалению, воспользоваться этим алгоритмом нельзя. Дело в том, что робот узнает о препятствии, только приблизившись вплотную, подобно слепому человеку. До этого абсцисса препятствия роботу не известна. Робот, однако, обладает надежной памятью,

и это как-то компенсирует отсутствие зрения. Изюминкой программы является фрагмент, благодаря которому робот оптимизирует траекторию движения с учетом «прошлого опыта». Робот запоминает вариант, где встретилось препятствие, и при повторном движении из прежней позиции выбирает путь, который без препятствий приводит к цели. Накопление опыта методом проб и ошибок в какой-то мере роднит робота с человеком.

## ЗАДАНИЯ

1. Разрешим роботу движение влево. В этом случае можно снять часть ограничений, касающихся положения целевой клетки: достаточно, чтобы она была выше позиций, исходно занимаемой роботом. В этом случае, если позиция робота не совпадает с целевой (команда 42 и далее), то направление движения определяется знаком содержимого Р9. Соответствующие изменения внесите в программу.

2. Предусмотрим в программе ситуацию, когда цель ограждена от робота препятствиями, занимающими всю ширину горизонтали. В этом случае коды препятствий должны различаться на 10. Теперь кроме содержимого РВ необходимо задать и число препятствий. Наткнувшись на препятствие, робот делает шаг вправо и движется в этом направлении, пока не найдет проход. Если робот в крайнем справа столбце, т. е. его код более 90, а вверх двигаться нельзя, то задача не имеет решения — на индикаторе сигнал ЕГГОГ. Составьте программу по этому условию.

## 3.8. КЛЕТКИ НА ПРЯМОЙ

В предыдущем параграфе отсутствие «зрения» у автомата компенсировала громоздкая вычислительная процедура. Эту тему можно развить в другой ситуации. Рассматриваемую задачу человек решает с одного взгляда — непосредственно считывая ответ с доски. Интересно построить модель воспроизведения автомата некоторых сторон мышления и восприятия человеком.

Клетки квадрата  $3 \times 3$  занумерованы цифрами 1—9 (рис. 17). Составить программу, позволяющую для любых трех клеток проверить, лежат ли их центры на одной прямой: горизонтали, вертикали, диагонали.

Легко убедиться, что если центры действительно расположены на одной прямой, то цифровые коды клеток, упорядоченные по возрастанию, образуют арифметическую прогрессию  $a, b, c$  с разностью  $d=1, 2, 3, 4$ . Пригодны, однако, не все прогрессии этого типа. Условию задачи удовлетворяют случаи:  $d=3$  или  $d=4$ ,  $d=2$  при  $a=3$  и  $d=1$  при  $c$ , кратном трем.

|        |        |        |
|--------|--------|--------|
| 1<br>• | 2<br>• | 3<br>• |
| 4<br>• | 5<br>• | 6<br>• |
| 7<br>• | 8<br>• | 9<br>• |

Рис. 17

## Алгоритм распознавания.

- Запишите коды рассматриваемых клеток в порядке возрастания:  $a, b, c$ .
- Проверьте, образуют ли они арифметическую прогрессию. Если нет, то центры клеток не лежат на одной прямой. В противном случае запомните значение  $d$  и перейдите к указанию 3.
- Если  $d=3$  или  $d=4$ , то центры клеток лежат на одной прямой.

4. Если  $d=2$ , проверьте, будет ли  $a=3$ . При  $a=3$  центры клеток лежат на прямой, в противном случае не на прямой. Если  $d \neq 2$ , переходите к следующему указанию.

5. Если  $d=1$ , то проверьте, будет ли  $c$  кратно трем. При выполнении этого условия центры лежат на прямой, в противном случае или в случае, когда  $d \neq 1$ , центры не лежат на прямой.

### Программа ЦЕНТРЫ КВАДРАТОВ НА ПРЯМОЙ

Распределение памяти  
РА, РВ, РС — коды данных клеток в порядке возрастания;  
РД — значение  $b-a$ .

|    |        |    |    |        |    |    |        |      |
|----|--------|----|----|--------|----|----|--------|------|
| 00 | X→ПС   | 4C | 18 | П→ХД   | 6Г | 36 | 1      | 01   |
| 01 | FO     | 25 | 19 |        | 4  | 04 | 37     | + 10 |
| 02 | X→ПВ   | 4L | 20 | —      | 11 | 38 | FX = 0 | 5E   |
| 03 | FO     | 25 | 21 | ×      | 12 | 39 | 52     | 52   |
| 04 | X→ПА   | 4— | 22 | FX ≠ 0 | 57 | 40 | П→ХС   | 6C   |
| 05 | П→ХВ   | 6L | 23 |        | 50 | 50 | 3      | 03   |
| 06 | П→ХА   | 6— | 24 | П→ХД   | 6Г | 42 | :      | 13   |
| 07 | —      | 11 | 25 | 2      | 02 | 43 | 1      | 01   |
| 08 | X→ПД   | 4Г | 26 | —      | 11 | 44 | 8      | 08   |
| 09 | П→ХС   | 6C | 27 | FX = 0 | 5E | 45 | 0      | 00   |
| 10 | П→ХВ   | 6L | 28 |        | 36 | 36 | ×      | 12   |
| 11 | —      | 11 | 29 | П→ХА   | 6— | 47 | F sin  | 1C   |
| 12 | —      | 11 | 30 |        | 3  | 03 | FX = 0 | 5E   |
| 13 | FX = 0 | 5E | 31 | —      | 11 | 49 | 52     | 52   |
| 14 | 52     | 52 | 32 | FX = 0 | 5E | 50 | 1      | 01   |
| 15 | П→ХД   | 6Г | 33 |        | 52 | 52 | C/P    | 50   |
| 16 | 3      | 03 | 34 |        | 1  | 01 | 0      | 00   |
| 17 | —      | 11 | 35 | C/P    | 50 | 53 | C/P    | 50   |

Инструкция. Введите коды данных клеток в порядке возрастания:  $a, b, c$ ;  $a \uparrow b \uparrow c$  В/О С/П. (Переключатель углов в положении Г.) Если «К»=«1», то центры лежат на одной прямой, если «К»=«0», то центры не лежат на одной прямой.

Контрольные примеры. 1 В↑ 5 В↑ 9 В/О С/П (8 с) «1», 4 В↑ 6 В↑ 8 В/О С/П (10 с) «0», 2 В↑ 5 В↑ 9 В/О С/П (5 с) «0».

### Структура программы

- запись данных в регистры памяти (указание 1 алгоритма).
- проверка, образуют ли данные арифметическую прогрессию с разностью, находящейся в РД (указание 2).
- проверка выполнения условия указания 3.
- проверка выполнения условия указания 4.
- проверка выполнения условия указания 5.

|        |        |        |
|--------|--------|--------|
| (1, 3) | (2, 3) | (3, 3) |
| (1, 2) | (2, 2) | (3, 2) |
| (1, 1) | (2, 1) | (3, 1) |

Рис. 18

### алг ЦЕНТРЫ НА ПРЯМОЙ (нат $a, b, c$ , лит $y$ )

```

арг a, b, c
рез y
нач nat d
d := b - a
если ( $d=c-b$ )  $\wedge$  (( $d=3$ )  $\vee$  ( $d=4$ ))  $\vee$  (( $d=2$ )  $\wedge$  ( $a=3$ ))  $\vee$  (( $d=1$ )  $\wedge$  ( $c-3*[c/3]=0$ ))
то  $y$  := «центры клеток лежат на прямой»
иначе  $y$  := «центры клеток не лежат на прямой»
все
кон

```

### ЗАДАНИЯ

- Замените команды 41—45 двумя командами.
- Указание 5 алгоритма распознавания можно сформулировать иначе. Если  $d=1$ , то проверьте условия  $a=1, a=4, a=7$ . При выполнении хотя бы одного из них центры клеток лежат на одной прямой. Если  $d \neq 1$ , то центры не лежат на одной прямой. Замените команды, начиная с 40, согласно новому варианту указания 5.

3. Точечная модель задачи. Квадрат можно описывать координатами центра  $(x_i; y_j)$ , где  $i=1, 2, 3$ . Тогда решение задачи для данных трех точек, например  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ , сводится к проверке равенства  $(x_3-x_1)(y_2-y_1) = (y_3-y_1)(x_2-x_1)$  (рис. 18). Постройте программу.

### ПОСЛЕСЛОВИЕ К ГЛАВЕ

В главе рассматривались развитые формы диалога человек — ЭВМ.

В задаче «Магические квадраты» человек зрительно распознавал «заблокированные» и «угловые» клетки, ПМК вычислял ходы, т. е. каждый из участников занимался несложной для него работой. Различие механизмов решения — аналитического, основанного на кодировании «геометрии» уравнениями, удобное для ЭВМ, и наглядно-образного, свойственного человеку — отчетливо заметно в задаче «Клетки на прямой». В задаче «Маги-

ческие квадраты» «слепоту» ПМК удается компенсировать численными методами распознавания, позволяющими исключить человека из контура решения. Это, конечно, не означает, что машина овладела человеческими способами решения: человек воспринимает связный рисунок, машина поэлементно вычисляет, но оба решают.

То, что человек присутствует при решении задачи машиной и участвует в нем, — хорошо это или плохо? На вопрос трудно ответить однозначно. В автоматизированных системах управления это, повидимому, недостаток. Но есть сфера деятельности, где важно постоянное участие человека, — это обучение. Обучаться только по результату, по ответу, выдаваемому машиной, нельзя. Человек должен участвовать непосредственно в процессе решения.

Ограниченностю памяти и скорости вычислений, присущие ПМК, несколько компенсируется интенсивным обменом информацией с человеком, при его участии в вычислительном процессе.

Есть у ПМК и «внутренние резервы». В игре «Крестики-нолики» ПМК не может прогнозировать ситуацию, но, захватив инициативу, каждым ходом ставит противника под угрозу проигрыша, заставляя его защищаться. Чтобы достичь этого, используется специальная, не опирающаяся на зрительное восприятие человека процедура распознавания центрально-симметричных клеток.

Достаточно сложную топологию удалось смоделировать в задаче «Тригекс» и игре «Бридж-ит», калькулятор может работать на уровне понятий «Внутренний треугольник», «Внешний треугольник», «Изолированная точка», «Параллельное смещение», «Горизонтальный ход» и др.

В задаче «Робот находит путь к цели» ПМК использует накопленный опыт для приспособления к уточненным в процессе решения условиям. В аналогичной ситуации в задаче «Ловля блоки» (см. гл. 2) ПМК действует в расчете лишь на случай, информация не накапливается и действия автомата малоэффективны. В программу «Робот» заложена возможность оптимизации траектории на основе прошлого опыта. Изучив расположение препятствия, робот при повторном движении выбирает другой путь к цели, свободный от препятствий.

В задаче Игра «5» мы возвращаемся к вопросу об активном сотрудничестве человека и машины при решении задачи. Человек благодаря особенностям восприятия без труда выбирает очередную фишку, указывает позицию, которую она должна занять после хода. Однако он должен задуматься, как преобразовать данную ситуацию в требуемую. ПМК, напротив, «ничего не видит», но располагает алгоритмом перемещения указанной фишкой на указанное место. Это и позволяет ПМК не фиксировать рас-

положения фишек на доске, для него в качестве внешней памяти выступает человек.

Моделирование решения в виде алгоритма на УАЯ показывает, что развитые языки программирования ЭВМ позволяют программно реализовать механизм слежения за перемещениями фишек, а это избавляет от необходимости обращаться за помощью к человеку. Это, кстати, одна из причин, по которым программы для ПМК дублируются алгоритмами на УАЯ.

— Тебе ведомо многое, Шахразада, научи же меня всему, что знаешь, — сказал царь.

— О повелитель, в тебе заговорила подлинная мудрость. Минут тысячелетия, уйдут в небытие народы и государства. Но вечен человеческий разум. Стану учить тебя учиться.

И тут наступило утро.

## Глава 4

# ПРОГРАММИРУЕМЫЙ МИКРОКАЛЬКУЛЯТОР В ОБУЧЕНИИ

## 4.1. ЗАДАЧИ ДВОИЧНОЙ ЛОГИКИ (ОБУЧЕНИЕ ЛОГИЧЕСКОМУ РАССУЖДЕНИЮ)

— Какие выводы вы сделали бы из следующих фактов, — спросил Шерлок Холмс у доктора Ватсона. — Если А виновен и В невиновен, то С виновен; А никогда не ходит на дело вместе с С; С никогда не действует в одиночку; никто, кроме А, В и С, в преступлении не замешан и по крайней мере один из этой тройки виновен.

— Боюсь, что из этих фактов я смогу извлечь не слишком много, сэр. Разве можно, опираясь лишь на них, доказать, кто из трех подозреваемых виновен и кто невиновен?

— Пока что не могу, дорогой Ватсон, — признался Холмс, — но чтобы выдвинуть неопровергнутое обвинение против одного из них, материалов вполне достаточно.

Лицо доктора выражало неподдельное изумление, но Холмс, казалось, решил переменить тему разговора.

— Знаете ли вы, Ватсон, что такое логическое уравнение?

Оставим на время наших герояев и попробуем поговорить об алгебре логики, не забывая о задаче — кто виновен?

Двоичными называют переменные  $x$ ,  $y$ ,  $z$ , принимающие одно из двух значений, 1 или 0 (да — нет, истинно — ложно). Логические операции с двоичными переменными подобны действиям в алгебраических выражениях. Основные из них: отрицание (инверсия, обозначается  $\neg x$ ), логическое сложение (дизъюнкция, соответствует в языке союзу «или» и обозначается  $x \vee y$ ), логическое умножение (конъюнкция, «и»,  $x \wedge y$ ), следование (импликация, «если..., то»,  $x \rightarrow y$ ), сложение по модулю 2 (разделительная дизъюнкция, «либо»,  $x \oplus y$ ). Выражения, образованные двоичными переменными и константами 1, 0 с помощью логических опера-

Таблица 7

| $x$ | $y$ | $\neg x$ | $\neg y$ | $x \vee y$ | $x \wedge y$ | $x - y$ | $x \oplus y$ |
|-----|-----|----------|----------|------------|--------------|---------|--------------|
| 0   | 0   | 1        | 1        | 0          | 0            | 1       | 0            |
| 0   | 1   | 1        | 0        | 1          | 0            | 1       | 1            |
| 1   | 0   | 0        | 1        | 1          | 0            | 0       | 1            |
| 1   | 1   | 0        | 0        | 1          | 1            | 1       | 0            |

ций, называют *двоичными функциями*. При заданных значениях переменных функции также принимают значения 1 или 0. Основные функции определены в табл. 7.

Кроме функций будем рассматривать логические уравнения и соответствующие арифметические модели — уравнения и неравенства. Во всех случаях неизвестными являются двоичные переменные. Связь между логической (двоичной) функцией и соответствующей арифметической моделью такова: при одних и тех же значениях двоичных переменных обе одновременно истинны или одновременно ложны. Арифметические модели логических операций и логических уравнений приведены в табл. 8. Соотношения легко проверить. Например,  $x \oplus y = 1$  и  $x - y = 0$  — оба соотношения либо выполняются, если  $x \neq y$ , либо не выполняются, если  $x = y$ .

С помощью двоичных функций конкретную задачу нетрудно описать системой логических уравнений  $n$  переменных. На основе уравнений строится соответствующая арифметическая модель, которая кодируется в программу для ПМК. Калькулятор проверяет допустимые размещения из цифр 0, 1. Каждое размещение, которое удовлетворяет условию задачи, выводят на индикатор. В противном случае рассматривается другое размещение, пока не завершится анализ всех  $2^n$  вариантов.

Хотя для разных задач программы различны, все они имеют общий фрагмент, отвечающий за перебор возможных размещений. Раз так, будем строить программу из двух частей: блока для генерирования размещений и блока поиска допустимых размещений. Этот блок удобно оформлять в виде подпрограммы, которую нетрудно менять в соответствии с условиями задачи.

Таблица 8

| Логическая операция | Арифметическое выражение | Логическая модель условия | Арифметическая модель |
|---------------------|--------------------------|---------------------------|-----------------------|
| $\neg x$            | $1-x$                    | $\neg x=1$                | $x=0$                 |
| $x \wedge y$        | $x \cdot y$              | $x \wedge y=1$            | $1-x \cdot y=0$       |
| $x \vee y$          | $x+y-x \cdot y$          | $x \vee y=1$              | $x+y \neq 0$          |
| $x-y$               | $1-x+y-x$                | $x-y=1$                   | $y-x \geq 0$          |
| $x \oplus y$        | $(x-y)^2$                | $x \oplus y=1$            | $x-y \neq 0$          |

Алгоритм решения основан на переборе последовательности двоичных чисел.

1. Формируются размещения с повторениями длины  $n$  из единиц и нулей. В качестве исходного размещения выбираем число 0.

2. В данном размещении рассматривается крайняя справа цифра.

3. Если она равна нулю, то заменяется единицей, и в зависимости от результата проверки условия задачи размещение индицируется или отклоняется. Переход к указанию 2.

4. Если рассматриваемая цифра единица, то заменяем ее на нуль. Рассматривается цифра смежного слева старшего разряда двоичного числа. Возврат к указанию 3.

На каждом этапе проверяется окончание. Признаком служит равенство числа рассмотренных размещений числу  $2^n$ .

### Программа ЛОГИКА

|       |                                                                                                                                                                                                                                                                  |    |           |    |           |     |    |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-----------|----|-----------|-----|----|
| P0    | — счетчик, используемый для формирования константы $2^n$ , очистки регистров начиная с P1 и при построении исходного нулевого размещения, а также для перехода к старшему разряду, если цифра данного разряда не нуль; в любом случае начальное значение — $n$ . |    |           |    |           |     |    |
| P1—P5 | — цифры текущего размещения, расположенные в обратном порядке. На основе этих цифр формируется многоразрядное двоичное число (но не более пяти), его старший разряд находится в P5.                                                                              |    |           |    |           |     |    |
| P6    | — для подсчета числа рассмотренных размещений.                                                                                                                                                                                                                   |    |           |    |           |     |    |
| PA    | — константа $n$ для периодического восстановления начального содержимого счетчика P0.                                                                                                                                                                            |    |           |    |           |     |    |
| PB    | — параметр для решения вопроса о формировании и индикации двоичного числа. Исходное значение $b=0$ . Если размещение удовлетворяет условию задачи, то в подпрограмме $b$ получает значение 1. Этот признак является сигналом к индикации.                        |    |           |    |           |     |    |
| PC    | — константа $2^n$ .                                                                                                                                                                                                                                              |    |           |    |           |     |    |
| 00    | X→P0 40                                                                                                                                                                                                                                                          | 21 | CX!       | 0Г | 42        | F я | 20 |
| 01    | X→PA 4—                                                                                                                                                                                                                                                          | 22 | X→PB 4L   | 43 | C/П       | 50  |    |
| 02    | 1 01                                                                                                                                                                                                                                                             | 23 | KП→XB↑ ГЕ | 44 | П→XB      | 6L  |    |
| 03    | B↑ 0E                                                                                                                                                                                                                                                            | 24 | 1 01      | 45 | FX≠0      | 57  |    |
| 04    | 2 02                                                                                                                                                                                                                                                             | 25 | ↔ 14      | 46 | 19        | 19  |    |
| 05    | X 12                                                                                                                                                                                                                                                             | 26 | — 11      | 47 | П→XA      | 6   |    |
| 06    | FL0 5Г                                                                                                                                                                                                                                                           | 27 | KХ→PB↑ LE | 48 | X→P0      | 40  |    |
| 07    | 04 04                                                                                                                                                                                                                                                            | 28 | FL0 5Г    | 49 | 0 00      |     |    |
| 08    | 1 01                                                                                                                                                                                                                                                             | 29 | 32 32     | 50 | B↑ 0E     |     |    |
| 09    | +                                                                                                                                                                                                                                                                | 30 | БП 51     | 51 | 1 01      |     |    |
| 10    | X→PС 4C                                                                                                                                                                                                                                                          | 31 | 34 34     | 52 | 0 00      |     |    |
| 11    | 0 00                                                                                                                                                                                                                                                             | 32 | FX≠0 57   | 53 | × 12      |     |    |
| 12    | X→P6 46                                                                                                                                                                                                                                                          | 33 | 23 23     | 54 | KП→XB↑ ГЕ |     |    |
| 13    | П→XA 6—                                                                                                                                                                                                                                                          | 34 | KП→X6 Г6  | 55 | +         | 10  |    |
| 14    | X→P0 40                                                                                                                                                                                                                                                          | 35 | ПП 53     | 56 | FL0 5Г    |     |    |
| 15    | CX 0Г                                                                                                                                                                                                                                                            | 36 | 61 61     | 57 | 51 51     |     |    |
| 16    | KХ→PB↑ LE                                                                                                                                                                                                                                                        | 37 | П→X6 66   | 58 | C/П 50    |     |    |
| 17    | FL0 5Г                                                                                                                                                                                                                                                           | 38 | П→XC 6C   | 59 | БП 51     |     |    |
| 18    | 15 15                                                                                                                                                                                                                                                            | 39 | — 11      | 60 | 19 19     |     |    |
| 19    | П→XA 6—                                                                                                                                                                                                                                                          | 40 | FX=0 5E   |    |           |     |    |
| 20    | X→P0 40                                                                                                                                                                                                                                                          | 41 | 44 44     |    |           |     |    |

Инструкция. Составьте логическую модель задачи (систему уравнений) и соответствующую арифметическую модель. На основе модели составьте подпрограмму и введите ее в память ПМК, начиная с команды 61. Не более чем через минуту после нажатия на В/О С/П на индикаторе должно появиться  $n$ -значное двоичное число. Чтобы продолжить вычисления, нажмите С/П. Индикация π означает завершение решения. Если число на индикаторе состоит менее чем из  $n$  цифр, подразумевается, что предшествующие цифры — нули.

Контрольный пример. Решим задачу Шерлока Холмса.

Обозначения:  $A, B, C = \begin{cases} 1, & \text{если виновен,} \\ 0 & \text{в противном случае.} \end{cases}$

| Модели: | логическая                   | арифметическая          |
|---------|------------------------------|-------------------------|
|         | $A \neg B \rightarrow C = 1$ | $C - A(1-B) \geq 0$     |
|         | $A \rightarrow \neg C = 1$   | $1 - C - A \geq 0$      |
|         | $C \rightarrow A \vee B = 1$ | $A + B - AB - C \geq 0$ |
|         | $A \vee B \vee C \neq 0$     | $A + B + C \neq 0$      |

### Подпрограмма ХОЛМС, ВАТСОН и ПМК

|    |                       |    |    |        |    |    |        |    |
|----|-----------------------|----|----|--------|----|----|--------|----|
| 61 | $\Pi \rightarrow X_1$ | 61 | 73 | +      | 10 | 85 | —      | 11 |
| 62 | $\Pi \rightarrow X_3$ | 63 | 74 | X → ПД | 4Г | 86 | FX ≥ 0 | 59 |
| 63 | —                     | 11 | 75 | Π → X9 | 69 | 87 | 95     | 95 |
| 64 | $\Pi \rightarrow X_2$ | 62 | 76 | —      | 11 | 88 | Π → ХД | 6Г |
| 65 | $\Pi \rightarrow X_3$ | 63 | 77 | Π → X1 | 61 | 89 | Π → X1 | 61 |
| 66 | ×                     | 12 | 78 | —      | 11 | 90 | +      | 10 |
| 67 | X → П9                | 49 | 79 | FX ≥ 0 | 59 | 91 | FX ≠ 0 | 57 |
| 68 | +                     | 10 | 80 | 95     | 95 | 92 | 95     | 95 |
| 69 | FX ≥ 0                | 59 | 81 | 1      | 01 | 93 | 1      | 01 |
| 70 | 95                    | 95 | 82 | Π → X1 | 61 | 94 | X → ПВ | 4L |
| 71 | $\Pi \rightarrow X_3$ | 63 | 83 | Π → X3 | 63 | 95 | B/O    | 52 |
| 72 | $\Pi \rightarrow X_2$ | 62 | 84 | +      | 10 |    |        |    |

3 В/О С/П (45 с) «010» С/П (25 с) «110» С/П (60 с) «011» С/П (45 с) «110».

Напомним, что первая цифра трехзначного двоичного числа, взятая из Р3, есть значение  $A$ , вторая, из Р2 —  $B$ , третья, из Р1 —  $C$ . Мы видим, что во всех трех индицируемых числах  $B=1$ . Следовательно, независимо от участия  $A$  и  $C$   $B$  виновен. Это имел в виду Шерлок Холмс, заявляя, что один из трех безусловно виновен.

### Структура программы

02—10: вычисление  $2^n$  и его запись в РС.

13—18: очистка регистров, начиная с Р1, т. е. формирование нулевого размещения.

19—22: подготовка содержимого счетчика Р0; очистка РВ в начале цикла построения очередного размещения.

23—27: инверсия рассматриваемого разряда двоичного числа.

28—36: если до инверсии в этом разряде был записан нуль, то к содержимому счетчика Р6 прибавляется 1, а управление переходит на подпрограмму, где, начиная с команды 61, организована проверка, удовлетворяет ли размещение условию задачи: если да, то содержимое РВ равно 1, иначе 0.

37—43: за подпрограммой следует проверка окончания — если содержимое Р6 равно содержимому РС, т. е.  $2^n$ , то π информирует об окончании решения, в противном случае управление переходит к команде 44.

44—46: проверка равенства содержимого РВ нулю — если да, то следует возврат к началу цикла для формирования очередного размещения (команда 19).

47—60: содержимое РВ равно единице, поэтому на основе размещения формируется и индицируется  $n$ -значное двоичное число. Возврат к команде 19.

Адреса алг РАЗМЕЩЕНИЯ (нат  $n$ , лит В)

```

команд   арг n
          рез В
          нач nat i, цел таб А [1:n]
          для i от 1 до n
          нц
          A [i] := 0
          кц
          для i от 1 до 2n
          нц
          выз А
          для j от 1 до n
          нц
          19—60      A [j] := A [j] + 1
          если A [j] = 2
          то A [j] := 0
          иначе j := n
          все
          кц
          кц
          В := «решение окончено»
          кон

```

— Это поразительно, Холмс, что ваш аналитический метод нашел опору в точной алгебре логики — восхищался Ватсон.

— Что вы, дорогой Ватсон, — Холмс впервые за весь вечер улыбнулся, — это вполне закономерно. Но мне кажется, что для того, чтобы этот метод проявил себя во всем блеске, нужны какие-нибудь механизмы для счета, наподобие тех, какие изобретает наш соотечественник Чарлз Бэббидж — профессор из Кембриджа...

### ЗАДАНИЯ

1. Диагностическая задача. Имеются два симптома,  $S_1$  и  $S_2$ , двух болезней,  $x_1$  и  $x_2$ . Известно: 1) при  $x_2$  есть  $S_1$ ; 2) при  $x_1$  и отсутствии  $x_2$  есть  $S_2$ ; 3) при  $x_2$  и отсутствии  $x_1$  нет  $S_2$ . Составьте программу, позволяющую по наличию симптомов вычислять наличие болезней.

2. Семья состоит из пяти человек: Алексей (A), Вера (B), Глеб (Г), Даша (Д), Евгений (E). Если телевизор смотрят  $A$ , то смотрят и  $B$ ; смотрят  $D$  или  $E$ ; смотрят либо  $B$ , либо  $G$ , но не вместе;  $D$  и  $G$  либо смотрят вместе, либо вовсе не смотрят; если смотрят  $E$ , то смотрят  $A$  и  $D$ . Кто смотрит телевизор?

3. Задача об управлении. По сигналу лампы (красной — K, желтой — Ж, зеленой — З) оператор производит определенные действия:  $D_1, D_2$ . Если Ж зажглась, а З нет, то производятся оба действия; если K зажглась, а Ж нет, то не производится второе действие; если производится  $D_1$  или не произ-

водится  $D_2$ , то не зажглась хотя бы одна из двух ламп,  $J$  или  $Z$ . Составьте программу, подсказывающую оператору, что делать в ответ на зажигание определенных сочетаний ламп.

4. Автоматизированный участок. На автоматизированном участке цеха пять станков. Если работают первый и третий станки, то четвертый не работает при условии, что включен пятый станок. Если пятый станок работает вместе со вторым при выключенном первом, то включен третий станок. Если выключен второй или пятый станок, то одновременно выключен и четвертый. Составьте программу, описывающую все возможные (допустимые) состояния участка.

5. Торговый автомат. В торговом автомате имеются четыре щели для монет в 5, 10, 20, 15 коп. Когда в любом сочетании поступает 25 коп., автомат выдает товар. В каждую щель можно опускать только по одной монете. Составьте программу, управляющую работой автомата и сигнализирующую об ошибке, если условие нарушено.

### ОТВЕТЫ

1. Арифметическая модель:  $S_1 - x_2 \geq 0$ ;  $S_2 - x_1(1 - x_2) \geq 0$ ;  $1 - S_2 - x_2(1 - x_1) \geq 0$ .

Инструкция. Значения  $S_1, S_2$  вводятся;  $x_1, x_2$  вычисляются, записываются в регистры 1 и 2 и индицируются двузначным числом:  $x_2x_1 : S_1 \ X \rightarrow P4$   $S_2 \ X \rightarrow P5$  2 В/О С/П.

После каждой индикации для продолжения вычислений нажать С/П. Индикация 1 означает, что задача решена. Ответы собраны в табл. 9. Если переменная принимает значение 1, это означает, что пациент болен (симптом имеется) если 0 — пациент здоров (симптом отсутствует).

Таблица 9

| $S_1$ | $S_2$ | $x_2$ | $x_1$ | $S_1$ | $S_2$ | $x_2$ | $x_1$ | $S_1$ | $S_2$ | $x_2$ | $x_1$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 0     | 1     | 0     | 1     | 0     | 1     | 1     | 0     | 1     |
| 0     | 1     | 0     | 1     | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 0     |
| 0     | 1     | 0     | 0     | 1     | 0     | 0     | 0     | 1     | 1     | 1     | 1     |
|       |       |       |       |       |       |       |       | 1     | 1     | 0     | 0     |

2. Арифметическая модель:  $B - A \geq 0$ ;  $D + E \neq 0$ ;  $B - G \neq 0$ ;  $D - G = 0$ ;  $A \cdot D - E \geq 0$ . Ответ: Телевизор смотрят Г и Д. Время решения — 3 мин.

3. Арифметическая модель:  $D_1D_2 - J(1 - Z) \geq 0$ ;  $1 - D_2 - K(1 - J) \geq 0$ ;  $D_2 \times X(1 - D_1) - JZ \geq 0$ .

Инструкция. Значения  $J, Z, K$  записывают в регистры от третьего до пятого. Соответствующие значения  $D_1$  и  $D_2$  вычисляются и хранятся в Р1 и Р2.

$J \ X \rightarrow P3$  3  $X \rightarrow P4$   $K \ X \rightarrow P5$  2 В/О С/П.

Таблица 10

| $K$ | $J$ | $Z$ | $D_2$ | $D_1$ | $K$ | $J$ | $Z$ | $D_2$ | $D_1$ |
|-----|-----|-----|-------|-------|-----|-----|-----|-------|-------|
| 0   | 0   | 0   | 1     | 0     | 0   | 0   | 1   | 1     | 0     |
| 0   | 0   | 0   | 0     | 1     | 0   | 0   | 1   | 0     | 1     |
| 0   | 0   | 0   | 1     | 1     | 0   | 1   | 0   | 1     | 0     |
| 0   | 0   | 0   | 0     | 0     | 0   | 1   | 1   | 0     | 1     |
| 0   | 0   | 1   | 1     | 0     | 0   | 1   | 0   | 1     | 1     |
| 0   | 0   | 1   | 0     | 0     | 1   | 1   | 1   | 1     | 0     |
| 0   | 0   | 1   | 1     | 0     | 1   | 1   | 1   | 1     | 1     |
| 0   | 0   | 1   | 0     | 1     | 0   | 1   | 1   | 1     | 0     |
| 0   | 1   | 0   | 1     | 0     | 1   | 1   | 0   | 1     | 0     |
| 0   | 0   | 1   | 0     | 0     | 1   | 1   | 1   | 0     | 0     |

После каждой индикации для продолжения вычислений нажать С/П. Индикация 1 означает, что задача решена. Ответы индицируются двузначными двоичными числами (первая цифра — значение  $D_2$ , вторая —  $D_1$ ), они собраны в табл. 10.

4. Арифметическая модель:  $1 - C_4 - C_1C_2C_5 \geq 0$ ;  $C_3 - (1 - C_1)C_2C_5 \geq 0$ ;  $C_2C_5 - C_4 \geq 0$ .

Инструкция.  $C_i = \begin{cases} 1, & \text{если } i\text{-й станок работает,} \\ 0 & \text{в противном случае, } i=1, 2, \dots, 5. \end{cases}$

Значения  $C_1 - C_5$  генерируются в регистрах 1—5; 5 В/О С/П. Результаты индицируются пятизначными двоичными числами. Ответы собраны в табл. 11.

5. Арифметическая модель:  $W + \varphi_2 + \varphi_3 + \varphi_4 = 25$ ;  $(\varphi_i - 5)(\varphi_i - 10)(\varphi_i - 15)(\varphi_i - 20) = 0$ , где  $\varphi_i$  — монета, опущенная в  $i$ -ю щель.

Для решения этой задачи можно составить специальную, упрощенную программу.

Таблица 11

| $C_5$ | $C_4$ | $C_3$ | $C_2$ | $C_1$ | $C_5$ | $C_4$ | $C_3$ | $C_2$ | $C_1$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1     | 0     | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 0     |
| 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 1     |
| 1     | 0     | 1     | 0     | 0     | 1     | 0     | 0     | 0     | 1     |
| 0     | 0     | 0     | 1     | 0     | 0     | 0     | 1     | 0     | 1     |
| 0     | 0     | 1     | 1     | 0     | 1     | 0     | 1     | 0     | 1     |
| 1     | 0     | 1     | 1     | 0     | 0     | 0     | 0     | 1     | 1     |

### 4.2. УРОК БИОЛОГИИ (УСЛОВНЫЙ РЕФЛЕКС)

У собаки, получавшей в  $n$  экспериментах пищу в сопровождении звонка, образуется условный рефлекс слюноотделения на звонок. Если рефлекс не подкрепляется — достаточно долго дается пища без звонка или звонок без пищи, то рефлекс угасает. Модель эксперимента нетрудно воссоздать с помощью ПМК. Значение  $n$  характеризует индивидуальные особенности собаки (стойкость рефлекса).

Обозначения:

$a = \begin{cases} 1, & \text{если дается звонок,} \\ 0 & \text{в противном случае;} \end{cases}$

$b = \begin{cases} 1, & \text{если собака получает пищу,} \\ 0 & \text{в противном случае.} \end{cases}$

Программа УСЛОВНЫЙ РЕФЛЕКС

|    |                     |    |    |                    |    |    |                    |    |
|----|---------------------|----|----|--------------------|----|----|--------------------|----|
| 00 | $X \rightarrow P1C$ | 4C | 09 | $P \rightarrow X4$ | 64 | 18 | $P \rightarrow XC$ | 6C |
| 01 | С/П                 | 50 | 10 | 1                  | 01 | 19 | —                  | 11 |
| 02 | $X \rightarrow P1B$ | 4L | 11 | —                  | 11 | 20 | $FX < 0$           | 5C |
| 03 | Х                   | 12 | 12 | $FX < 0$           | 5C | 21 | 07                 | 07 |
| 04 | $FX \neq 0$         | 57 | 13 | 15                 | 15 | 22 | $P \rightarrow XB$ | 6L |
| 05 | 09                  | 09 | 14 | CX                 | 0Г | 23 | $FX = 0$           | 5E |
| 06 | $KP \rightarrow X4$ | 4Г | 15 | $X \rightarrow P4$ | 44 | 24 | 07                 | 07 |
| 07 | 1                   | 01 | 16 | 1                  | 01 | 25 | B/O                | 52 |
| 08 | B/O                 | 52 | 17 | +                  | 10 |    |                    |    |

**Инструкция.** Проверьте содержимое Р4 — вначале оно должно быть равно 0. Наберите  $n$  В/О С/П, ПМК тотчас же выведет на индикатор  $n$ . Введите значения  $a$  и  $b$  (нули, единицы):  $a$  В↑  $b$  С/П (3 с) « ». Индикация 1 означает, что слюноотделение есть, 0 — слюноотделения нет. Можно приступать к очередному эксперименту — вводить новые значения  $a$  и  $b$  и т. д.

**Эксперименты.**  $n=3$ . СХ X→Р4 3 В/О С/П (1 с) «3».

| Номер эксперимента | Ввод значений |     | Ввод данных<br>$a$ В↑ $b$ С/П С/П | Индикация реакции собаки | Пояснения                                                                                                                |
|--------------------|---------------|-----|-----------------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------|
|                    | $a$           | $b$ |                                   |                          |                                                                                                                          |
| 1                  | 1             | 1   | 1 В↑ С/П (3 с)                    | «1»                      | Звонок сопровождается выдачей пищи, слюна отделяется; в Р4—1                                                             |
| 2                  | 1             | 1   | 1 В↑ С/П (3 с)                    | «1»                      | В Р4—2, постепенно накапливается готовность к образованию рефлекса                                                       |
| 3                  | 1             | 1   | 1 В↑ С/П (3 с)                    | «1»                      | В Р4—3, оно равно $n$ , рефлекс выработан                                                                                |
| 4                  | 1             | 0   | 1 В↑ 0 С/П (5 с)                  | «1»                      | Рефлекс «на звонок» сработал; слюна выделяется несмотря на отсутствие пищи. Однако он неустойчив: в Р4—2                 |
| 5                  | 1             | 0   | 1 В↑ 0 С/П (5 с)                  | «0»                      | Неподкрепленный рефлекс угас, в Р4—1                                                                                     |
| 6                  | 0             | 1   | 0 В↑ 1 С/П (6 с)                  | «1»                      | Слюноотделение связано с наличием пищи. Не сопровождаемое звонком, оно тормозит рефлекс, уменьшая его потенциал: 0 в Р4. |

Эксперимент можно продолжать. Чем больше единиц «аккумулировано» в Р4, тем устойчивее условный рефлекс.

### Структура программы

Число  $n$  — минимальное число тренировок, необходимое для образования рефлекса, записывается в РС. Каждый раз вводятся «значения» звонка и пищи (1 — есть, 0 — нет):  $a$  В↑  $b$  С/П, где  $a$  — «значение» звонка,  $b$  — пищи. «Значение» пищи хранится в РВ (команды 00, 02). Если  $a=b=1$ , то содержимое Р4 увеличивается на 1; индицируется реакция собаки (команды 01, 06—08). В остальных случаях содержимое Р4 уменьшается на 1, однако оно не становится меньше нуля (команды 09—15). Если предыдущее содержимое Р4 меньше  $n$ , то при наличии пищи индицируется 1, при отсутствии — 0 (команды 16—20, 22, 24, 25). Если содержимое Р4 не меньше  $n$ , то рефлекс выработан — индицируется 1 (команды 21, 07, 08).

### алг УСЛОВНЫЙ РЕФЛЕКС

```

нач цел  $a$ ,  $b$ ,  $n$ , nat  $N$ ,  $i$ , ЧЭ, лит с
ком  $a$ ,  $b$ —значения звонка, пищи (0,1),  $n$ —число экспериментов, в которых
 $a=b=1$ ,  $N$ —порог выработки рефлекса, ЧЭ—число экспериментов
ввод  $N$ , ЧЭ;  $n:=0$ 
для  $i$  от 1 до ЧЭ
нц
    ввод  $a$ ,  $b$ 
    если  $(b=1) \vee (a=1) \wedge (n \geq N)$ 
        то  $c :=$  слюна «выделяется»
        иначе  $c :=$  «слюна не выделяется»
    все
    выв  $c$ 
    если  $a*b=1$ 
        то  $n := n + 1$ 
        иначе  $n := n - 1$ ; если  $n < 0$ 
            то  $n := 0$ 
    все
    кц
кон

```

### ЗАДАНИЯ

1. В описанных экспериментах сформированный рефлекс тормозится как только содержимое Р4 станет меньше  $n$ . Реальный физиологический механизм сложнее и включает элемент памяти, т. е. различаются интервалы формирования рефлекса и его утери. Чтобы учесть это, введем наряду с  $n$  характеристику «остаточного рефлекса»  $k < n$ . Если после формирования рефлекса в результате торможений содержимое Р4, ставшее меньше  $n$ , все же остается больше или равно  $k$ , то реакция слюноотделения сохраняется. Составьте программу по этой модели.

**Указание.** Одновременно вводятся  $n$  и  $k$ . До выработки рефлекса (первый индикации 1) содержимое Р4 сравнивается с  $n$ , после — с  $k$ . Если содержимое Р4 стало меньше  $k$ , то процедура начинается сначала.

2. Законы наследственности (первый закон Менделея).  $A$ ,  $B$  — генотипы скрещиваемых особей, каждый характеризуется парой аллельных состояний из некоторого множества. Пусть возможных аллелей два, обозначим их  $Ч$ ,  $К$ . Тогда возникают следующие генотипы:  $ЧЧ$ ,  $ЧК$ ,  $КЧ$ ,  $KK$ . Если, например, скрещиваются особи с генотипами  $ЧЧ$  и  $ЧК$ , то могут возникнуть гены  $ЧЧ$ ,  $ЧК$ ,  $ЧК$ .

Генотипы будем кодировать двузначными числами, их цифры в случае двух возможных аллельных состояний 9, 6; в случае трех — 9, 6, 7.

**Алгоритм.** Для случая двух аллелей возможны следующие коды генов: 99, 96, 66. Пусть  $A=99$ ,  $B=96$  — гены одной особи — находятся в аллельных состояниях 9, 9, второй — 9, 6. Итогом скрещивания  $A \cdot B = 99 \cdot 96$  являются четыре двузначных числа: 99, 96, 99, 96.

Условимся вводить данные в ПМК следующим образом:  $A$  В↑  $B$  В/О С/П. ПМК будет выводить на индикатор все возникающие генотипы. Составьте программу «Законы наследственности». С помощью ПМК решите следующие задачи.

2.1. Ген черной окраски крупного рогатого скота доминирует над геном красной окраски.

1. Какое потомство получится при скрещивании черного гомозиготного быка с красными коровами?

2. Каким будет потомство от скрещивания между собой таких гибридов?  
 3. Какие телята рождаются от красного быка и гибридных коров?

**Пояснение.** Ген называется доминирующим, если свойство,носителем которого он является, присваивается потомству независимо от другого гена, входящего с ним в единый генотип. Генотип гомозиготен, если оба его гена находятся в одинаковых аллельных состояниях.

**2.2.** Группа крови человека определяется наследственным признаком, который зависит от одного гена, имеющего три аллели — *A*, *B*, *O*. Первую группу крови имеют лица с генотипом *OO*, вторую — *AA* или *AO*, третью — *BB* или *BO*, четвертую — *AB*.

1. Какие группы крови возможны у детей, если у матери вторая группа крови, а у отца первая?

2. Какие группы крови возможны у детей, если у матери первая группа крови, а у отца четвертая?

3. Могут ли дети унаследовать группу крови своих родителей?

4. В родильном доме перепутали двух мальчиков. Родители одного из них имеют первую и вторую группу крови, другого — вторую и четвертую. Исследование показало, что дети имеют первую и вторую группу крови. Определите, кто чей сын.

## ОТВЕТЫ

### 2.1. Программа ЗАКОНЫ НАСЛЕДСТВЕННОСТИ\*

Распределение памяти

P1 — код *A* (двухзначное число);  
 P2 — код *B*;

РА, РВ, РС, РД — соответственно вторая и первая цифры *A* и вторая и первая цифры *B*.

|    |      |    |    |      |    |    |       |    |
|----|------|----|----|------|----|----|-------|----|
| 00 | X→P2 | 42 | 17 | X    | 12 | 34 | 0     | 00 |
| 01 | ↔    | 14 | 18 | X→P4 | 44 | 35 | :     | 13 |
| 02 | X→P1 | 41 | 19 | P→ХД | 6Г | 36 | X→P7  | 47 |
| 03 | 1    | 01 | 20 | +    | 10 | 37 | KП→X7 | Г7 |
| 04 | 4    | 04 | 21 | PП   | 53 | 38 | FО    | 25 |
| 05 | X→P0 | 40 | 22 | 47   | 47 | 39 | P→X7  | 67 |
| 06 | 2    | 02 | 23 | P→X4 | 64 | 40 | KХ→P0 | L0 |
| 07 | X→P3 | 43 | 24 | P→ХС | 6С | 41 | —     | 11 |
| 08 | P→X2 | 62 | 25 | +    | 10 | 42 | 1     | 01 |
| 09 | PП   | 53 | 26 | PП   | 53 | 43 | 0     | 00 |
| 10 | 33   | 33 | 27 | 47   | 47 | 44 | ×     | 12 |
| 11 | P→X1 | 61 | 28 | P→XA | 6— | 45 | KХ→P0 | L0 |
| 12 | PП   | 53 | 29 | FL3  | 5— | 46 | B/O   | 52 |
| 13 | 33   | 33 | 30 | 15   | 15 | 47 | C/P   | 50 |
| 14 | P→XB | 6L | 31 | Fп   | 20 | 48 | B/O   | 52 |
| 15 | 1    | 01 | 32 | C/P  | 50 |    |       |    |
| 16 | —0   | 00 | 33 | 1    | 01 |    |       |    |

**Инструкция.** *A* В↑ *B* В/O С/П. Через 23 с индицируется возможный вариант генотипа потомства. После нажатия С/П на индикатор выводится

\* Автором программы является М. И. Данелич, учитель школы.

следующий генотип и т. д. Индикация восьмизначного приближения я означает, что задача решена.

В качестве контрольной решим задачу 2.1. Исходные генотипы *ЧЧ*, *КК*: у коровы исключен ген *Ч*, иначе, вследствие доминирования этого гена, она имела бы черную масть.

Кодирование: *Ч* — 9, *К* — 6.

1) 99×66: 99 В↑ 66 В/O С/П (17 с) «96» С/П (1 с) «96» С/П (3 с) «96» С/П «96» С/П (1 с) «л». Все потомки первого поколения окажутся черными, но поскольку генотипы не гомозиготны, в следующих поколениях возможны красные потомки.

2) Скрещиваются полученные гибриды 96×96: 96 В↑ В/O С/П «99» С/П «96» С/П «69» С/П «66» С/П «л». С вероятностью 3/4 потомство черной масти, 1/4 — красной.

3) 96×66: 96 В↑ 66 В/O С/П «96» С/П «96» С/П «66» С/П «66» С/П «л». С равной вероятностью потомство черного или красного цвета.

Остановимся лишь на некоторых ключевых фрагментах. Команды 14—30: расчет генотипов, возникающих в результате скрещивания; 33—46: нахождение кодов аллелей данных генотипов и их запись в регистрах А—Д (подпрограмма). Следует особо пояснить подпрограмму, начинаяющуюся с команды 47. Если в задаче выводится генотип, образующийся при скрещивании, то она не нужна. Вместо команд 21, 22, а также 26, 27 в этом случае достаточно поставить С/П. Обращение к подпрограмме предусмотрено на тот случай, когда выводятся не сами генотипы, а числовая характеристика некоторого свойства, определяемого генотипами, а потому подпрограмма зависит от конкретного типа задачи. Такая ситуация возникает в задаче «Группа крови», где удобно выводить символы 1, 2, 3, 4, т. е. номера групп крови, которые вычисляются на основе возникших генотипов. Ниже приведена соответствующая подпрограмма (переключатель углов в положении Г).

|    |        |    |    |        |    |    |        |    |
|----|--------|----|----|--------|----|----|--------|----|
| 47 | X→P9   | 49 | 57 | 1      | 01 | 67 | ×      | 12 |
| 48 | 6      | 06 | 58 | —      | 11 | 68 | F sin  | 1C |
| 49 | 6      | 06 | 59 | FX = 0 | 5E | 69 | FX = 0 | 5E |
| 50 | —      | 11 | 60 | 64     | 64 | 70 | 74     | 74 |
| 51 | FX = 0 | 5E | 61 | 4      | 04 | 71 | 2      | 02 |
| 52 | 56     | 56 | 62 | БП     | 51 | 72 | БП     | 51 |
| 53 | 1      | 01 | 63 | 75     | 75 | 73 | 75     | 75 |
| 54 | БП     | 51 | 64 | П→X9   | 69 | 74 | 3      | 03 |
| 55 | 75     | 75 | 65 | 6      | 06 | 75 | C/P    | 50 |
| 56 | 3      | 03 | 66 | 0      | 00 | 76 | B/O    | 52 |

С ее помощью нетрудно решить задачу 2.2 («Группа крови»).

Кодирование: *A* — 9, *B* — 7, *O* — 6.

2.2.1. 99 В↑ 66 В/O С/П — последовательно индицируются «2», «2», «2», «2», «л»; 96 В↑ 66 В/O С/П — «2», «2», «1», «1», «л». В шести случаях из восьми, т. е. с вероятностью 3/4 у детей вторая группа крови, с вероятностью 1/4 — первая.

2.2.4. Группа крови, унаследованная сыном первых родителей, — вторая или первая, см. решение задачи 2.2.1. Группа крови у сына вторых родителей

99×97 или 96×97, 99 В↑ 97 В/О С/П, индикации «2», «4», «2», «4», «л»; 96 В↑ 97 В/О С/П «2», «4», «2», «3», «л». Ребенок второй пары родителей наследует кровь второй, третьей или четвертой группы. Теперь не составляет труда сделать вывод, что ребенок с первой группой крови является сыном первой пары родителей, второй ребенок — сын второй пары родителей.

#### 4.3. ПСИХОЛОГИЧЕСКИЙ ЭКСПЕРИМЕНТ. ПРОВЕРКА И ТРЕНИРОВКА ПАМЯТИ

На индикаторе последовательно появляются три случайные цифры  $A$ ,  $B$ ,  $C$ , каждая высвечивается совсем недолго и исчезает. Испытуемый старается запомнить экспонируемые числа и поочередно ввести их в ПМК. Калькулятор проверяет правильность чисел и сообщает об успешном исходе индикацией 1, на-против, 0 означает, что результаты не совпадают. Каждый раз, когда задача решена успешно, порядок чисел увеличивается на единицу, а значит, все труднее справится с задачей. Так проверяется объем кратковременной числовой памяти человека — максимальная длина чисел, которые он в состоянии быстро запоминать и воспроизвести.

Имеется следующий алгоритм.

Случайные дроби из промежутка  $[0; 1]$  строятся по формуле  $x_{n+1} = (1/\pi) \arccos(\cos 10^9 x_n)$ , где  $x_0$  произвольно и находится в этом же интервале. Целые случайные числа получаются как результат умножения  $x$  на 10, 100, 1000 и т. д., причем определяются целые части произведений. В итоге образуются однозначные, двузначные и т. д. числа. (В действительности числа не совсем случайны, в первую очередь, потому, что при достаточно больших значениях  $n$  наступает повторение. Такие числа принято называть псевдослучайными.\*).

#### Программа ПРОВЕРКА ПАМЯТИ

##### Распределение памяти

- P0 — счетчик на три повторения цикла, по количеству экспонируемых и запоминаемых чисел.
- P1 — число повторений  $m$  экспонируемых чисел для их индикации в течение необходимого времени;
- P4 и P5 — счетчики числа верных ответов и ошибок;
- P9 —  $n$  — множитель для вычисления содержимого РА, РС на основе содержимого РД;
- РА—РС — запись целых однозначных, двузначных и т. д. чисел, предназначенных для вывода на индикатор;
- РД — формирование случайных дробей.

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

|    |                     |    |    |       |    |    |       |    |
|----|---------------------|----|----|-------|----|----|-------|----|
| 00 | X→ПД                | 4Г | 23 | +     | 10 | 46 | X→П0  | 40 |
| 01 | СХ                  | 0Г | 24 | X→П2  | 42 | 47 | F0    | 25 |
| 02 | X→П4                | 44 | 25 | КП→Х2 | Г2 | 48 | 1     | 01 |
| 03 | X→П5                | 45 | 26 | П→Х2  | 62 | 49 | 3     | 03 |
| 04 | П→Х9                | 69 | 27 | КХ→П6 | Л6 | 50 | X→П3  | 43 |
| 05 | ВП                  | 0С | 28 | FL0   | 5Г | 51 | F0    | 25 |
| 06 | 1                   | 01 | 29 | 12    | 12 | 52 | КП→Х3 | Г3 |
| 07 | X→П9                | 49 | 30 | 3     | 03 | 53 | —     | 11 |
| 08 | 3                   | 03 | 31 | X→П0  | 40 | 54 | FX=0  | 5Е |
| 09 | X→П0                | 40 | 32 | 9     | 09 | 55 | 63    | 63 |
| 10 | 9                   | 09 | 33 | X→П6  | 46 | 56 | FL0   | 5Г |
| 11 | X→П6                | 46 | 34 | П→Х7  | 67 | 57 | 51    | 51 |
| 12 | П→ХД                | 6Г | 35 | X→П1  | 41 | 58 | КП→Х4 | Г4 |
| 13 | ВП                  | 0С | 36 | КП→Х6 | Г6 | 59 | 1     | 01 |
| 14 | 9                   | 09 | 37 | B↑    | 0Е | 60 | С/П   | 50 |
| 15 | F cos               | 1Г | 38 | FL1   | 5L | 61 | БП    | 51 |
| 16 | F cos <sup>-1</sup> | 1— | 39 | 37    | 37 | 62 | 04    | 04 |
| 17 | Fл                  | 20 | 40 | FL0   | 5Г | 63 | КП→Х5 | Г5 |
| 18 | :                   | 13 | 41 | 34    | 34 | 64 | 0     | 00 |
| 19 | X→ПД                | 4Г | 42 | Fл    | 20 | 65 | С/П   | 50 |
| 20 | П→Х9                | 69 | 43 | С/П   | 50 | 66 | БП    | 51 |
| 21 | ×                   | 12 | 44 | B↑    | 0Е | 67 | 30    | 30 |
| 22 | 1                   | 01 | 45 | 3     | 03 |    |       |    |

Инструкция. ВП  $n$  X→П9  $m$  X→П7. Наберите произвольную дробь  $N$  из промежутка  $0 < N < 1$ , В/О С/П. Следите за индикатором, ожидая появления трех чисел (однозначных, двузначных и т. д.), каждое из которых будет мерцать на индикаторе в течение нескольких секунд. Останов с индикацией «л» означает, что случайные целые числа записаны в регистрах  $A$ ,  $B$ ,  $C$ , и можно вводить свои числа  $A$ ,  $B$ ,  $C$ :  $A$  B↑  $B$  B↑  $C$  С/П.

Если вы верно воспроизвели все три числа, то примерно через 10 с индицируется 1; после нажатия С/П индицируется новая тройка чисел с порядком на единицу большим чем у предшествующих, и т. д. Если хотя бы одно из трех чисел воспроизведено неверно, то индицируется 0, и после нажатия С/П вторично индицируются те же числа. Так продолжается до тех пор, пока не будут верно введены все три числа.

Число верных ответов хранится в Р4, число ошибок — в Р5.

Контрольный эксперимент. ВП 1 X→П9 3 X→П7 0,327 В/О С/П (45 с) «л» 55 B↑ 74 B↑ 6 С/П (10 с) «1» С/П (45 с) «л» 57 B↑ 730 B↑ 303 С/П (10 с) «1» С/П «л» 5187 B↑ 383 B↑ 1103 С/П «0» — испытуемый не запомнил два четырехзначных и одно трехзначное число (всего 11 цифр) при трехкратной индикации каждого числа в течение 1,5 с. Повторим эксперимент. После нажатия С/П через 17 с появляется «л». На этот раз испытуемый вводит 5178 B↑ 337 B↑ 1013 С/П (10 с) «0» снова неверно.

Интересно, что ошибка приходится на второе число, первое и третье воспроизведены верно. В этом проявляется известный в психологии «эффект края» — в мнемонической деятельности человека преимущественно запоминаются начальный и конечный отрезок ряда.

Увеличим время индикации чисел, изменив содержимое Р7: 5 X→П7. Теперь каждое число будет находиться на индикаторе около 2,5 с.

Продолжим эксперимент. С/П «л» 5178 B↑ 373 B↑ 1013 С/П «1» — барьер удалось преодолеть. С/П «л» 16392 B↑ 22998 B↑ 5905 С/П «0», снова увеличиваем время индикации и т. д. Характерно, что объем кратковременной памяти можно тренировать. Однако, начиная с некоторого момента, даже при очень большом времени экспозиции испытуемые уже не в состоянии запомнить числа.

Так экспериментально можно установить объем кратковременной числовой памяти и его зависимость от времени предъявления информации.

## Структура программы

- 00—09: ввод начальных данных — запись  $N$  в РД, очистка Р4, Р5 (команды 01—03), увеличение порядка пробных чисел на единицу (команды 04—07), подготовка счетчика, управляющего индикацией (команды 08—09).
- 10—27: формирование трех случайных целых чисел и их запись в РА—РС; установка начального адреса для записи чисел в РА—РС командой КП6 (команды 10, 11), генерирование псевдослучайных дробей  $x_{n+1} = (1/\pi) \times \pi \operatorname{arccos}(\cos 10^{\circ} x_n)$  (команды 12—19); вычисление целой части произведения псевдослучайной дроби (РД) и константы, отвечающей за порядок чисел на индикаторе (Р9). Результат направляется в один из регистров А—С (команды 20—27).
- 28, 29: проверка окончания цикла.
- 30—43: вывод чисел на индикатор, продолжительность индикации устанавливает значение (Р7). Промелькнувшее на индикаторе число  $m$  — это сигнал «на старт», вслед за ним начинает мигать очередное число, которое требуется запомнить.
- 44—67: проверка совпадения введенных чисел с заданными; при совпадении индицируется 1, при несовпадении — 0. В первом случае цикл повторяется, начиная с вычисления следующей случайной дроби, во втором — цикл повторной индикации.

Если вы решили проверить свою память или заняться тренировками, то делать это лучше всего вечером, при неярком освещении. Попробуйте также исключить команды 59, 60, 64, 65, т. е. обратную связь. Проверьте, как это повлияет на обучаемость. В любом случае итоги эксперимента нетрудно найти в регистрах 4 и 5.

## ЗАДАНИЕ

Индикатор ПМК светится не очень ярко, поэтому эксперимент нельзя провести при естественном дневном освещении. В связи с этим предусмотрим в программе останов ПМК для прочтения числа с индикатора. Нажав С/П, включим ПМК каждый раз для продолжения вычислений. Измените программу так, чтобы ПМК останавливался для индикации очередного числа.

Указание. В новой программе не требуется ввод  $m$  Х→Р7, нужно исключить команды 34, 35, 37—39; после КП→Хб ставится команда С/П; после FL0 переход к команде КП→Хб. Соответственно должна измениться нумерация команд и адреса переходов в командах начиная с 34.

## 4.4. КОНТРОЛЬ ЗНАНИЙ

Еще один вид диалога реализуется в программах обучения и контроля знаний.

Для контроля знаний понадобятся карточки. В карточке четыре вопроса, причем каждому вопросу соответствует несколько ответов, среди которых только один верный. Отвечая на вопрос, учащийся находит в списке правильный по его мнению ответ и вводит в ПМК двузначное число — код ответа: первая цифра — номер вопроса, вторая — номер ответа. ПМК проверяет ответ: при верном ответе индицируется символ 1 (да), при неверном — 0 (нет). Одновременно к оценке прибавляется единица, если ответ верный. После того, как ответ дан на все четыре вопроса, ПМК сообщает оценку, равную в простейшем случае числу правильных ответов — 0, 1, 2, 3, 4.

Правильные ответы может вводить учитель, но для самоконтроля придется искать другое решение, учитывая, что контроль знаний лишен смысла, если правильные ответы вводят ученик. И все же выход есть — учащийся должен вводить нужную ПМК информацию в зашифрованном виде. Разумеется, шифр, которым пользуется учитель, составляя коды верных ответов, должен быть не известен ученику. В этом случае программу нужно дополнить процедурой расшифровки и размещения кодов.

В предлагаемой ниже программе «Самоконтроль» к кодам правильных ответов прибавляются четыре — к первому, единица — ко второму, двойка — к третьему, тройка — к четвертому. В итоге получаются двузначные числа  $A_4, A_1, A_2, A_3$ , где индексы — это число прибавляемых единиц. На их основе строится восьмизначное ключевое число  $A = \overline{A_4 A_1 A_2 A_3}$ , которое прилагается к заданию (черта сверху означает, что речь идет о едином числе, а не произведении  $A_4$  на  $A_1$  и т. д.).

### Программа САМОКОНТРОЛЬ (A — ключевое число)

- P0 — Распределение памяти  
— используется для анализа верных ответов, начальное значение 3, конечное 0. При исполнении второй части программы там хранится параметр для организации внутренних циклов; начальное значение 13, конечное 10.
- P1 — счетчик циклов для ввода ответов, начальное значение 4, конечное 0.
- P2 — счетчик, контролирующий правильность ответов; начальное значение 4, конечное 0.
- P4 — хранение кода ответа ученика.
- P5 — счетчик верных ответов; начальное значение нуль, конечное — оценка за работу.
- P8 — вычисление верных ответов на основе кодированных ответов.
- РА—РД — двузначные коды верных ответов на четыре вопроса.

|    |       |    |    |                  |    |    |        |    |
|----|-------|----|----|------------------|----|----|--------|----|
| 01 | X→P8  | 48 | 20 | F10 <sup>x</sup> | 15 | 40 | П→X4   | 64 |
| 02 | 0     | 00 | 21 | ×                | 12 | 41 | —      | 11 |
| 03 | X→P5  | 45 | 22 | П→X0             | 60 | 42 | FX=0   | 5E |
| 04 | 4     | 04 | 23 | —                | 11 | 43 | 51     | 51 |
| 05 | X→P1  | 41 | 24 | KX→P3            | L3 | 44 | KП→X5  | F5 |
| 06 | 3     | 03 | 25 | FL0              | 5F | 45 | 9      | 09 |
| 07 | X→P0  | 40 | 26 | 10               | 10 | 46 | KX→PB↑ | LE |
| 08 | 1     | 01 | 27 | П→X8             | 68 | 47 | 1      | 01 |
| 09 | 4     | 04 | 28 | 4                | 04 | 48 | C/P    | 50 |
| 10 | X→P3  | 43 | 29 | —                | 11 | 49 | БП     | 51 |
| 11 | P→X8  | 68 | 30 | X→PA             | 4— | 50 | 56     | 56 |
| 12 | ВП    | 0C | 31 | Fп               | 20 | 51 | KП→X0  | F0 |
| 13 | 2     | 02 | 32 | C/P              | 50 | 52 | FL2    | 58 |
| 14 | /—/   | 0L | 33 | X→P4             | 44 | 53 | 39     | 39 |
| 15 | X→P8  | 48 | 34 | 4                | 04 | 54 | 0      | 00 |
| 16 | KП→X8 | F8 | 35 | X→P2             | 42 | 55 | C/P    | 50 |
| 17 | ↔     | 14 | 36 | 1                | 01 | 56 | FL1    | 5L |
| 18 | P→X8  | 68 | 37 | 3                | 03 | 57 | 31     | 31 |
| 19 | —     | 11 | 38 | X→P0             | 40 | 58 | P→X5   | 65 |
|    | 2     | 02 | 39 | KП→XB↑           | FE | 59 | C/P    | 50 |

Примечание. Для МК-61 и МК-52 в программу необходимо внести изменения: 39.ПП 40.60 46.ПП 47.66 60.П→X0 61.X→ПЕ 62.FО 63.KП→ХЕ 64.П→X4 65.B/O 66.П→X0 67.X→ПЕ 68.FО 69.KХ→ПЕ 70.1 71.B/O.

Программа состоит из трех циклов: независимого вычисления в регистрах A, B, C, D кодов правильных ответов; внешнего по количеству введенных обучающимся ответов; внутреннего цикла, «вложенного» в него, для сопоставления фиксированного ответа обучающегося с правильными ответами.

Наберите восьмизначный код A и запустите счет, нажав клавиши В/О С/П, индикация π приглашает вводить двузначное число — ответ на любой вопрос: первая цифра — номер вопроса, вторая — номер ответа.

Введете ответ, нажмите С/П. Появление на индикаторе 1 означает, что ответ верный, 0 — неверный. Нажмите С/П, через 2 с на индикаторе появляется π — приглашение ко вводу следующего ответа и т. д. Если введены все ответы, то, чтобы узнать оценку, нажмите С/П.

Инструкция. Нажав клавиши В/О FПРГ, перейдите в режим программирования. Введите и отредактируйте программу, нажав клавиши FABГ, вернитесь в режим вычислений.

Контрольный пример. Коды правильных ответов на вопросы задания 12, 24, 36, 41. Коды ответов ученика: 12, 36, 41, 25.

Учитель заранее кодирует верные ответы:  $A_4 = 12 + 4 = 16$ ,  $A_1 = 24 + 1 = 25$ ,  $A_2 = 36 + 2 = 38$ ,  $A_3 = 41 + 3 = 44$ ,  $A = 16253844$ .

16253844 B/O C/P (20 с) «π», 12 C/P (20 с) «1», C/P (2 с) «π», 36 C/P (10 с) «1», C/P (2 с) «π», 41 C/P (7 с) «1», C/P (2 с) «π», 25 C/P (14 с) «0». C/P (2 с) «3» — оценка учащегося за три верных ответа из четырех

## Структура программы

- 01—09 : ввод исходных данных для вычисления оценки, построения внешнего цикла, формирования и записи кодов верных ответов.
- 10—32 : формирование правильных ответов на основе A и их запись в регистры РА—РД. Индикация π — это сигнал ко вводу ответа.
- 33 : запись ответа ученика в Р4.

34—38 : ввод данных для построения внутреннего цикла. Сравнение ответа ученика с очередным верным ответом (команда KП→XB↑ работает так же, как KП→X0, только без модификации).

39—42 : если коды равны, то содержимое Р5 увеличивается на единицу (44) и индицируется 1 (47, 48).

44—48 : засылка девятки в регистр, где находится верный ответ, совпадший с ответом обучающегося (45, 46). Это делается для того, чтобы нельзя было дважды ввести один и тот же верный ответ для накопления баллов. Использованный верный ответ стирается из памяти ПМК.

43, 49—52, если сравниваемые коды не равны (43), то содержимое Р0 уменьшается на единицу (51) и проверяется завершение исполнения внутренних циклов (52). Возможны два случая: цикл завершен, тогда на индикаторе 0 (54, 55), цикл не завершен (53, 39), тогда ответ обучающегося сравнивается с другим верным ответом. После завершения внутреннего цикла и запуска счета командой (С/П) совершается переход к команде 56.

56—59 : если в Р1 нуль, то индицируется содержимое Р5 — оценка учащегося. В противном случае управление переходит к командам 31, 32, на индикаторе π. Вводится следующий ответ обучающегося, пока Р1 очистится.

На УАЯ задача ставится в более общем виде — для n вопросов. При кодировании к коду первого верного ответа прибавляется n, второго — 1, третьего — 2 и т. д. К коду n-го ответа прибавляется n—1. Тогда 2n-значное ключевое число имеет вид  $A = A_n A_1 A_2 \dots A_{n-1}$ ; здесь  $A_i$  — двузначное число; i — число, прибавляемое к коду ответа.

Адрес алг САМОКОНТРОЛЬ (нат n, цел оц, вещ A)  
 команды арг n, A  
 рез оц  
 нач нат i, j, с, цел b,k, nat tab отв [1:n]  
 00—32 i := n; оц := 0  
 пока i ≥ 2  
 нц  
 A := A/100; b := A; отв [i] := (A - b)\*100 - i + 1; A := b; i := i - 1  
 кц  
 отв [1] := A - n  
 i := 1  
 пока i ≤ n  
 нц  
 ввод с; j := 1; k := 0  
 пока (j ≤ n) ∧ (k = 0)  
 нц  
 если с = отв [j]  
 то оц := оц + 1; k := 1; отв [j] := 9



```

нц
  если  $B = A[j]$ 
    то  $R := \text{«с}; T := 0$ 
  все
     $j := j + 1$ 
  кц
  если  $T = 1$ 
    то  $R := \text{«з»}$ 
  все
  сл [3:3] := R; выв сл
  если  $A = R$ 
    то выв «вы написали верно»; оц := оц*10 + 5
    иначе выв «вы сделали ошибку»; оц := оц*10 + 2
  все
кц
кон

```

### ЗАДАНИЯ

1. Вы учитель (математики, физики, литературы и т. д.). Придумайте несколько контрольных заданий по своему предмету, проверьте для них работу программы «Самоконтроль».

2. Пусть числовые коды правильных ответов вводятся в ПМК учителем, тогда становится ненужной процедура кодирования в программе «Самоконтроль». Упростите программу.

Форма ввода исходных данных:  $a\ B\dagger\ b\ B\dagger\ c\ B\dagger\ d\ B/O\ C/P$ . Здесь  $a, b, c, d$  — двузначные коды правильных ответов на все четыре вопросы. После этого ученик вводит свой ответ.

3. Если программу «Самоконтроль» использовать достаточно долго, то учащиеся несомненно обнаружат, что ПМК хранит правильные ответы на вопросы в регистрах А—Д. Интересно составить программу так, чтобы верные ответы вычислялись непосредственно перед сравнением с ответом учащегося, но не хранились в регистрах.

4. Алгоритм «Самоконтроль» на УАЯ построен с помощью пока-циклов. Замените их циклами с параметром.

5. В программе «Приставки» не предусмотрен случай, когда первая буква корня гласная (в приставке пишется ə). Включите в программу этот случай.

6. Правописание has и have в английском языке\*. В английском языке только глаголы настоящего времени, единственного числа, третьего лица оканчиваются на s. На основе программы «Приставки» составьте программу «Глагол». Можно предложить следующий вариант кодирования исходных данных и результата:

$$v(\text{время глагола}) = \begin{cases} 1, \text{ если глагол настоящего времени,} \\ 2 \text{ в противном случае;} \end{cases}$$

$$u(\text{число глагола}) = \begin{cases} 1, \text{ если оно единственное,} \\ 2, \text{ если множественное;} \end{cases}$$

$$l(\text{лицо глагола}) = \begin{cases} 1, \text{ если глагол первого лица,} \\ 2, \text{ если второго лица,} \\ 3, \text{ если третьего лица,} \end{cases}$$

$$R(\text{окончание}) = \begin{cases} 19, \text{ если окончание } s \text{ (19-я буква английского алфавита),} \\ 22, \text{ если окончание } ve \text{ (v — 22-я буква алфавита).} \end{cases}$$

\* Составьте программы типа «Консультант» на основе всех программ, рассмотренных в этом параграфе. Задача сформулирована учителем М. И. Данилич.

Данные вводятся в ПМК в виде трехразрядного числа, где цифры означают соответственно время — число — лицо.

7. Программа «Консультант». ПМК способен работать в качестве репетитора: обучающийся вводит вопрос, ПМК отвечает. Например, в программе «Самоконтроль» вводится 2, индицируется 6. Это значит, что верный ответ на второй вопрос записан под номером шесть.

### 4.6. ЛОГИЧЕСКИЕ ФУНКЦИИ И ПРОГРАММЫ КОНТРОЛЯ ЗНАНИЙ

Для программирования самоконтроля удобно использовать ПМК «Электроника МК-61» и «Электроника МК-52», в языке которых есть команды вычисления логических функций. Рассмотрим, как их можно использовать на конкретном примере. Контрольное задание состоит из семи вопросов, каждому из которых соответствует несколько ответов. В каждой группе содержится не более семи ответов, причем один из них обязательно верный. Из номеров верных ответов ученик должен составить и ввести в ПМК многозначное число.

Инструкция для учителя. Составьте число: первая цифра числа — номер правильного ответа на первый вопрос, вторая — на второй и т. д. Закодируйте полученное число: от первой цифры отнимите единицу, ко второй и третьей прибавьте единицу, от четвертой отнимите единицу, к пятой и шестой прибавьте единицу, от седьмой отнимите единицу. Если цифр, т. е. вопросов, меньше семи, то кодирование прервается ранее.

Постройте из полученных цифр в порядке их следования многозначное число и припишите впереди его любую цифру, отличную от нуля. Образовавшееся многозначное число можно сообщить обучающемуся наряду с заданием, программой и инструкцией для обучающегося.

Для кодирования удобно пользоваться следующей таблицей:

| Число вопросов           | 7      | 6     | 5    | 4   | 3  | 2 | 1 |
|--------------------------|--------|-------|------|-----|----|---|---|
| Вычитаемая константа $C$ | 890891 | 89089 | 8909 | 891 | 89 | 9 | 1 |

Поясним, как получены константы. Пусть в задании семь вопросов. От семизначного числа отнимается 1001001, и к нему прибавляется 110110. В итоге из числа вычитается 890891. Таким образом, каждая следующая константа получается делением предыдущей на 10 с округлением до целого. Достаточно, следовательно, помнить значение  $C$  для семи вопросов. Пусть ваше число 6|3125431, оно семизначно: цифра, отделенная чертой, не учитывается. Следуя таблице, вычтите из числа 890891. Разность 62234540 — это код правильного ответа. Для числа 3|15867 (оно пятизначно) код ответа находится как 315867—8909=306958.

## Алгоритм «Контроль».

1. Ввести в ПМК  $A$  — семизначное число, составленное из номеров ответов, выбранных учеником, и  $K$  — восьмизначное число, закодированный верный ответ.

2. Чтобы расшифровать ответ, к значению  $K$  прибавляется константа  $C$ .

3. Число  $A$  сложить с  $10^7$  (подготовка  $A$  к сложению с  $K$  по модулю 2). Это связано с особенностями устройства ПМК — в первом разряде  $A$  должна стоять любая значащая цифра, например 1.

4. Результаты выполнения указаний 2 и 3 сложить по модулю 2 (обозначается  $\oplus$ ). Итогом является восьмизначное число, в котором первая цифра 8 отделена десятичной точкой и является незначащей.

5. Число нулей в логической сумме равно числу правильных ответов, позиции нулей указывают вопросы, на которые даны верные ответы.

Если полученная сумма заканчивается нулями, то они не индицируются, так как находятся в конце числа после запятой. С другой стороны, не индицируются также шестнадцатеричные знаки 15 (— пробел). Однако на этот раз все обстоит благополучно: в нашем случае число 15 в результате  $\oplus$  получиться не может, поскольку цифры в  $A$  и  $K$  не превышают семи и в шестнадцатеричной записи начинаются с нуля.

### Программа КОНТРОЛЬ-М

|    |      |    |    |                  |    |    |      |    |
|----|------|----|----|------------------|----|----|------|----|
| 00 | P→X0 | 60 | 03 | 7                | 07 | 06 | K⊕   | 39 |
| 01 | +    | 19 | 04 | F10 <sup>x</sup> | 15 | 07 | X→ПA | 4— |
| 02 | ↔    | 14 | 05 | +                | 10 | 08 | C/P  | 50 |

**Инструкция.** Вы получили карточку с вопросами и ответами, каждому вопросу соответствует только один верный ответ. Кроме того, в задании имеется код  $K$ . Составьте число  $A$  из номеров правильных по вашему мнению ответов. Введите в ПМК программу и вернитесь в режим вычислений. Введите:  $A$  В↑  $K$  В/О С/П. Через 2 с индицируется число. В этом числе нули указывают, на какие вопросы вы ответили правильно. Количество нулей есть число верных ответов. Поясним инструкцию конкретным примером.

**Контрольное задание.** Семь вопросов;  $C=890891$ ,  $K=62234540$ ,  $A=2134435$ ; учащийся считает, что правильный ответ на первый вопрос стоит под номером 2, на второй — 1 и т. д. Учитель ввел код 890891 X→П0, ввод учащегося: 2134435 В↑ 62234540 В/О С/П (2 с) «8,1011004». Верные ответы даны на вопросы 2, 5, 6.

После запятой должно быть ровно семь цифр. Отсутствующие цифры соответствуют верным ответам. 2526431 В↑ 32234540 В/О С/П (2 с) «8,1403». Правильные ответы даны на вопросы 3, 5, 6, 7.

## ЗАДАНИЯ

1. Если число вопросов в задании меньше семи, то в Р0 записываются соответствующие константы. Составьте программы, рассчитанные на шесть, пять, четыре, три вопроса.

2. Значение  $C$  вводят учитель: если поручить это учащемуся, то процедура кодирования теряет смысл. Постройте вариант программы, в котором константа  $C$  вводится самой программой, причем вычисляется как промежуточный результат достаточно сложного алгебраического выражения. Удобно использовать косвенную адресацию.

3. Можно предложить другие подходы к кодированию, например к номеру правильного ответа на вопрос прибавляют остаток от деления цифры номера на 3. Пусть правильный ответ 6154327, тогда к цифрам прибавляются соответственно 0, 1, 2, 1, 0, 2 и 1. В ПМК будет введено 6275348. Первая цифра не изменилась: остаток от деления 6 на 3 равен нулю. Пятерка превратилась в семерку, так как остаток от деления 5 на 3 равен 2. Переформулируйте инструкцию для учителя и внесите изменения в программу.

4. Задание для студентов технических специальностей. Составьте контрольную работу на тему «Решение дифференциальных уравнений первого порядка». Возможный список ответов: 1) уравнение с разделяющимися переменными, 2) линейное уравнение или уравнение Бернуlli, 3) однородное уравнение, 4) уравнение приводящееся к однородному, 5) уравнение в полных дифференциалах, 6) уравнение с интегрирующим множителем, зависящим только от  $x$ , 7) уравнение с интегрирующим множителем, зависящим только от  $y$ . В контрольной работе в любой последовательности предлагаются для распознавания семь уравнений.

5. Задание на классификацию. В обучении нередко приходится проверять принадлежность объекта одному или нескольким классам. В этом случае на один вопрос возможно получить несколько верных ответов. Условимся пользоваться цифрой 1, если объект принадлежит данному классу, и цифрой 0 в противном случае. Тогда каждый объект относительно заданных классов можно описать многозначным (по числу классов) числом, состоящим из единиц и нулей, т. е. двоичным числом. По аналогии с нашей программой придумайте систему кодирования и составьте программу «Классификация». Подготовьте контрольные задания:

5.1. Функции. Список возможных ответов: 1) четная функция, 2) нечетная функция, 3) монотонная функция, 4) непрерывная функция, 5) линейная функция, 6) квадратичная функция, 7) функция, имеющая асимптоты. Контрольная может включать произвольное число функций, которые требуется распознавать.

5.2. Числа. Список ответов: 1) четное число, 2) нечетное число, 3) простое число, 4) натуральное число, 5) целое число, 6) рациональное число, 7) действительное число.

6. Дополните программу «Классификация» из задания процедурой оценки ответа по пятибалльной системе. Для этого удобно в цикле найти все цифры полученного результата, каждую из них вычесть из единицы и сложить полученные разности.

## Глава 5

### ЗАНИМАТЕЛЬНЫЕ ЗАДАЧИ

#### 5.1. РАЗМЕН БЕЗ СДАЧИ

В кассе имеются денежные купюры достоинством в 10 руб. и 3 руб. Выдайте без сдачи зарплату 237 руб.

Составим программу для решения подобных задач, записав условие в общем виде.

Необходимо разменять без сдачи с рублей с помощью денежных купюр достоинством  $a$  и  $b$  рублей,  $a > b$ . Обозначим  $x$  и  $y$  — число купюр достоинством  $a$  и  $b$ , тогда задача сводится к решению уравнения  $ax + by = c$  в целых неотрицательных числах.

В качестве первой пробы примем  $x = [c/a]$ . Если  $y = (c - ax)/b$  целое, тогда задача решена. В противном случае значение  $x$  уменьшается на 1, процедура повторяется и т. д. Если решение существует, оно будет обнаружено в процессе перебора.

### Программа РАЗМЕН

Распределение памяти

P0 — число купюр достоинства  $a$ , удовлетворяющее условию;

P1 — числа купюр достоинства  $b$ , удовлетворяющее условию;

PA —  $a$  — величина (достоинство) большей купюры;

PB —  $b$  — величина меньшей купюры;

PC — сумма денег, которую необходимо набрать данными купюрами.

|    |       |    |    |      |    |    |      |    |
|----|-------|----|----|------|----|----|------|----|
| 00 | X→ПС  | 4C | 12 | П→Х0 | 60 | 24 | ×    | 12 |
| 01 | F0    | 25 | 13 | П→ХА | 6  | 25 | Fsin | 1C |
| 02 | X→ПВ  | 4L | 14 | ×    | 12 | 26 | FX=0 | 5E |
| 03 | F0    | 25 | 15 | П→ХС | 6C | 27 | 32   | 32 |
| 04 | X→ПА  | 4  | 16 | ↔    | 14 | 28 | П→Х0 | 60 |
| 05 | П→ХС  | 6C | 17 | —    | 11 | 29 | C/P  | 50 |
| 06 | ↔     | 14 | 18 | П→ХВ | 6L | 30 | П→Х1 | 61 |
| 07 | :     | 13 | 19 | :    | 13 | 31 | C/P  | 50 |
| 08 | I     | 01 | 20 | X→ПИ | 41 | 32 | П→Х0 | 60 |
| 09 | +     | 10 | 21 | I    | 01 | 33 | FX=0 | 5E |
| 10 | X→П0  | 40 | 22 | 8    | 08 | 34 | 11   | 11 |
| 11 | KП→Х0 | G0 | 23 | 0    | 00 | 35 | K—   | 27 |

Инструкция.  $a \uparrow b \uparrow c \uparrow$  В/О С/П. (Переключатель углов на Г.) При первом останове индицируется значение  $x$  — число купюр достоинством  $a$ , после нажатия С/П  $y$  — число купюр достоинством  $b$ . Если задача не имеет решения, индицируется сигнал ЕГГОГ.

Контрольные примеры.

1. Выплатить 281 руб. денежными купюрами достоинством в 5 руб. и 3 руб.

5  $\uparrow$  3  $\uparrow$  281 В/О С/П. (Переключатель в положении Г.) (17 с)  
«55» С/П (1 с) «2». Ответ: 55 купюр по 5 руб. и 2 по 3 руб.

2. Отсчитать 134 руб. купюрами достоинством 25 руб. и 10 руб.

25  $\uparrow$  10  $\uparrow$  134 В/О С/П. (Переключатель в положении Г.) (54 с)  
«ЕГГОГ» — решения нет.

### Структура программы

00—04 : запись данных в соответствующие регистры памяти.

05—12 : вычисление первого допущения  $[c/a]$  и запись его в P0.

13—20 : вычисление  $y = (c - ax)/b$ .

21—26 : проверка, является ли  $y$  целым числом.

28—31 :  $y$  — целое, индицируются значения  $x$  и  $y$ .

27, 32,

33, 35 : индикация сигнала ЕГГОГ, если  $y$  не целое, а  $x=0$ .

34, 11

и т. д. : повторение цикла вычислений при  $y$  не целом, а  $x>0$ .

### ЗАДАНИЯ

1. Выплатить 253 руб. 50-рублевыми купюрами, если получатель располагает для сдачи достаточным числом трехрублевок. Сделайте расчет по программе. Ответ: выдается 8 купюр по 50 руб., сдача 49 купюр по 3 руб.

Указание. Уравнение  $ax - by = c$ ;  $y = (ax - c)/b$ . По сравнению с задачей «Размен без сдачи» несколько изменяется алгоритм. Первая гипотеза  $x = [c/a] + 1$ , с каждой следующей пробой  $x$  увеличивается на 1. Не забудьте изменить в связи с этим в предыдущей программе Р1, например, на Р4.

2. Через один кран вода вливается в бак со скоростью  $a [л/с]$  ( $a > b$ ). Сколько секунд должен быть открыт каждый кран, чтобы набрать в бак  $c$  литров воды? Составьте программу для автомата, который открывает и закрывает кран. Убедитесь с помощью программы, что при  $a=12$ ,  $b=9$  невозможно  $c=16$ , если кран можно переключать только раз в секунду.

Пример.  $a=5$ ,  $b=3$ ,  $c=184$ . Ответ: 38 с открыт 1-й кран, 2 с — 2-й. Автомат управляет работой на основе программы задания 1: первые 2 с открыты оба крана, затем 2-й кран закрывается; через 36 с закрывается 1-й кран.

Нетрудно заметить следующую закономерность. Наибольший общий делитель  $a$  и  $b$  является делителем  $c$ . В противном случае задача не имеет решения.

3. С помощью сосудов емкостью  $a[л]$  и  $b[л]$  ( $a > b$ ) набрать в большой сосуд  $c[л]$  воды,  $c < a$ .

Замечание. Задачи такого типа решаются на основе программы задания 1: с помощью 1-го сосуда вода наливается, 2-го — выливается. Можно предложить другой независимый алгоритм решения задачи. Наполняется больший сосуд, вода переливается в меньший до его наполнения и выливается из него. Это повторяется до тех пор, пока больший сосуд не опорожнится. Затем его снова наполняют и т. д. Если задача имеет решение, то на каком-то этапе в большем сосуде окажется  $c[л]$  воды. Если с какого-то момента остатки в большем сосуде начнут повторяться, то решения нет.

Постройте программу для ПМК на основе сформулированного алгоритма.

Пример.  $a=5$ ,  $b=3$ ,  $c=1$ . Ниже указано количество воды в сосудах в процессе решения задачи:

|           |                 |
|-----------|-----------------|
| Сосуд $a$ | 5 2 2 0 5 4 4 1 |
| Сосуд $b$ | 0 3 0 2 2 3 0 3 |

4. В классе выписывают три журнала: «Квант», «Техника молодежи», «Наука и жизнь», общее количество экземпляров меньше  $c$ . Количество подписчиков на «Технику — молодежи» в пять раз меньше, чем на «Науку и жизнь». Если число выписываемых «Квантов» увеличить в четыре раза, то их станет в 21 больше, чем число журналов «Наука и жизнь». Сколько журналов каждого наименования выписывают в классе?

Составьте программу, которая выдает все решения задачи для любого заданного  $c$ .

Обозначения:  $x$ ,  $y$ ,  $z$  — количество выписываемых журналов «Квант», «Техника — молодежи», «Наука и жизнь».

$$x + y + z < c \rightarrow x + 6y < c, \quad (1)$$

$$z = 5y \rightarrow z = 5y, \quad (2)$$

$$4x - z = 21 \rightarrow x = (21 + 5y)/4, \quad (3)$$

$y$  нечетно, иначе  $x$  не будет целым числом [см. (1)]. Значения  $y$  ограничены условием (2). На этом основан следующий алгоритм.

Генерируются значения  $y=1, 3, 5, \dots$ . Для каждого вычисляется соответствующее значение  $x$  по (1). Далее проверяется (2). Если неравенство не удовлетворено, то решений нет, точнее, других решений нет: при увеличении  $y$  неравенство тем более не выполняется.

Пусть условие (2) удовлетворяется, тогда следует проверить, будет ли вычисленное значение  $x$  целым. Если да, то индицируются значения  $x, y, z=5y$ . В любом случае  $y$  увеличивается на 2, и цикл вычислений повторяется.

Распределение памяти ПМК: значения  $x, y, z$  хранятся в Р4, Р5, Р6. По программе решите задачу для  $c=138$ . Ответы:  $x=9, y=3, z=15; 14, 7, 35; 19, 11, 55; 24, 15, 75$ .

5. Число двухкомнатных квартир в доме в 4 раза больше числа однокомнатных; число трехкомнатных кратно числу однокомнатных. Если количество трехкомнатных квартир увеличить в 5 раз, то их станет на 22 больше, чем двухкомнатных. Сколько одно-, двух и трехкомнатных квартир в доме, если общее число квартир меньше  $c$ ?

Составьте программу, которая выдает все решения задачи при любом фиксированном  $c$ .

Обозначения: число одно-, двух- и трехкомнатных квартир  $x, y, z$ .

$$5z - y = 22 \rightarrow z = (22 + 4x)/5, \quad (1)$$

$$x + y + z < c \rightarrow 5x + z < c, \quad (2)$$

$$z \text{ кратно } x \rightarrow z \text{ кратно } x, \quad (3)$$

$$y = 4x \rightarrow y = 4x. \quad (4)$$

Нетрудно убедиться, что при  $x=2$   $z$  — целое число. Последующие значения  $x$ , для которых  $z$  целое, отличаются от предыдущих на 5: 2, 7, 12, ..., [см. (1)]. Теперь нетрудно сформулировать алгоритм.

Решите по программе задачу для  $c=160$ . Ответы:  $x=2, y=8, z=6, x=22, y=88, z=22$ .

6. На авиалинии между двумя городами летают самолеты трех типов. Каждый самолет первого, второго, третьего типа может принять на борт соответственно 230, 110, 40 пассажиров и одновременно 27, 12, 5 контейнеров. Все самолеты линии могут принять на борт одновременно 760 пассажиров и 88 контейнеров. Найти число участвующих в перевозках самолетов каждого типа, если их общее количество менее  $c$ .

Составьте программу для решения задачи.

Обозначив число самолетов первого, второго и третьего типа  $x, y, z$ , запишем условие задачи

$$230x + 110y + 40z = 760 \rightarrow y = 4 - x, \quad (1)$$

$$27x + 12y + 5z = 88 \rightarrow z = 8 - 3x, \quad (2)$$

$$x + y + z < c \rightarrow z < c - 4, \quad (3)$$

Решите по программе задачу для  $c=25$ . Ответы:  $x=2, y=2, z=2; 1, 3, 5; 0, 4, 8$ .

7. Найти все двузначные числа, которые при делении на произведение своих цифр дают в частном  $q$  и в остатке  $r$ .  $x$  — цифра десятков,  $y$  — единица, число  $10x+y$ .

Алгоритм. Строятся значения  $y$  от 9 до 1 с шагом  $-1$ . По формуле  $x = (r-y)/(10-qy)$  вычисляются соответствующие значения  $x$ . Точнее, после нахождения знаменателя  $10-qy$  проверяется, не обращается ли он в нуль. Если да, то значение  $y$  уменьшается на 1, и вычисления повторяются в цикле, управляемом счетчиком с начальным значением 9. В противном случае проверяется, будет ли остаток  $r$  меньше делителя  $xy$ . Если это условие выполнено, а  $x$  — положительное целое число, то значения  $x$  и  $y$  индицируются в качестве цифр искомого числа. В любом случае происходит обращение к счетчику циклов, который проверяет окончание решения. Индикация  $\pi$  означает, что задача решена.

Распределение памяти. Остаток  $r$  хранится в Р4, частное  $q$  — в Р5.

Инструкция. Начальный ввод:  $q \rightarrow \text{П} \rightarrow \text{С/П}$ . После каждого нажатия С/П отдаём команду продолжить вычисления. Пара выведенных друг за другом чисел является первой и второй цифрами искомого числа.

8. Рыбаки. Три товарища наловили рыбы и, уставшие, но довольные, тут же на берегу уснули у догощающего костра. Один из них проснулся рано утром и заторопился домой. Посчитав число рыб, он обнаружил, что оно не делится на три. Чтобы никого не обидеть, он выпустил лишнюю рыбину в реку и, взяв свою долю улова, ушел домой. Когда проснулся второй, он, не сомневаясь, что его товарищи спят, взял свою треть, выпустив как и предыдущий, одну оставшуюся рыбку в воду, и ушел. То же самое сделал последний, не подозревая, что рыба уже поделена и товарищи ушли. Сколько всего было рыбы? Для решения задачи составьте программу на основе следующего алгоритма:

1. Высказывается гипотеза о числе рыб.
2. Из числа вычитается единица.
3. Если разность не кратна 3, то возврат к указанию 1.
4. Если разность кратна 3, то в качестве очередного числа берется  $2/3$  этой разности. Возврат к указанию 2.

Для каждой гипотезы процедура повторяется не более 3 раз. Если гипотеза выдержала «трехкратное испытание», то она принимается в качестве решения.

В качестве первой гипотезы удобно взять  $x=4$ , тогда условие указания 4 выполнено по крайней мере один раз. Из этих же соображений целесообразно выбирать последующие значения  $x$  с интервалом в 3 единицы, т. е. 7, 10, 13, ... Далее, если  $x=a$  есть решение задачи, то  $x=a+27$  также решение (от прибавления числа 27, т. е.  $3^3$ , выполнение условий в указаниях 3 и 4 не изменится).

9.  $a, b, c$  — целые положительные числа. Составьте программу для вычисления целых неотрицательных  $x, y$ , удовлетворяющих условию  $ax+by=c$ .

Переменной  $u$  присваиваются последовательно значения 0, 1, 2, ... По формуле  $x = (c - by)/a$  вычисляются соответствующие значения  $x$ . Если окажется  $c - by < 0$ , то уравнение не имеет решений, кроме ранее найденных. Если  $x$  не целое, то  $y$  увеличиваются на 1. Появление ЕГГОГ означает, что других решений нет. При останове индицируется значение  $x$ , после нажатия С/П —  $y$ . Следующее нажатие С/П запускает продолжение вычислений. [см. 12, ч. II, задача 32, с. 102].

## ОТВЕТЫ

4. Переключатель в положении Г. Начальный ввод:  $c \rightarrow \text{В/О} \rightarrow \text{С/П}$ . После каждого останова считывается индицируемое число. Для продолжения вычислений нажать клавишу С/П. Три числа, поочередно появляющиеся на индикаторе, это удовлетворяющий условию набор значений  $x, y, z$ . Индикация восьми знаков числа  $\pi$  означает окончание решения.

|    |                             |    |    |                             |    |    |                             |    |
|----|-----------------------------|----|----|-----------------------------|----|----|-----------------------------|----|
| 00 | $X \rightarrow \text{ПС}$   | 4C | 14 | $\times$                    | 12 | 28 | $\Pi \rightarrow X_4$       | 64 |
| 01 | 1                           | 01 | 15 | +                           | 10 | 29 | $\text{С/П}$                | 50 |
| 02 | $X \rightarrow \text{П5}$   | 45 | 16 | $\Pi \rightarrow \text{ХС}$ | 6C | 30 | $\Pi \rightarrow X_5$       | 65 |
| 03 | $\Pi \rightarrow \text{Х5}$ | 65 | 17 | $-$                         | 11 | 31 | $\text{С/П}$                | 50 |
| 04 | 5                           | 05 | 18 | $FX < 0$                    | 5C | 32 | 5                           | 05 |
| 05 | $\times$                    | 12 | 19 | 39                          | 39 | 33 | $\times$                    | 12 |
| 06 | 2                           | 02 | 20 | $\Pi \rightarrow X_4$       | 64 | 34 | $\text{С/П}$                | 50 |
| 07 | 1                           | 01 | 21 | 1                           | 01 | 35 | $\text{КП} \rightarrow X_5$ | Г5 |
| 08 | +                           | 10 | 22 | 8                           | 08 | 36 | $\text{КП} \rightarrow X_5$ | Г5 |
| 09 | 4                           | 04 | 23 | 0                           | 00 | 37 | $\text{БП}$                 | 51 |
| 10 | :                           | 13 | 24 | $\times$                    | 12 | 38 | 03                          | 03 |
| 11 | $X \rightarrow \text{П4}$   | 44 | 25 | $F \sin$                    | 1C | 39 | $F \pi$                     | 20 |
| 12 | $\Pi \rightarrow \text{Х5}$ | 65 | 26 | $FX = 0$                    | 5E | 40 | $\text{С/П}$                | 50 |
| 13 | 6                           | 06 | 27 | 35                          | 35 |    |                             |    |

Контрольный пример. 138 В/О С/П «9» С/П «3» С/П «15»; С/П «14» С/П «7» С/П «35»; С/П «19» С/П «11» С/П «55», С/П «24» С/П «15» С/П «75».

5. Инструкция та же, что в задаче 4.

|    |      |    |    |      |    |    |       |    |
|----|------|----|----|------|----|----|-------|----|
| 00 | X→ПС | 4C | 15 | X→П6 | 46 | 30 | X     | 12 |
| 01 | 3    | 03 | 16 | П→Х4 | 64 | 31 | F sin | 1C |
| 02 | /—/  | 0L | 17 | 5    | 05 | 32 | FX=0  | 5E |
| 03 | X→П4 | 44 | 18 | ×    | 12 | 33 | 04    | 04 |
| 04 | П→Х4 | 64 | 19 | +    | 10 | 34 | П→Х4  | 64 |
| 05 | 5    | 05 | 20 | П→ХС | 6C | 35 | C/П   | 50 |
| 06 | +    | 10 | 21 | —    | 11 | 36 | 4     | 04 |
| 07 | X→П4 | 44 | 22 | FX<0 | 5C | 37 | ×     | 12 |
| 08 | 4    | 04 | 23 | 43   | 43 | 38 | C/П   | 50 |
| 09 | ×    | 12 | 24 | П→Х6 | 66 | 39 | П→Х6  | 66 |
| 10 | 2    | 02 | 25 | П→Х4 | 64 | 40 | C/П   | 50 |
| 11 | 2    | 02 | 26 | :    | 13 | 41 | БП    | 51 |
| 12 | +    | 10 | 27 | 1    | 01 | 42 | 04    | 04 |
| 13 | 5    | 05 | 28 | 8    | 08 | 43 | F π   | 20 |
| 14 | :    | 13 | 29 | 0    | 00 | 44 | C/П   | 50 |

Контрольный пример. 160 В/О С/П. Первое решение: «2» С/П «8» С/П «6». Второе решение: С/П «22» С/П «88» С/П «22». Других решений нет.

6. Инструкция та же, что в задаче 4.

|    |      |    |    |      |    |    |       |    |
|----|------|----|----|------|----|----|-------|----|
| 00 | X→ПС | 4C | 10 | 27   | 27 | 20 | —     | 11 |
| 01 | CX   | 0Г | 11 | П→ХС | 6C | 21 | C/П   | 50 |
| 02 | X→П4 | 44 | 12 | 4    | 04 | 22 | П→Х6  | 66 |
| 03 | 8    | 08 | 13 | —    | 11 | 23 | C/П   | 50 |
| 04 | П→Х4 | 64 | 14 | —    | 11 | 24 | KП→Х4 | Г4 |
| 05 | 3    | 03 | 15 | FX<0 | 5C | 25 | БП    | 51 |
| 06 | ×    | 12 | 16 | 24   | 24 | 26 | 03    | 03 |
| 07 | —    | 11 | 17 | 4    | 04 | 27 | F π   | 20 |
| 08 | X→П6 | 46 | 18 | П→Х4 | 64 | 28 | C/П   | 50 |
| 09 | FX≥0 | 59 | 19 | C/П  | 50 |    |       |    |

Контрольный пример. 25 В/О С/П. Первое решение: «0» С/П «4» С/П «8». Второе решение: С/П «1» С/П «3» С/П «5». Третье решение: С/П «2» С/П «2» С/П «2». Других решений нет.

7. q X→ПВ r В/О С/П. Переключатель в положении Г.

|    |      |    |    |      |    |    |       |    |
|----|------|----|----|------|----|----|-------|----|
| 00 | X→ПА | 4— | 15 | X→П4 | 44 | 29 | 8     | 08 |
| 01 | 9    | 09 | 16 | П→Х0 | 60 | 30 | 0     | 00 |
| 02 | X→П0 | 40 | 17 | ×    | 12 | 31 | ×     | 12 |
| 03 | П→ХА | 6— | 18 | П→ХА | 6— | 32 | F sin | 1C |
| 04 | П→Х0 | 60 | 19 | ↔    | 14 | 33 | FX=0  | 5E |
| 05 | —    | 11 | 20 | —    | 11 | 34 | 39    | 39 |
| 06 | 1    | 01 | 21 | FX<0 | 5C | 35 | П→Х4  | 64 |
| 07 | 0    | 00 | 22 | 39   | 39 | 36 | C/П   | 50 |
| 08 | П→Х0 | 60 | 23 | П→Х4 | 64 | 37 | П→Х0  | 60 |
| 09 | П→ХВ | 6L | 24 | /—/  | 0L | 38 | C/П   | 50 |
| 10 | ×    | 12 | 25 | FX<0 | 5C | 39 | FL0   | 5Г |
| 11 | —    | 11 | 26 | 39   | 39 | 40 |       |    |
| 12 | FX≠0 | 57 | 27 | П→Х4 | 64 | 41 | F π   | 20 |
| 13 | 39   | 39 | 28 | 1    | 01 | 42 | C/П   | 50 |
| 14 | :    | 13 |    |      |    |    |       |    |

Контрольный пример. q=4, r=1. 4 X→ПВ 1 В/О С/П «1», С/П «3», С/П «π» — задача решена. Ответ: 13.

С помощью программы нетрудно убедиться, что задача имеет решение при  $q=2$  (19 для  $r=1$ ; 18 и 26 для  $r=2$ ; 17 для  $r=3$ ; 16 для  $r=4$ ); при  $q=3$  (14 для  $r=2$ ; 23 для  $r=5$ ; 33 для  $r=6$ ; 43 для  $r=7$ ; 53 для  $r=8$ ; 63 для  $r=9$ ); при  $q=4$  (13 для  $r=1$ ). Других решений нет.

#### 8. Программа РЫБАКИ

|    |      |    |    |       |    |    |      |    |
|----|------|----|----|-------|----|----|------|----|
| 00 | X→ПА | 4— | 11 | 2     | 02 | 21 | П→ХВ | 6L |
| 01 | 3    | 03 | 12 | ×     | 12 | 22 | FL0  | 5Г |
| 02 | X→П0 | 40 | 13 | X→ПВ  | 4L | 23 | 07   | 07 |
| 03 | П→ХА | 6— | 14 | 1     | 01 | 24 | П→ХА | 6— |
| 04 | 3    | 03 | 15 | 8     | 08 | 25 | C/П  | 50 |
| 05 | +    | 10 | 16 | 0     | 00 | 26 | 2    | 02 |
| 06 | X→ПА | 4— | 17 | ×     | 12 | 27 | 7    | 07 |
| 07 | 1    | 01 | 18 | F sin | 1C | 28 | +    | 10 |
| 08 | —    | 11 | 19 | FX=0  | 5E | 29 | БП   | 51 |
| 09 | 3    | 03 | 20 | 01    | 01 | 30 | 25   | 25 |
| 10 | :    | 13 |    |       |    |    |      |    |

Инструкция. 1 В/О С/П. Переключатель в положении Г. Примерно через 2 мин после нажатий 1 В/О С/П индицируется 25. После нажатий клавиши С/П на индикаторе другие решения.

Контрольные примеры. 1 В/О С/П (1 мин 52 с) «25» С/П (2 с) «52» С/П (2 с) «79» С/П (2 с) «106» и т. д.

Условимся наряду с рыбами рассматривать «антирыбы» ( $x < 0$ ); тогда одним из решений задачи будет  $x = -2$ , т. е. две «антирыбы». Действительно, выбросим одну рыбку, останется три «антирыбы». Первый взял свою «антирыбу», оставил две другие товарищам. То же самое поочередно сделали остальные. Такое решение задачи связывают с именем одного из создателей квантовой механики, автора теории антивещества П. Дирака.

Заменив в программе в командах 05 и 28 «+» на «—», через 20 с получим —2. Другие решения из мира «антирыб» можно получить, продолжая нажимать клавишу С/П.

#### 9. a B↑ b B↑ c B/O С/П. Переключатель в положении Г.

|    |      |    |    |      |    |    |       |    |
|----|------|----|----|------|----|----|-------|----|
| 00 | X→ПС | 4C | 11 | —    | 11 | 22 | Fsin  | 1C |
| 01 | FO   | 25 | 12 | FX<0 | 5C | 23 | FX=0  | 5E |
| 02 | X→ПВ | 4L | 13 | 15   | 15 | 24 | 29    | 29 |
| 03 | ↔    | 14 | 14 | K×   | 28 | 25 | П→ХД  | 6Г |
| 04 | X→ПА | 4— | 15 | П→ХА | 6— | 26 | C/П   | 50 |
| 05 | CX   | 0Г | 16 | :    | 13 | 27 | П→Х4  | 64 |
| 06 | X→П4 | 44 | 17 | X→ПД | 4Г | 28 | C/П   | 50 |
| 07 | П→ХС | 6C | 18 | 1    | 01 | 29 | KП→Х4 | Г4 |
| 08 | П→ХВ | 6L | 19 | 8    | 08 | 30 | БП    | 51 |
| 09 | П→Х4 | 64 | 20 | 0    | 00 | 31 | 07    | 07 |
| 10 | ×    | 12 | 21 | ×    | 12 |    |       |    |

Контрольный пример. a=3, b=4, c=8. 3 B↑ 4 B↑ 18 B/O С/П «6» С/П «0» С/П «2» С/П «3» С/П «ЕГГОГ». Результат:  $x_1=6$ ,  $y_1=4$ ;  $x_2=2$ ,  $y_2=3$ . Других решений нет.

#### 5.2. В МНОГОКВАРТИРНОМ ДОМЕ

Требуется срочно доставить телеграмму в квартиру № 325 нового 12-этажного дома на 432 квартиры с шестью подъездами, в котором еще не успели вывесить указатели квартир и нет почтовых ящиков.

Сформулируем задачу в общем виде. В  $m$ -этажном  $k$ -квартирном доме с  $n$  подъездами найти подъезд и этаж с номером  $T$ .

- Вычислить  $a = k/n$  — число квартир в подъезде и  $b = a/m = k/(nm)$  — число квартир на одном этаже в подъезде.
- Найти целую часть и остаток от деления  $T$  на  $a$ .
- Если остаток равен нулю, то целая часть — номер подъезда, а номер этажа  $m$ . Конец. В противном случае переходят к указанию 4.

- Номер подъезда равен целой части, увеличенной на 1.
- Найти целую часть и остаток от деления предыдущего остатка на  $b$ .

6. Если остаток равен нулю, то целая часть — номер этажа. В противном случае номер этажа равен целой части, увеличенной на 1.

**Пример.** 120-квартирный 5-этажный дом с четырьмя подъездами. Найти квартиру 67;  $k=120$ ,  $m=5$ ,  $n=4$ ,  $T=67$ .  
 1.  $a=120/4=30$ ,  $b=30/5=6$  — в подъезде 30 квартир по 6 на каждом этаже.

2. Целая часть 2, остаток 7.

3. Остаток не 0.

4. Номер подъезда 3.

5. Целая часть 1, остаток 1.

6. Остаток не 0 — номер этажа 2.

Квартира 67 находится в третьем подъезде на втором этаже.

### Программа НОМЕР КВАРТИРЫ

Распределение памяти

P2, P3 — рабочие регистры;

PA —  $k/n$  — число квартир в подъезде;

PB —  $k/(nm)$  — число квартир на одном этаже в подъезде;

PC —  $T$  — номер искомой квартиры;

PD —  $m$  — число этажей в доме.

|    |      |    |    |       |    |    |      |    |
|----|------|----|----|-------|----|----|------|----|
| 00 | X→ПС | 4C | 13 | X→ПА  | 4— | 26 | X→ПЗ | 43 |
| 01 | F0   | 25 | 14 | П→ХС  | 6C | 27 | FX=0 | 5E |
| 02 | :    | 13 | 15 | П→ХА  | 6— | 28 | 33   | 33 |
| 03 | X→ПА | 4— | 16 | :     | 13 | 29 | П→Х2 | 62 |
| 04 | ↔    | 14 | 17 | 1     | 01 | 30 | C/P  | 50 |
| 05 | X→ПД | 4Г | 18 | +     | 10 | 31 | П→ХД | 6Г |
| 06 | :    | 13 | 19 | X→П2  | 42 | 32 | C/P  | 50 |
| 07 | X→ПВ | 4L | 20 | KП→Х2 | Г2 | 33 | П→Х2 | 62 |
| 08 | ПП   | 53 | 21 | П→ХС  | 6C | 34 | 1    | 01 |
| 09 | 14   | 14 | 22 | П→ХА  | 6— | 35 | +    | 10 |
| 10 | П→Х3 | 63 | 23 | П→Х2  | 62 | 36 | C/P  | 50 |
| 11 | X→ПС | 4C | 24 | X     | 12 | 37 | B/O  | 52 |
| 12 | П→ХВ | 6L | 25 | —     | 11 |    |      |    |

Инструкция.  $m$  В↑  $k$  В↑  $n$  В↑  $T$  В/O С/P (9 с) — индицируется номер подъезда, С/P — индицируется номер этажа.

Контрольные примеры.  $m=5$ ,  $k=120$ ,  $n=4$ ,  $T=67$ . 5 В↑ 120 В↑ 4 В↑ 67 В/O С/P (9 с) «3» С/P (9 с) «2» — третий подъезд, второй этаж.  
 В том же доме  $T=60$ . 5 В↑ 120 В↑ 4 В↑ 60 В/O С/P (9 с) «2» С/P (1 с) «5» — второй подъезд, пятый этаж.

### Структура программы

- 00—07 : формирование содержимого регистров А—Д.
- 08, 09,
- 14—35 : вычисление и индикация номера подъезда, номера этажа, если он самый верхний в доме ( $m$ -й).
- 10—13,
- 14—32 : вычисление номера этажа, если этаж не верхний.
- 14—26 : вычисление целой части и остатка от деления содержимого РС на содержимое РА.
- 27—37 : если остаток равен нулю, то в первом обращении к подпрограмме индицируется номер подъезда и этажа (команды 27, 29—32), в противном случае — только номер подъезда (команды 28, 33—36). Во втором обращении к подпрограмме индицируется номер этажа, если он не выведен в первом обращении.

```

алг НОМЕР КВАРТИРЫ (нат  $m$ ,  $k$ ,  $n$ ,  $T$ , ПОДЪЕЗД, ЭТАЖ)
    арг  $m$ ,  $k$ ,  $n$ ,  $T$ 
    рез ПОДЪЕЗД, ЭТАЖ
нач нат  $a$ ,  $b$ , цел  $q$ ,  $r$ 
     $a := k/n$ ;  $b := a/m$ ;  $q := [T/a]$ ;  $r := T - q*a$ 
    если  $r = 0$ 
        то ПОДЪЕЗД :=  $q$ ; ЭТАЖ :=  $m$ 
        иначе ПОДЪЕЗД :=  $q + 1$ ;  $q := [r/b]$ ;  $r := r - q*b$ 
        если  $r = 0$ 
            то ЭТАЖ :=  $q$ 
            иначе ЭТАЖ :=  $q + 1$ 
    все
    кон

```

### ЗАДАНИЯ

1. Решите задачу, если еще требуется доставить письма в квартиры № 214 и № 288 того же дома. Ответы: 5-й подъезд, 7-й этаж; 3-й подъезд, 12-й этаж; 4-й подъезд, 12-й этаж.

2. В задаче «Номер квартиры» отыскивается одно из решений в целых числах уравнения  $T=ax+by+r$  ( $a>b>r$ ). Теперь ясно, что мы столкнулись с простейшим вариантом задачи «Размен», рассмотренной в предыдущем параграфе. Нашу задачу можно сформулировать иначе, например так: разменять  $T$  руб. максимально возможным числом купюр достоинством  $a$ ; остаток — купюрами достоинства  $b$  и, наконец, второй остаток, если размен нацело невозможен, выдать рублями. Аналогично составьте программу для рассматриваемого типа задач на размен денег.

### 5.3. ГОЛОВОЛОМКА АВТОМОБИЛИСТА

Милиционер спрашивает у автомобилиста: «Вы не обратили внимания на номер только что проехавшего самосвала?» — «Номера я не запомнил, только заметил, что второе двузначное число получается из первого перестановкой цифр, а первое больше

второго на сумму цифр одного из них». Милиционер достал из кармана микрокалькулятор и, немного поколдовав над ним, сказал: «Спасибо».

Как ему удалось справиться с задачей?

Милиционер рассуждал так. Первая цифра первого двузначного числа несомненно больше второй, так как первое число больше второго. Пусть вторая цифра будет, например, 0, а первую начнем последовательно увеличивать на 1, начиная со значения, равного второй цифре. Каждое полученное двузначное число представим в условие задачи. Если оно не будет выполнено, то увеличим вторую цифру на единицу и повторим расчет. Через несколько циклов-итераций задача будет решена.

### Программа НОМЕР МАШИНЫ

#### Распределение памяти

P4, P5 — вторая и первая цифры первого двузначного числа, начальное значение второй цифры 0;  
P6 — первое число.

|    |                        |    |    |                       |    |    |                        |    |
|----|------------------------|----|----|-----------------------|----|----|------------------------|----|
| 00 | СХ                     | 0Г | 11 | $\Pi \rightarrow X_4$ | 64 | 22 | 31                     | 31 |
| 01 | $X \rightarrow P_4$    | 44 | 12 | ВП                    | 0С | 23 | $\Pi \rightarrow X_5$  | 65 |
| 02 | $\Pi \rightarrow X_4$  | 64 | 13 |                       | 1  | 01 | 24                     | 9  |
| 03 | $X \rightarrow P_5$    | 45 | 14 | $\Pi \rightarrow X_5$ | 65 | 25 | —                      | 11 |
| 04 | $K\Pi \rightarrow X_5$ | Г5 | 15 | +                     | 10 | 26 | $F_X = 0$              | 5E |
| 05 | $\Pi \rightarrow X_5$  | 65 | 16 | —                     | 11 | 27 | 04                     | 04 |
| 06 | ВП                     | 0C | 17 | $\Pi \rightarrow X_4$ | 64 | 28 | $K\Pi \rightarrow X_4$ | Г4 |
| 07 | 1                      | 01 | 18 | $\Pi \rightarrow X_5$ | 65 | 29 | БП                     | 51 |
| 08 | $\Pi \rightarrow X_4$  | 64 | 19 | +                     | 10 | 30 | 02                     | 02 |
| 09 | +                      | 10 | 20 | —                     | 11 | 31 | $\Pi \rightarrow X_6$  | 66 |
| 10 | $X \rightarrow P_6$    | 46 | 21 | $F_X \neq 0$          | 57 | 32 | С/П                    | 50 |

Инструкция. В/О С/П (3,5 мин) «54». Номер машины 5445.  
Проверка: 54—45=5+4.

### Структура программы

- 00, 01: засылка нуля в P4, в котором хранится вторая цифра: перебор начинается с фиксирования второй цифры значением 0.  
02—04: передача содержимого P4 в P5, увеличение содержимого P5 на 1.  
05—10: формирование первого двузначного числа и его хранение в P6.  
11—15: построение в RX второго двузначного числа.  
16—32: сравнение разности чисел с суммой цифр. В случае равенства индицируется первое число (команды 31, 32). В противном случае возможны два исхода: если первая цифра не 9, то она увеличивается на 1 (команда 04), и

далее процедура повторяется в цикле. Если же первая цифра 9, то вторая увеличивается на 1 (команда 28), и в соответствии с командами 29, 30 управление переходит к началу внешнего цикла передачи содержимого P4 в P5 (команды 02, 03); снова внутренний цикл и т. д.

Адреса команд алг НОМЕР МАШИНЫ (нат НОМЕР)  
рэз НОМЕР  
нач цел Ц2, nat Ц1, А, В  
Ц1:=1; Ц2:=0; А:=10; В:=1  
пока А=В Ц1+Ц2  
иц  
если Ц1<9  
то Ц1:=Ц1+1  
иначе Ц2:=Ц2+1; Ц1:=Ц2+1  
иц  
все  
А:=10\*Ц1+Ц2; В:=10\*Ц2+Ц1  
кц  
НОМЕР; = А\*100 + В  
кон

### ЗАДАНИЕ

Шестиклассник Илья возвел натуральное число в куб и записал результат в тетради. Когда он на следующий день заглянул в тетрадь, то оказалось, что последние три цифры числа залиты чернилами и разглядеть их невозможно. Самым скверным оказалось то, что нельзя было разобрать и основание куба. На бумаге можно было прочитать:  $* * - 19 * * *$ . Расстроенный Илюша обратился к отцу за помощью. «Сейчас разберемся», — сказал отец, — на этот случай у нас есть микрокалькулятор». Через несколько минут он полностью восстановил условие и ответ. Прежде всего отец вычислил  $22^3 = 10648$ . «Мало, — подумал он, — однако гадать больше не будем. Пусть ПМК прибавляет к числу 22 единицу, два и т. д., пока первые две цифры куба предполагаемого основания не образуют число 19. Тогда станут известны основание и сам куб».

Составьте соответствующую программу, решите задачу и постройте алгоритм на УAY.

Ответ: 19 683.

### 5.4. ВЫБОРЫ

Задача 1. *n* человек выбирают делегата на конференцию из восьми включенных в список. Проходит кандидат, набравший наибольшее число голосов. Если таких окажется несколько человек, то голосование повторяется.

Каждый голосующий нажимает клавишу ПМК, соответствующую номеру кандидата, которого он рекомендует.

Построить программу голосования, по которой в регистрах 1—8 подсчитывается число голосов, поданных соответственно за первого, второго и т. д. кандидатов из списка.

# Программа ВЫБОРЫ-1

|    |        |    |    |       |    |    |       |    |
|----|--------|----|----|-------|----|----|-------|----|
| 00 | X→ПД   | 4Г | 07 | П→ХД  | 6Г | 14 | KX→ПА | L— |
| 01 | 8      | 08 | 08 | X→П0  | 40 | 15 | FL0   | 5Г |
| 02 | X→П0   | 40 | 09 | C/P   | 50 | 16 | 09    | 09 |
| 03 | 0      | 00 | 10 | X→ПА  | 4— | 17 | Fп    | 20 |
| 04 | KX→ПВ↑ | LE | 11 | KП→ХА | Г— | 18 | C/P   | 50 |
| 05 | FL0    | 5Г | 12 | I     | 01 |    |       |    |
| 06 | 03     | 03 | 13 | +     | 10 |    |       |    |

**Инструкция.** Председатель собрания набирает на клавиатуре число голосующих  $n$  и, нажав В/О С/П, включает счет. Через 10 с ПМК останавливается на индикаторе  $n$ . Каждый голосующий набирает номер кандидата в списке, которому отдает предпочтение. Председатель нажимает клавишу С/П. По окончании голосования индицируются восемь знаков числа  $\pi$ . Проматывая содержимое Р1—Р8 (П→Х1, П→Х2, ..., П→Х8), можно узнать число голосов, поданных за каждого.

**Контрольный пример.**  $n=23$ . 23 В/О С/П (10 с) «23» 2 С/П (2 с) «1» 1 С/П (2 с) «1» 3 С/П, 3 С/П, 4 С/П, 3 С/П, 1 С/П, 2 С/П, 7 С/П, 8 С/П, 4 С/П, 6 С/П, 7 С/П, 3 С/П, 1 С/П, 2 С/П, 8 С/П, 6 С/П, 4 С/П, 3 С/П, 7 С/П, 8 С/П, 8 С/П «π» — голосование закончено. П→Х1 «3» — за первого голосовали трое; П→Х2 «3»; П→Х3 «5»; П→Х4 «3» П→Х5 «0»; П→Х6 «2»; П→Х7 «3»; П→Х8 «4». Большинством голосов избран третий кандидат. Бремя решения задачи 2—3 мин.

## Структура программы

00—06 : очистка регистров 1—8.

07—14 : прибавление единицы (одного голоса) к содержимому соответствующего регистра.

15, 16 : проверка окончания, возврат на повторение цикла, если задача не решена.

17, 18 : индикация  $\pi$ , когда задача решена.

**Замечание.** Задачу можно сформулировать и так. В звене каждый из восьми человек выбирает одного, с которым хотел бы дружить. Найти тайным голосованием самых уважаемых людей в звене.

Пусть ПМК выдает номер человека, набравшего наибольшее число голосов, а также соответствующее количество голосов.

Рекомендуется следующий способ индикации:  $m, k$ , где  $m$  — максимальное число полученных голосов;  $k$  — номер кандидата в списке. Например, число на индикаторе 6, 2 означает, что кандидат № 2 собрал максимальное число голосов — 6.

В программу «Выборы-1» внесены следующие изменения:

|    |        |    |    |      |    |    |      |    |
|----|--------|----|----|------|----|----|------|----|
| 17 | —7     | 07 | 26 | FX<0 | 5C | 35 | 1    | 01 |
| 18 | X→П0   | 40 | 27 | 32   | 32 | 36 | 0    | 00 |
| 19 | П→Х8   | 68 | 28 | FBX  | 0  | 37 | ×    | 12 |
| 20 | X→ПС   | 4C | 29 | X→ПС | 4C | 38 | P→ХВ | 6L |
| 21 | 8      | 08 | 30 | P→Х0 | 60 | 39 | +    | 10 |
| 22 | X→ПВ   | 4L | 31 | X→ПВ | 4L | 40 | 1    | 01 |
| 23 | П→ХС   | 6C | 32 | FL0  | 5Г | 41 | 0    | 00 |
| 24 | KП→ХВ↑ | ГЕ | 33 | 23   | 23 | 42 | :    | 13 |
| 25 | —      | 11 | 34 | P→ХС | 6C | 43 | C/P  | 50 |

В рассмотренном выше контрольном примере через 12 с после того, как все проголосовали, индицируется 5,3 — кандидат № 3 собрал максимальное число голосов — 5.

Нередко, например в психологических и социологических исследованиях, экспериментатору важно знать не только количество полученных человеком голосов, но и от кого они получены. С другой стороны, испытуемые заинтересованы в том, чтобы их выбор оставался неизвестным другим членам группы.

**Задача 2.** В комитете из восьми человек тайным голосованием избирается председатель. Каждый голосует только за одного. Построить программу «Выборы-2».

# Программа ВЫБОРЫ-2

|    |        |    |    |       |    |    |       |    |
|----|--------|----|----|-------|----|----|-------|----|
| 00 | CX     | 0Г | 12 | 0     | 00 | 24 | I     | 01 |
| 01 | 8      | 08 | 13 | :     | 13 | 25 | 0     | 00 |
| 02 | X→П0   | 40 | 14 | X→ПД  | 4Г | 26 | X     | 12 |
| 03 | 0      | 00 | 15 | KП→ХД | ГГ | 27 | P→ХД  | 6Г |
| 04 | KX→ПВ↑ | LE | 16 | Fо    | 25 | 28 | +     | 10 |
| 05 | FL0    | 5Г | 17 | P→ХД  | 6Г | 29 | KП→ПА | Л— |
| 06 | 03     | 03 | 18 | —     | 11 | 30 | FL0   | 5Г |
| 07 | 8      | 08 | 19 | 1     | 01 | 31 | 09    | 09 |
| 08 | X→П0   | 40 | 20 | 0     | 00 | 32 | Fп    | 20 |
| 09 | C/P    | 50 | 21 | X     | 12 | 33 | C/P   | 50 |
| 10 | B↑     | 0E | 22 | X→ПА  | 4— |    |       |    |
| 11 | I      | 01 | 23 | KП→ХА | Г— |    |       |    |

**Инструкция.** После нажатия В/О С/П через 10 с ПМК останавливается. Голосующий вводит двузначное число: первая цифра — номер голосующего, вторая — кого он выбирает. Например, ввод числа 23 означает, что второй отдал свой голос третьему. Нажав С/П, продолжаем счет. После останова голосует следующий и т. д. По окончании голосования индицируются восемь знаков числа  $\pi$ . Опросив регистры 1—8, выясняем, как прошло голосование. Пусть после нажатия П→Х3 на индикаторе 1245 — значит, третий член комитета рекомендован первым, вторым, четвертым и пятым членами.

**Пример.** В/О С/П 18 С/П, 23 С/П, 38 С/П, 48 С/П, 53 С/П, 68 С/П, 78 С/П, 83 С/П «π». П→Х1 «0», П→Х2 «0», П→Х3 «258», П→Х4 «0», П→Х5 «0», П→Х6 «0», П→Х7 «0», П→Х8 «13467». Наибольшее число голосов получил восьмой член комитета, за него голосовали первый, третий, четвертый, шестой, седьмой.

## Структура программы

00—06 : очистка регистров 1—8.

07, 08 : формирование счетчика для решения в цикле вопроса об окончании голосования.

09—22 : вычисление первой цифры введенного числа, ее запись в РД, расчет второй цифры, запоминание в РА.

23—29 : приписывание справа второй цифры введенного числа к содержимому соответствующего регистра и запоминание результата в этом же регистре.

30 : проверка окончания.  
 31—33 : повторение цикла ввода и обработки очередного двузначного числа, если задача не решена, и индикация π в противном случае.

## ЗАДАНИЯ

1. Преобразуйте программу «Выборы-1», чтобы она выводила на индикатор номера тех кандидатов, за которых голосовали больше половины избирателей. Если таких нет, индицируются восемь знаков числа π.
2. Постройте алгоритмы на УАЯ для обеих задач голосования.
3. Внесите изменения в программы, чтобы их можно было использовать на ПМК «Электроника МК-61».

## 5.5. ДЕНЬ НЕДЕЛИ. БИОРИТМЫ

Определить день недели, если известны число (дата)  $J$ , месяц  $M$ , год  $A$ .

Алгоритм (до 2000 г.)

1. Вычислить двузначные числа  $A_1$  и  $A_2$ , образованные первой и второй парами цифр числа  $A$ .

2. Если  $A_2=0$  и  $A_1$  кратно четырем или  $A_2 \neq 0$  и  $A_2$  кратно четырем, то год високосный. Это можно записать так:

$$b = \begin{cases} 1, & \text{если год високосный,} \\ 0 & \text{в противном случае.} \end{cases}$$

3. Вычислить код месяца  $N$ :

$$N = \begin{cases} 1, & \text{если } M=1 \text{ или } M=2 \text{ при } b=1 \text{ (январь или февраль} \\ & \text{високосного года),} \\ 2, & \text{если } M=1 \text{ или } M=2 \text{ при } b=0, \\ 0 & \text{в остальных случаях.} \end{cases}$$

4. Вычислить код дня  $D = [365,25 \times A_2] + [30,56 \times M] + J + N - 4$ .

5. Вычислить остаток  $R$  от деления  $D$  на 7. Остаток равен номеру искомого дня недели: 0 — воскресенье, 1 — понедельник и т. д.

### Программа ДЕНЬ НЕДЕЛИ

Распределение памяти

|                         |           |                           |
|-------------------------|-----------|---------------------------|
| +P1 — 365,25            | константы | P6 — $b$                  |
| +P2 — 30,56             |           | P7, P8 — рабочие регистры |
| R3 — $N$                |           | PA — $A$ (год)            |
| R4 — $J$ (дата)         |           | PB — $A_1$                |
| P5 — $M$ (номер месяца) |           | PC — $A_2$                |

|    |                  |    |    |        |    |    |       |    |
|----|------------------|----|----|--------|----|----|-------|----|
| 00 | ВП               | 0С | 32 | БП     | 51 | 64 | +     | 10 |
| 01 | 2                | 02 | 33 | 36     | 36 | 65 | X→P8  | 48 |
| 02 | / — /            | 0L | 34 | 0      | 00 | 66 | П→X5  | 65 |
| 03 | X→ПВ             | 4L | 35 | X→П6   | 46 | 67 | П→X2  | 62 |
| 04 | КП→ХВ            | ГL | 36 | П→Х5   | 65 | 68 | ПП    | 53 |
| 05 | F0               | 25 | 37 | 1      | 01 | 69 | =     | 80 |
| 06 | П→ХВ             | 6L | 38 | —      | 11 | 70 | +     | 10 |
| 07 | —                | 11 | 39 | П→Х5   | 65 | 71 | X→P8  | 48 |
| 08 | 2                | 02 | 40 | 2      | 02 | 72 | 7     | 07 |
| 09 | F10 <sup>x</sup> | 15 | 41 | —      | 11 | 73 | :     | 13 |
| 10 | ×                | 12 | 42 | ×      | 12 | 74 | ПП    | 53 |
| 11 | X→ПС             | 4C | 43 | FX = 0 | 5E | 75 | 81    | 81 |
| 12 | F0               | 25 | 44 | 54     | 54 | 76 | 7     | 07 |
| 13 | X→П5             | 45 | 45 | П→Х6   | 66 | 77 | ×     | 12 |
| 14 | F0               | 25 | 46 | FX = 0 | 5E | 78 | —     | 11 |
| 15 | X→П4             | 44 | 47 | 51     | 51 | 79 | C/P   | 50 |
| 16 | П→ХВ             | 6L | 48 | 2      | 02 | 80 | ×     | 12 |
| 17 | ПП               | 53 | 49 | БП     | 51 | 81 | X→П7  | 47 |
| 18 | 86               | 86 | 50 | 55     | 55 | 82 | КП→Х7 | Г7 |
| 19 | П→ХС             | 6C | 51 | 1      | 01 | 83 | П→Х8  | 68 |
| 20 | +                | 10 | 52 | БП     | 51 | 84 | П→Х7  | 67 |
| 21 | FX ≠ 0           | 57 | 53 | 55     | 55 | 85 | B/O   | 52 |
| 22 | 30               | 30 | 54 | 0      | 00 | 86 | 4     | 04 |
| 23 | П→ХС             | 6C | 55 | П→Х4   | 64 | 87 | :     | 13 |
| 24 | FX ≠ 0           | 57 | 56 | +      | 10 | 88 | 1     | 01 |
| 25 | 34               | 34 | 57 | 4      | 04 | 89 | 8     | 08 |
| 26 | ПП               | 53 | 58 | —      | 11 | 90 | 0     | 00 |
| 27 | 86               | 86 | 59 | X→П8   | 48 | 91 | ×     | 12 |
| 28 | FX = 0           | 5E | 60 | П→ХС   | 6C | 92 | F sin | 1C |
| 29 | 34               | 34 | 61 | П→Х1   | 61 | 93 | B/O   | 52 |
| 30 | 1                | 01 | 62 | ПП     | 53 |    |       |    |
| 31 | X→П6             | 46 | 63 | 80     | 80 |    |       |    |

Инструкция. Переключатель углов в положении Г. 365,25 X→П1, 30, 56 X→П2  $J$  В↑  $M$  В↑  $A$  В/O С/П (36 с) «». Индицируется номер дня недели: 0 — воскресенье, 1 — понедельник и т. д. Ввод в Р1 и Р2 только при решении первой задачи.

Контрольные примеры.

1. 3 мая 1985 г. 365,25 X→П1, 30,56 X→П2 3 В↑ 5 В↑ 1985 В/O С/П (36 с) «». Ответ: пятница.

2. 7 января 1990 г. 7 В↑ 1 В↑ 1990 В/O С/П (36 с) «0». Ответ: воскресенье.

## Структура программы

- 00—11 : вычисление и запись  $A_1$  в РВ и  $A_2$  в РС.
- 12—35 : вычисление и запись  $J$  в Р4,  $M$  в Р5,  $b$  в Р6.
- 36—54 : вычисление и запись  $N$  в Р3.
- 55—79 : вычисление  $D$  и  $R$  (запись в РХ), индикация  $R$ .
- 80—85 : подпрограмма вычисления целой части.
- 86—93 : подпрограмма проверки делимости числа на 3.

Адрес алг ДЕНЬ НЕДЕЛИ (цел J, M, A, лит ДЕНЬ, лит, таб ДЕНЬ НЕДЕЛИ  
команды [0:6])  
алг J, M, A, ДЕНЬ НЕДЕЛИ  
рез ДЕНЬ  
нач цел A1, A2, N, b, D, R  
A1 := [A/100]; A2 := A - A1\*100  
если (A2 ≠ 0) ∧ (A2 : 4) ∨ (A2 = 0) ∧ (A1 : 4)  
то b := 1  
иначе b := 0  
все  
выбор  
при (b = 1) ∧ ((M = 1) ∨ (M = 2)): N := 1  
при (b = 0) ∧ ((M = 1) ∨ (M = 2)): N := 2  
иначе N := 0  
все  
выбор  
при (b = 1) ∧ ((M = 1) ∨ (M = 2)): N := 1  
при (b = 0) ∧ ((M = 1) ∨ (M = 2)): N := 2  
иначе N := 0  
все  
55—70  
71—79  
D := [365.25\*A2] + [30.56\*M] + J + N - 4  
R := D - [D/7]\*7  
для i от 1 до 6  
нц  
если i = R  
то ДЕНЬ := ДЕНЬ НЕДЕЛИ [i]  
все  
кц  
кон

Примечание. J, M, A — дата, месяц, год, ДЕНЬ НЕДЕЛИ — воскресенье, понедельник и т. д., ДЕНЬ — один из элементов таблицы ДЕНЬ НЕДЕЛИ, K : T означает: K кратно T, т. е. делится нацело на T.

### ЗАДАНИЯ

1. Количество дней. Рассмотрим два события: первое, как и раньше, характеризуется величинами J, M, A, второе J', M', A'. Тогда разность D' - D есть число дней, прошедших между событиями.

Вычислите, сколько дней вы прожили на свете. Найдите, в какой день за- кончится XX в. и сколько дней останется до этой даты 1 января 1989 г.

Ответы: пятница; 4437 дней.

2. Существует и другой алгоритм для решения задачи, который действи- телен с 1 марта 1900 г. по 28 февраля 2100 г.

$Y = [365.25 G] + [30.6 I] + J - 694.066$ , где Y — число дней, прошедших от 1 марта 1900 г. до дня с заданными J, M, A. Здесь G и I вычисляются из соотношений: если  $M < 3$ , то  $G = A - 1$ ,  $I = M + 13$ ; если  $M \geq 3$ , то  $G = A$ ,  $I = M + 1$ . Остаток от деления Y на 7, как и ранее, образует номер дня недели: 0 — вос- кресенье, 1 — понедельник и т. д. Число дней между двумя событиями опре- деляется тем же способом, как в описании к заданию 1.

Составьте программу на основе этого алгоритма. Решите по ней зада- ние 1.

3. Существует мнение, что в момент рождения у человека заводятся «би- ологические часы», основанные на трех основных его биоритмах: физическом, эмоциональном, интеллектуальном с периодически изменяющейся интенсивно- стью. Периоды составляют: для физического ритма  $T_f = 23$  дня, эмоционального  $T_e = 28$  дней, интеллектуального  $T_i = 33$  дня. В конце каждого полупериода (11,5; 14; 16,5 дней) интенсивность обращается в нуль. «Нулевые» дни назы- вают критическими, в эти дни не рекомендуется заниматься активной дея- тельностью.

Знание каждым человеком «своих» дней весьма полезно. К сожалению, есть все основания сомневаться в наличии столь стабильных ритмов, и все же многие вычисляют свои биоритмы. Для этого есть даже специальные вычисли-

тельные устройства. Можно составить программу для ПМК, вычисляющую «критические дни», согласно следующему алгоритму.

Находится по известным программам количество дней, прожитых челове- ком со дня рождения до данного дня.

Рассмотрим, как определить один из циклов, например физический. Опре- деляется остаток от деления числа прожитых дней на период цикла, в дан- нем случае 23 (для других случаев делителем является 28 или 33). От бли- жайшего полупериода, не меньшего остатка, отнимается остаток. Это и есть количество дней, считая от данного, которые отделяют соответствующий кри- тический день ( $\Phi$ ,  $\Theta$ ,  $\Pi$ ). Достаточно один раз определить критический день, оставшиеся находятся прибавлением к нему соответствующих полупериодов 11,5; 14; 16,5.

З а м е ч а н и е. Составляя программу, удобно найти остаток от деления числа прожитых дней на полупериод и вычесть его из полупериода.

Пример. Человек родился 12 февраля 1930 г. Расчет выполнен 13 сентября 1984 г. Число дней  $Y$  от начала отсчета до дня рождения (см. алгоритм в задании 2) 10 972, от начала отсчета до сего дня 30 909. Прожито 19 937 дней.

Остаток от деления этого числа на 23 равен 19, расстояние до конца бли- жайшего полупериода  $23 - 19 = 4$ . Обычно в качестве ответа указывают ин- тервал в 1—2 дня. Следовательно, в нашем случае ближайший критический день  $\Phi$  наступит через 3—5 дней после данного.

Остаток от деления числа прожитых дней на 28 равен 1. Конец ближай- шего полупериода 14. Расстояние до критического дня  $\Theta : 14 - 1 = 13$ , т. е. он наступит через 12—14 дней.

Для  $\Pi$  разность равна 11,5, критический день надо ожидать через 11—12 дней.

Теперь вы сможете рассчитать свои критические дни, а также дни своих родственников, знакомых, товарищей.

### Программа БИОЛОГИЧЕСКИЕ РИТМЫ \*

|    |       |    |    |       |    |    |       |    |
|----|-------|----|----|-------|----|----|-------|----|
| 00 | 3     | 03 | 21 | K[X]  | 34 | 42 | 49    | 49 |
| 01 | X→P0  | 40 | 22 | P→XС  | 6С | 43 | KП→X3 | Г3 |
| 02 | F0    | 25 | 23 | Х     | 12 | 44 | P→X5  | 65 |
| 03 | 6     | 06 | 24 | П→ХД  | 6Г | 45 | 1     | 01 |
| 04 | X→P6  | 46 | 25 | —     | 11 | 46 | 2     | 02 |
| 05 | F0    | 25 | 26 | C/P   | 50 | 47 | +     | 10 |
| 06 | PП    | 53 | 27 | FL0   | 5Г | 48 | X→P5  | 45 |
| 07 | 31    | 31 | 28 | 15    | 15 | 49 | P→X1  | 61 |
| 08 | X→PС  | 4C | 29 | Fπ    | 20 | 50 | P→X3  | 63 |
| 09 | C/P   | 50 | 30 | C/P   | 50 | 51 | ×     | 12 |
| 10 | PП    | 53 | 31 | X→P3  | 43 | 52 | K[X]  | 34 |
| 11 | 31    | 31 | 32 | F0    | 25 | 53 | P→X2  | 62 |
| 12 | P→XС  | 6C | 33 | X→P5  | 45 | 54 | P→X5  | 65 |
| 13 | —     | 11 | 34 | F0    | 25 | 55 | ×     | 12 |
| 14 | X→PД  | 4Г | 35 | X→P4  | 44 | 56 | K[X]  | 34 |
| 15 | P→ХД  | 6Г | 36 | P→X5  | 65 | 57 | +     | 10 |
| 16 | KП→X6 | 6Г | 37 | KП→X5 | 6Г | 58 | P→X4  | 64 |
| 17 | X→PС  | 4C | 38 | F0    | 25 | 59 | +     | 10 |
| 18 | :     | 13 | 39 | 3     | 03 | 60 | P→XA  | 6  |
| 19 | 1     | 01 | 40 | —     | 11 | 61 | —     | 11 |
| 20 | +     | 10 | 41 | FX<0  | 5C | 62 | B/O   | 52 |

Инструкция. Введите 6 констант: 365.25 X→P1, 30.6 X→P2, 694066 X→P4, 11.5 X→P7, 14 X→P8, 16.5 X→P9. Если вычисления производятся многократно, то константы вводятся только первый раз. Введите значения

*J, M, A* — дата, месяц и год рождения:  $J \uparrow M \uparrow A$   $B \uparrow B/O$  С/П. На индикаторе — число дней, прошедших от начала отсчета (1 марта 1900 г.) до дня рождения. После останова введите  $J'$ ,  $M'$ ,  $A'$  — дата, месяц, год момента вычисления:  $J' \uparrow M' \uparrow A' \uparrow C/P$  индицируется искомое значение для ритма Ф. В качестве ответа выбираем интервал в 1—2 дня в окрестности этой величины. Далее С/П «значение для ритма Э», С/П «значение для И» С/П — задача решена.

**Контрольный пример.** Человек родился 13 мая 1970 г.; его критические дни вычисляются 24 октября 1987 г.

365.25 X→P1, 30.6 X→P2, 694066 X→PA, 11.5 X→P7, 14 X→P8, 16.5 X→P9 13 B↑ 5 B↑ 1970 B↑ B/O С/П (12 с) «25672», 24 B↑ 10 B↑ 1987 С/П (12 с) «9,5», С/П (4 с) «11», С/П (4 с) «12,5», С/П (2 с) «л» — задача решена.

Ответы:  $\Phi \leftarrow 9-10$ ,  $\mathcal{E} \leftarrow 10-12$ ,  $I \leftarrow 12-13$ .

Проверьте по программе решение примера из данного задания, который решали по алгоритму.

### ЗАДАНИЕ

**Математический фокус.** Предложите товарищу сыграть с вами в игру. Умножь дату своего рождения на 100, к произведению прибавь 900 и номер месяца рождения. Результат умножь на 100, прибавь 725 и число, образованное последними двумя цифрами года рождения. Назови результат, и я угадаю твой день, месяц и год рождения.

**Алгоритм.** Из введенного числа вычтите 90 725. Разбейте разность справа налево на пары цифр. Первая, вторая и третья пары дают соответственно день, месяц и последние две цифры года рождения.

**Пример.** Вы родились 3 января 1970 г.  $(3 \cdot 100 + 900 + 1) \cdot 100 + 725 + 70 = 120\,895$ ;  $120\,895 - 90\,725 = 30:170$ .

Будем использовать привычные обозначения  $J$ ,  $M$ ,  $A$  — день, месяц, год рождения;  $A_2$  — число, образованное последними двумя цифрами  $A$ ;  $J$ ,  $M$ ,  $A_2$  — двухзначные числа.  $(100J + 900 + M) \cdot 100 + 725 + A_2 = J \cdot 10^4 + M \cdot 10^2 + A_2 + 9\,000\,000 = JMA_2 + 9\,000\,000$ . Черта означает, что выражение является числом, а не произведением.

Составьте программу для решения этой задачи. Найдите по программе и проверьте непосредственным вычислением по алгоритму день, месяц, год рождения по следующим данным: 311195 (XIX в); 300849 (XX в). Определите время события по числу 161841 (XX в).

Ответы: 22 апреля 1870 г.; 21 января 1924 г.; 7 ноября 1917 г.

Можно придумать немало аналогичных задач. Убедитесь в справедливости тождества

$$(100J + 900 + M) \cdot 100 + A_2 = J \cdot 10^4 + M \cdot 10^2 + A_2 + 9\,000\,000 = JMA_2 + 9\,000\,000.$$

На этот раз, чтобы получить ответ, достаточно из результата вычесть  $9 \cdot 10^6$  ( $A_2$  — число, образованное первыми двумя цифрами  $A$ ). Составьте правила игры и соответствующую программу. Она должна быть проще: теперь не требуется указывать век.

### 5.6. МИССИОНЕРЫ И ЛЮДОЕДЫ

На левом берегу реки встретились  $n$  людоедов и  $n$  миссионеров. Им надо переправиться на правый берег, но в лодку могут поместиться только двое, зато каждый умеет грести. Переяправу

надо организовать так, чтобы ни на одном берегу не оказалось людоедов больше, чем миссионеров \*.

Будем решать задачу в диалоге с ПМК. Мы отвечаем за левый берег, ПМК — за правый. Переправа лодки на противоположный берег называется ходом. Число людоедов в лодке обозначим  $l$ , число миссионеров —  $m$  ( $1 \leq l+m \leq 2$ ). Ход закодируем двузначным числом  $l \cdot 10 + m$ . Задачу будем решать для  $n=3$ .

Нам не потребуется вести вычисления. Нетрудно сообразить, кого можно посадить в лодку, чтобы не нарушить равновесия на обоих берегах. ПМК для ответных ходов нужна специальная вычислительная процедура.

В распоряжении ПМК имеются три варианта организации переправы: в лодку садится один людоед; в лодке один миссионер; в лодке один людоед и один миссионер. Остальные варианты (два людоеда или два миссионера) приведут к нарушению условия или к повторению предыдущей ситуации.

#### Алгоритм.

Если хотя бы на одном берегу нет миссионеров, то в лодку садится один людоед: если нет миссионеров на правом берегу, то все они на другом, отправление туда людоеда не нарушит равновесия; при отсутствии миссионеров и на левом берегу этот ход также безопасен.

Если на правом берегу только один миссионер, он садится в лодку: допустим на каждом берегу равновесие сил людоедов и миссионеров, тогда переправа налево людоеда нарушит равновесие; напротив, переправа миссионера безопасна для обоих берегов.

Во всех случаях, кроме двух перечисленных, в лодку садятся двое — людоед и миссионер, таким образом, условие задачи не нарушено.

Поясним алгоритм схемой на рис. 19.

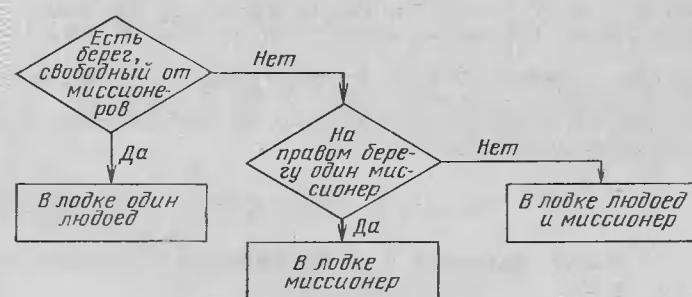


Рис. 19

\* Кибернетики (Ньюэлл, Шоу, Саймон и др.) использовали эту задачу для уточнения логических возможностей компьютера в работах по искусственному интеллекту. Задача решается методом динамического программирования.

### Распределение памяти ПМК

**P0** — число людоедов в лодке при очередном ходе человека (первая цифра индицируемого числа);  
**P1** — число миссионеров в лодке при очередном ходе человека (вторая цифра индицируемого числа);  
**P2, P3** — число людоедов и миссионеров на правом берегу. Исходное значение — 0, 0;  
**P4, P5** — число людоедов и миссионеров на левом берегу. Исходное значение —  $n$ ,  $n$  (в нашей задаче 3,3).

|    |      |    |    |       |    |    |       |       |
|----|------|----|----|-------|----|----|-------|-------|
| 00 | X→Π1 | 41 | 19 | Π→X0  | 60 | 37 | 43    | 43    |
| 01 | ↔    | 14 | 20 |       | 1  | 01 | 38    | KΠ→X2 |
| 02 | X→Π0 | 40 | 21 |       | 0  | 00 | 39    | KΠ→X4 |
| 03 | Π→X4 | 64 | 22 | ×     | 12 | 40 | 1     | G2    |
| 04 | Π→X0 | 60 | 23 | [Π→X1 | 61 | 41 | 0     | G4    |
| 05 | —    | 11 | 24 | +)    | 10 | 42 | C/P   | 00    |
| 06 | X→Π4 | 44 | 25 | C/Π   | 50 | 43 | KΠ→X3 | 50    |
| 07 | Π→X5 | 65 | 26 | Π→X4  | 64 | 44 | KΠ→X5 | G3    |
| 08 | Π→X1 | 61 | 27 | Π→X5  | 65 | 45 | Π→X3  | 63    |
| 09 | —    | 11 | 28 | +)    | 10 | 46 | FX=0  | 5E    |
| 10 | X→Π5 | 45 | 29 | FX=0  | 5E | 47 | 50    | 50    |
| 11 | Π→X2 | 62 | 30 |       | 33 | 48 | 1     | 01    |
| 12 | Π→X0 | 60 | 31 | F π   | 20 | 49 | C/P   | 50    |
| 13 | +    | 10 | 32 | C/Π   | 50 | 50 | KΠ→X2 | G2    |
| 14 | X→Π2 | 42 | 33 | Π→X3  | 63 | 51 | KΠ→X4 | G4    |
| 15 | Π→X3 | 63 | 34 | Π→X5  | 65 | 52 | 1     | 01    |
| 16 | Π→X1 | 61 | 35 | ×     | 12 | 53 | 1     | 01    |
| 17 | +    | 10 | 36 | FX=0  | 5E | 54 | C/P   | 50    |
| 18 | X→Π3 | 43 |    |       |    |    |       |       |

**Инструкция.** *n* X→P4 X→P5 0 X→P2 X→P3. Ввод требуется только один раз — в начале решения. Игрок, управляющий посадкой в лодку на левом берегу, вводит свои ходы нажатиями *l* B↑ *m* B/O С/П. Ходы индицируются двузначными числами: первая цифра *l*, вторая *m*. После индикации очередного хода нажатием клавиши С/П ПМК запускается на вычисление своего ответного хода — состава людей в лодке при переправе справа налево. Ход ПМК закодирован соответствующим двузначным числом. Затем свой ход вводит человек и т. д., пока на индикаторе не появится *п*, что означает окончание решения. Бывает, что человек еще до этого приходит к выводу, что решения нет.

*Контрольный пример.*  $n=3$ ; 3 X $\rightarrow$ П4 X $\rightarrow$ П5, 0 X $\rightarrow$ П2 X $\rightarrow$ П3; 1 B $\uparrow$  B/O C/P (8 с) «11», C/P (5 с) «1», 2 B $\uparrow$  0 B/O C/P (8 с) «20», C/P (5 с) «10» и т. д. Индикация π означает, что задача решена. Первый ход а человеком, далее ходы чередуются.

## Структура программы

- 00—02 : запись значений  $l$  и  $m$ , введенных человеком, в Р0 и Р1.  
03—18 : преобразование содержимого Р2—Р5 после очередного хода человека.  
19—25 : формирование двузначного числа, описывающего ход человека, и индикация этого числа. Нажатием клавиши С/П запускается продолжение вычислений.

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26—54                 | : формирование и индикация хода ПМК, если задача еще не решена.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 26—36                 | : проверка окончания решения. Если на левом берегу людей нет, т. е. в Р4 и Р5 нули, то на индикаторе π — задача решена. В противном случае (команда 30) проверяется, есть ли берег, где миссионеров нет.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 37                    | : на каждом берегу есть миссионеры (переход к команде 43).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 38—42                 | : производятся преобразования в Р2 и Р4, когда хотя бы на одном из берегов нет миссионеров. Индикация числа 10 означает, что в лодку садится один людоед.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 43, 44                | : в лодке будет находиться миссионер, т. е. содержимое Р3 уменьшается на 1, Р5 увеличивается на 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 45, 46                | : проверка варианта, когда на правом берегу только один миссионер.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 48, 49                | : если да, то индицируется 01 — в лодку садится миссионер.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 47, 50—54             | : на правом берегу более одного миссионера. В лодку садятся людоед и миссионер. Проводятся необходимые преобразования в Р2, Р4, индицируется 11 — ход ПМК.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Адреса команд         | <pre> алг МИССИОНЕРЫ И ЛЮДОЕДЫ (нат n, R2, R3, R4, R5, лит А) арг n, R2, R3, R4, R5 рез А нач nat x, цел l, m, B     A := «задача имеет решение»; R2 := 0; R3 := 0; R4 := n; R5 := n     пока A = «задача имеет решение»         нц             ввод x             l := [x/10]; m := x - l*10             R2 := R2 + l; R3 := R3 + m; R4 := R4 - l; R5 := R5 - m             выв x             если R4 + R5 = 0                 то A := «задача решена»             иначе выбор                 при R3*R5=0 : x := 10; R2 := R2 - 1; R4 := R4 + 1                 при R3=1 : x := 1; R3 := R3 - 1; R5 := R5 + 1                 иначе x := 11; R3 := R3 - 1; R5 := R5 + 1;                     R2 := R2 - 1; R4 := R4 + 1             все             выв x             если B = 0                 то A := «нет решения»             все         кон     кон </pre> |
| 00—25 {               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 26—32 {               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33—41                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 43—48                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 43, 44,               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 50—53                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 42, 49, 54            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Выполняется человеком |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

Здесь  $x=10*l+m$ ;  $B=1$ , если существует продолжение решения задачи, и  $B=0$  в противном случае.

## ЗАДАНИЯ

1. За исключением первых ходов человека и ПМК игра протекает единственным возможным способом. Определите все возможные варианты первого хода.

2. Решите, используя программу, задачу для  $n=2$ .
3. Докажите, что при  $n \geq 4$  решения нет.

4. Условие задачи прежнее, только теперь в лодку могут сесть трое. Одно место в ней постоянно занимает Геракл\*, который во время переездов не выходит из лодки, хотя сам и не гребет. В его присутствии люди на берегу, к которому причалила лодка, ведут себя спокойно.

Сформулируйте алгоритм, составьте на его основе диалоговую программу для любого  $n$  и запишите алгоритм на УАЯ.

## ВМЕСТО ЗАКЛЮЧЕНИЯ. К ЧИТАТЕЛЬЮ

На этом мы завершаем вторую часть, а с нею и всю книгу. Пришла пора расстаться, наш терпеливый читатель.

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

У программируемого микрокалькулятора есть своя экологическая ниша, и он еще долго послужит человеку. И все же завтра в школу, а затем и к нам домой придет персональный компьютер. Смело садитесь за него, вам освоить его будет легче, чем тем, кто впервые столкнется с вычислительной техникой.

## ПРИЛОЖЕНИЕ \*\*

1. Составьте программу решения линейного уравнения  $ax=b$ , где  $a, b$  — вещественные числа (см. [12, ч. I, А3, с. 91]).

Уравнение имеет единственный корень, если  $a \neq 0$ ; не имеет корней, если  $a=0, b \neq 0$ . Если же  $a=0, b=0$ , то в качестве корня может быть любое число.

Кодирование: выводится значение корня, если он единственный;  $\pi$ , если любое число является корнем; ЕГГОГ, если уравнение не имеет корней.

2. Составьте программу, подсчитывающую число корней квадратного уравнения.

Квадратное уравнение  $ax^2+bx+c=0$  имеет два различных корня, если дискриминант  $D > 0$ , один корень (второй кратности), если  $D=0$ , не имеет корней, если  $D < 0$ . ПМК сообщает: «0» при отсутствии корней, «1» при наличии кратного корня и при наличии двух корней «2».

3. Составьте программу вычисления наибольшего общего делителя (НОД) двух натуральных чисел методом последовательного вычитания (см. [12, ч. I, А6, с. 92]) по следующему алгоритму.

\* Герой древнегреческой мифологии, отличавшийся необыкновенной физической силой.

\*\* Большинство задач взяты из [12]. Программирование решений на ПМК может пригодиться учителям и учащимся.

1. Если числа равны, то каждое из них есть НОД.

2. Из большего числа вычитается меньшее, большее число заменяется полученной разностью. Возвратиться к указанию 3.1.

Распределение памяти

$PA$  — большее число  $a$ ,  $PB$  — меньшее  $b$ .

4. Составьте программу, отыскивающую в линейном массиве (линейной таблице) наименьший элемент (см. [12, ч. I, А7]).

5. Составьте программу, определяющую число ненулевых элементов в таблице из  $n$  элементов  $a_1, a_2, a_3, \dots, a_n$ ;  $n < 9$  (см. [12, ч. II, задача 1, с. 100]).

Элементы условимся хранить в регистрах  $D, C, B, A, \dots$

6. Составьте программу, определяющую в таблице число элементов, абсолютная величина которых больше семи (см. [12, ч. II, задача 2, с. 100]).

7. Составьте программу, дающую ответ «да» (1) или «нет» (0) в зависимости от наличия или отсутствия числа 7 в таблице (см. [12, ч. II, задача 3, с. 100]).

8. В таблице на  $n$  элементов найдите разность наибольшего и наименьшего чисел. Составьте программу (см. [12, ч. II, задача 4, с. 100]),  $n < 10$ .

9. Составьте программу вычисления количества элементов таблицы, больших среднего арифметического всех ее элементов,  $n < 9$  (см. [12, ч. II, задача 7, с. 100]).

10. Составьте программу вычисления числа повторений в таблице из  $n$  элементов максимального числа,  $n < 9$  (см. [12, ч. II, задача 9, с. 100]).

11. Составьте программу поиска в таблице элементов, равных нулю. Если они есть, ПМК должен сообщить номер первого из них, в противном случае выводится 0,  $n < 10$  (см. [12, ч. II, задача 12, с. 101]).

12. Проверьте, есть ли в таблице отрицательные элементы. Если есть, найдите номер последнего из них, в противном случае на индикатор должен быть выведен нуль. Составьте программу,  $n < 10$  (см. [12, ч. II, задача 13, с. 101]).

13. Натуральное число называют совершенным, если оно равно сумме всех своих делителей, не считая его самого. Например,  $6=1+2+3$ , число 6 совершенное. Составьте программу, проверяющую, является ли данное число совершенным (см. [12, ч. II, задача 23, с. 101]).

14. Составьте программу, которая упорядочивала бы элементы линейной таблицы чисел по возрастанию (см. [12, ч. I, А8, с. 93]).

Задача решается с помощью двух циклов перебора внешнего и внутреннего, включенного в его состав. Рассматривается  $a_n$  ( $n < 7$ ). Для каждого  $i < n$  во внутреннем цикле сравниваются  $a_i$  и  $a_n$ . Если  $a_i > a_n$ , то эти два элемента меняются местами. После полного перебора по  $i$  на последнем месте находится наибольший элемент, который из дальнейшего рассмотрения исключается. Затем процедура, но уже во внешнем цикле, повторяется для второго с конца элемента  $a_{n-1}$  и т. д. Распределение памяти может быть следующее:  $PO$  — счетчик внешних циклов на  $n-1$  повторение, каждый элемент, начиная с  $a_n$ , сравнивается со всеми предшествующими.

$P1$  — переменная для косвенного обращения к элементам таблицы во внешнем цикле, начальное значение 11.

$P2$  — счетчик внутренних циклов, начальное значение — содержимое  $PO$ .

$P3$  — переменная для косвенного обращения к элементам таблицы во внутреннем цикле, начальное значение на 1 меньше содержимого  $P1$ .

$P4$  —  $PA$  — таблица на семь элементов или меньше, которые вводятся с клавиатуры в регистры в последовательности:  $A, 9, 8, \dots$ .

15. Данна целочисленная таблица  $A[1 : n]$ . Найдите наименьшее число элементов  $k$ , которые нужно выкинуть из последовательности  $A[1], A[2], \dots, A[n]$ , чтобы осталась возрастающая последовательность (см. [12, ч. II, задача 29, с. 102]). Алгоритм решения задачи следующий.

Строится вспомогательная таблица  $K$  длины  $n$ . Начальное значение всех ее элементов — единица. Удобно использовать циклы с параметрами  $i$  и  $j$ . Параметр  $i$  изменяет значение от  $n$  до 2 с шагом, равным  $-1$ , для каждого значения  $i$ , параметр  $j$  изменяется от  $i-1$  до 1. При последовательном сравнении

$a_i > a_j$  и  $k_i > k_j$ . Если они выполнены, то  $k_j$  увеличивается на единицу, в противном случае  $k_j$  не изменяется.

После окончания обоих циклов для каждого элемента таблицы  $A$  в таблице  $K$  записано число элементов, в составе которых данный элемент может входить в возрастающую последовательность. Наибольшее число из  $K$  соответствует количеству элементов, которые следует сохранить. Если вычесть его из  $n$ , то получим наименьшее число элементов, которые нужно исключить из  $A$ . Перебор элементов таблицы с отрицательным шагом (с конца) вызван необходимостью привести алгоритм в соответствие с программой для ПМК.

Поясним алгоритм примером. Таблица  $A$ : 3, 1, 2, 5, 1;  $n=5$ . Начальный вид  $K$ : 1, 1, 1, 1, 1. Сравниваем последний член  $A$  с каждым предшествующим. Условия  $a_i > a_j$  и одновременно  $k_i > k_j$  выполнены только для второго члена —  $K$ : 1, 2, 1, 1, 1. Далее предпоследний член — 5 сравнивается с предшествующими. Он больше любого из них, но для второго члена не выполнено  $k_i > k_j$  —  $K$ : 2, 2, 1, 1, 1, для  $a_i=2$  получаем  $K$ : 2, 3, 2, 1, 1. После сопоставления членов 1 и 3 таблица  $K$  не изменится. Максимальное число членов, которые можно сохранить, три. Среди них обязателен второй член 1 со значением  $k=3$ . Сохраняется последовательность 1, 2, 5.

Обоснуйте алгоритм. К элементу  $k_j$  прибавляется 1 от каждого  $k_i$ , удовлетворяющего условиям  $a_j < a_i$ ,  $k_j < k_i$ . Так осуществляется перебор справа налево. Соотношение  $a_j < a_i$  нужно выдерживать потому, что в противном случае придется исключить хотя бы один элемент  $a_i$  или  $a_j$ . Условие  $k_j < k_i$  объясняется следующими соображениями. При  $k_j > k_i$  существовало бы такое  $a_{i+m}$ , что  $a_j < a_{i+m}$  и  $a_j > a_{i+m}$ . Поскольку  $a_j$  входит в возрастающую последовательность вместе с  $a_{i+m}$ , следовательно, оно не может войти в последовательность одновременно с  $a_i$ . Поясним это примером.

Дана таблица 1, 3, 2.  $K$ : 1, 1, 1; следующие значения  $K$ : 2, 1, 1, так как  $a_2 > a_3$  и  $a_1 < a_3$  и  $k_1=2$  означает, что  $a_1$  вместе с  $a_3$  могли бы войти в возрастающую последовательность. Далее,  $a_1 < a_2=3$ , тем не менее  $a_2$  придется исключить, так как  $a_2$  не может войти в группу сохранившихся элементов вместе с  $a_3$ , а  $a_1$  и  $a_3$  уже включены.

Для реализации алгоритма на ПМК приходится учитывать некоторые его особенности. Это прежде всего невозможность хранения в памяти больших таблиц. Будем рассматривать таблицы не более чем на  $n=6$  однозначных чисел, причем, чтобы еще более упростить программу, чисел неотрицательных. Расположим их в регистрах 9, 8, 7, ..., 4. Данные будем вводить двузначными числами, где цифры отделены друг от друга запятой. Вторая цифра — элемент таблицы, первая — соответствующее значение  $k$  (в начале 1). Например, 1.4 X→P7 означает ввод в P7 элемента таблицы (4) при  $k=1$ . В ходе решения содержимое регистров 4—9 разбивается на две цифры. Сравниваться будут вторые цифры  $a_i$ ,  $a_j$  и первые  $k_i$ ,  $k_j$ .

Если для данных  $i$  и  $j$  окажется  $a_i > a_j$ ,  $k_i > k_j$ , то первая цифра в регистре, соответствующем  $j$ , увеличивается на единицу, в противном случае не изменяется.

По выходе из циклов остается найти в таблице  $K$  максимальную первую цифру и вычесть ее из  $n$ .

Прежде чем составить программу, распределим память ПМК. В Р0 и Р1 будем хранить значения параметров для косвенного обращения к регистрам 9, 8 и т. д., т. е. для организации циклов по  $i$  и  $j$ , начальное значение содер-жимого Р1 — 9, Р0 — 8, Р2, Р3 — счетчики циклов по  $i$  и  $j$ , начальное значение  $n$ . Р4 — Р9 — для хранения введенных с пульта введенных с пульта двузначных чисел, разделенных запятыми,  $n \leq 6$ . РА, РВ и РС, РД — для хранения первых и вторых цифр рассматриваемой пары чисел.

16. Составьте программу, отыскивающую все возможные расстановки ферзей на досках  $5 \times 5$  и  $7 \times 7$  так, чтобы ферзи не были друг друга. Для шахматной доски  $8 \times 8$  построите программу, которая отыскивает один из таких способов расстановки (см. [12, ч. II, задача 38, с. 133]).

Будем описывать положение ферзя двузначным числом: первая цифра — абсцисса клетки, вторая — ордината. Это соответственно — номера столбца и строки, на пересечении которых находится клетка. Например, число 23 означает Ф(2, 3) — ферзь находится на пересечении 2-го столбца и 3-й строки.

Начнем с решения задачи на досках  $5 \times 5$  и  $7 \times 7$ . Длина шага  $h$  для абсциссы  $h_x$  (ординаты  $h_y$ ) равна разности между абсциссами (ординатами) данной и предшествующей клеток, занятых ферзями в рассматриваемой расстановке. Алгоритм будет следующий.

1. Задается произвольное число  $m=0, 1, \dots, n-1$ ;  $x=m+1$ ,  $y=1$ .
2.  $h_y=1$ . Если  $m < n-2$ , то  $h_x=2$ , иначе  $h_x=3$ .
3. Определяется позиция  $\Phi(x, y)$ .
4. Вычисляется координата очередной клетки — к соответствующей предыдущей координате прибавляется длина шага.
5. Если вновь вычисленная абсцисса больше  $n$ , то из нее вычитается  $n$ .
6. Возврат к указанию 3.

Процедура повторяется  $n$  раз, что соответствует числу возможных расстановок ферзей, удовлетворяющих условию задачи.

Пример.  $n=5$ ,  $m=3$  — начальная позиция ферзя  $\Phi(4, 1)$ .  $m=n-2$ , следовательно,  $h_x=3$ ,  $h_y=1$ .

$\Phi(4; 1)$ ;  $\Phi(4+3; 2)=\Phi(7; 2)=\Phi(7-5; 2)=\Phi(2; 2)$ ;  $\Phi(2+3; 3)=\Phi(5; 3)$ ;  $\Phi(5+3; 4)=\Phi(8; 4)=\Phi(3; 4)$ ;  $\Phi(3+3; 5)=\Phi(6; 5)=\Phi(1; 5)$  (рис. 20).

Алгоритм решения задачи для шахматной доски  $8 \times 8$ .

1. Первая позиция ферзя описывается числом 42 (4 — абсцисса, 2 — ордината), вторая — числом на 19 больше, каждая из двух последующих — числом на 12 больше предыдущего.

2. Для каждой позиции существует другая позиция, которая получается вычитанием первой из 99.

17. Рекурсивная последовательность. Последовательность  $a_1, a_2, a_3, \dots$  определяется так:

$$a_1=1, \quad (1) \quad a_{2n}=a_n, \quad (2) \quad a_{2n-1}=a_n+a_{n+1}. \quad (3)$$

Постройте программу для ПМК, вычисляющую  $a_n$  по номеру  $n$ . Составьте алгоритм решения задачи на УАЯ (см. [12, ч. II, задача 28, с. 102]). Алгоритм решения задачи следующий.

Если  $n$  четно, то согласно формуле (2) оно делится пополам до тех пор, пока частное не окажется нечетным. Рассмотрим частное; оно равно единице,

|   |   |   |   |   |   |
|---|---|---|---|---|---|
|   |   |   |   |   |   |
| 5 | * |   |   |   |   |
| 4 |   | * |   |   |   |
| 3 |   |   |   | * |   |
| 2 | * |   |   |   |   |
| 1 |   |   |   |   |   |
|   | 1 | 2 | 3 | 4 | 5 |

Рис. 20

если  $a_n$  — целая положительная степень двойки. В этом случае задача решена.

Если  $n \neq 1$  нечетно, то  $a_n$  преобразуется по формуле (3), одно слагаемое с четным номером, другое — с нечетным.

Чтобы сократить число слагаемых, обратим внимание на их коэффициенты и будем приводить подобные члены. Вначале оба коэффициента равны единице. Слагаемое с нечетным номером преобразуем по (3), с четным — по (2). В сумме из трех слагаемых два обязательно подобны, что видно из сопоставления формул (2) и (3). После приведения подобных  $a_n$ , как и ранее, представлено суммой двух членов с соответствующими коэффициентами.

Описанные преобразования по формулам (2) и (3) повторяются до тех пор, пока член с нечетным номером не окажется равным  $a_1$ , в этом случае другой член есть  $a_2$ , тогда  $a_n$  равен сумме этих членов.

Иллюстрируем алгоритм конкретным примером:

$$\begin{aligned} a_{84} = a_{42} = a_{21} = a_{11} + a_{10} = a_5 + a_6 + a_5 = 2a_5 + a_6 = 2(a_3 + a_2) + a_3 = \\ = 3a_3 + 2a_2 = 3(a_1 + a_2) + 2a_1 = 5a_1 + 3a_2 = 8. \end{aligned}$$

Распределим память ПМК: члены с нечетными и четными номерами будем хранить соответственно в РА и РВ, их коэффициенты в Р4 и Р5. В качестве оперативных используем регистры 2 и 3 — в Р2 будем хранить нечетные номера, когда  $a_{2n+1}$  будет представлено по (3); в Р3 — четные номера.

18. Составьте алгоритм и программу, которая поможет найти все двузначные и трехзначные числа, у которых при умножении на 2, 3, 4, ..., 9 сумма цифр не изменится. Запишите алгоритм на УАЯ.

Решение. Полный перебор натуральных чисел от 10 до 999 на ПМК неэффективен, поэтому перебор нужно ограничить. Искомые числа  $n$  кратны 9. Действительно,  $9n$  делится на 9, его сумма цифр, следовательно, кратна 9. Но по условию у числа  $n$  такая же сумма цифр, как у  $9n$ ;  $n$ , следовательно, делится на 9, т. е. принадлежит множеству 9, 18, ..., 189 и т. д. На основе этих рассуждений можно составить алгоритм сокращенного перебора, а также соответствующую программу.

Требуемые числа  $n$  выбираются среди чисел, кратных 9. Каждый раз проверяется совпадение сумм цифр для чисел  $n$  и  $kn$  ( $k=2, 3, \dots, 8$ ). Наименьшее двузначное число, кратное 9, это 18. С него и начинаем испытание. Следующие числа будем формировать на основе предыдущих с шагом 9.

19. Найдите все четырехзначные числа, являющиеся точным квадратом, если первые, а также последние две его цифры одинаковы.

Известно, что квадрат имеет вид  $10^8x + 10^2x + 10y + y = 11(100x + y)$ . Квадрат, как и те двузначные числа, что образованы первой и второй парами его цифр, кратны 11, тогда и основание кратно 11. Следовательно, его надо искать среди чисел интервала от 33 до 99 с шагом 11.

Ответ: 7744 = 88<sup>2</sup>.

20. Найдите знак числа  $9\pi^4 - 240\pi^2 + 1492$ .

21. Вычислите с помощью ПМК  $3\sqrt[3]{\sqrt[3]{5} - \sqrt[3]{4}} - \sqrt[3]{2} - \sqrt[3]{20} + \sqrt[3]{25}$ . Имейте в виду, что восьмой знак может оказаться неверным из-за погрешности вычислений. Получите результат также алгебраическим преобразованием, не используя ПМК.

22. Вычислите точное значение  $(649^2 + 13 \cdot 180^2)^2 - 13(2 \cdot 649 \cdot 180)^2$ .

23. Числа  $a$  и  $b$  — натуральные; составьте программу вычисления количества цифр  $n$  в числе  $a^b$ . Найдите по программе количество цифр в числах:  $5^4, 25^{36}, 1987^{1987}$ .

24. Составьте программу вычисления суммы  $1/1 + 1/(1 \cdot 2) + 1/(1 \cdot 2 \cdot 3) + \dots + 1/(1 \cdot 2 \cdot 3 \cdots n)$ . Чтобы упростить решение, удобно преобразовать выражение к виду  $\frac{1}{n!}((1/n+1)/(n-1)+1)/(n-2)+\dots+1)/1$  — число делений  $n$ . Нетрудно убедиться, что последовательность рассматриваемых сумм сходится к числу  $e-1 \approx 1,7182818$ .

25. Последовательность  $\{x_n\}$  задана рекуррентным способом — каждый следующий член выражается через предыдущий:  $x_1=1, x_{n+1}=\sqrt[3]{3+2x_n}$ . Составьте программу для вычисления членов последовательности. Найдите с ее помощью первые 10 членов и убедитесь, что последовательность сходится к числу 3.

26. Докажите, что следующие последовательности сходятся к  $\sqrt{2}$ :

а)  $x_1=1, x_{n+1}=1+1/(1+x_n)$  при  $n \geq 1$ ;

б)  $x_1=1, x_{n+1}=x_n/2+1/x_n$  при  $n \geq 1$ .

27. Последовательность  $\pi^4/(1+1/2^4+1/3^4+1/4^4+\dots+1/n^4)$  сходится к числу 90. Проверьте это вычислением по программе.

## ОТВЕТЫ

### 1. Программа ЛИНЕЙНОЕ УРАВНЕНИЕ, $b$ В↑ $a$ В/О С/П.

|    |             |    |    |      |    |    |     |    |
|----|-------------|----|----|------|----|----|-----|----|
| 00 | FX $\neq$ 0 | 57 | 04 | ↔    | 14 | 07 | F π | 20 |
| 01 | 04          | 04 | 05 | FX=0 | 5E | 08 | C/P | 50 |
| 02 | :           | 13 | 06 | 09   | 09 | 09 | K—  | 27 |
| 03 | C/P         | 50 |    |      |    |    |     |    |

Контрольные примеры. 3 В↑ 2 В/О С/П «1,5», 5 В↑ 0 В/О С/П «ЕГГОГ», 0 В↑ В/О С/П «Л».

### 2. Программа ЧИСЛО КОРНЕЙ, $b$ В↑ $a$ В↑ $c$ В/О С/П

|    |                 |    |    |      |    |    |     |    |
|----|-----------------|----|----|------|----|----|-----|----|
| 00 | ×               | 12 | 06 | FX=0 | 5E | 11 | 14  | 14 |
| 01 | 4               | 04 | 07 | 10   | 10 | 12 | 2   | 02 |
| 02 | ×               | 12 | 08 | 1    | 01 | 13 | C/P | 50 |
| 03 | ↔               | 14 | 09 | C/P  | 50 | 14 | 0   | 00 |
| 04 | FX <sup>2</sup> | 22 | 10 | FX<0 | 5C | 15 | C/P | 50 |
| 05 | —               | 11 |    |      |    |    |     |    |

Контрольные примеры.  $x^2 - 2x + 1 = 0: 2 / - /$  В↑ 1 В↑ В/О С/П «1»,  $x^2 - 3x + 2 = 0: 3 / - /$  В↑ 1 В↑ 2 В/О С/П «2»,  $x^2 - 3x + 5 = 0: 3 / - /$  В↑ 1 В↑ 5 В/О С/П «0».

### 3. Программа НАИБОЛЬШИЙ ОБЩИЙ ДЕЛИТЕЛЬ. Метод последовательных вычитаний; $a, b$ — данные числа. $a$ В↑ $b$ В/О С/П.

|    |      |    |    |      |    |    |      |    |
|----|------|----|----|------|----|----|------|----|
| 00 | X→ПВ | 4L | 07 | П→ХА | 6— | 14 | ↔    | 14 |
| 01 | ↔    | 14 | 08 | C/P  | 50 | 15 | X→ПВ | 4L |
| 02 | X→ПА | 4— | 09 | FX<0 | 5C | 16 | П→ХА | 6— |
| 03 | П→ХВ | 6L | 10 | 16   | 16 | 17 | П→ХВ | 6L |
| 04 | —    | 11 | 11 | П→ХА | 6— | 18 | —    | 11 |
| 05 | FX=0 | 5E | 12 | П→ХВ | 6L | 19 | БП   | 51 |
| 06 | 09   | 09 | 13 | X→ПА | 4— | 20 | 02   | 02 |

Контрольные примеры. 6 В↑ 9 В/О С/П (12 с) «3», 18 В↑ 11 В/О С/П (25 с) «1».

4. Возможны два случая: 1) длина таблицы (массива) превышает емкость адресуемой памяти ПМК, элементы вводятся последовательно, одновременно выбирается наименьший; 2) емкость памяти ПМК достаточна для хранения данных.

### Программа ПОРЯДОК-1

|    |       |    |    |      |    |    |      |    |
|----|-------|----|----|------|----|----|------|----|
| 00 | X→ПА  | 4— | 09 | FO   | 25 | 17 | П→Х6 | 66 |
| 01 | ↔     | 14 | 10 | П→ХА | 6— | 18 | X→ПВ | 4L |
| 02 | 1     | 01 | 11 | ↔    | 14 | 19 | FL0  | 5Г |
| 03 | X→ПВ  | 4L | 12 | —    | 11 | 20 | 07   | 07 |
| 04 | X→П6  | 46 | 13 | FX≥0 | 59 | 21 | П→ХВ | 61 |
| 05 | —     | 11 | 14 | 19   | 19 | 22 | П→ХА | 6— |
| 06 | X→П0  | 40 | 15 | FBX  | 0  | 23 | Fп   | 20 |
| 07 | C/P   | 50 | 16 | X→ПА | 4— | 24 | C/P  | 50 |
| 08 | KП→Х6 | G6 |    |      |    |    |      |    |

В качестве наименьшего элемента в РА вводится  $a_1$ , одновременно в РВ запоминается его номер 1. При очередных остановах вводятся элементы таблицы  $a_2, a_3, \dots$ . В Р6 хранится число рассмотренных элементов. Содержимое регистров А и В формируется программой.

**Инструкция.** Последовательный ввод:  $n$  B↑  $a_1$  B/O C/P ( $n$  — число элементов,  $a_1$  — первый из них). После каждого останова введите очередное  $a_i$  начиная с  $i=2$ , C/P. После индикации **я** нажмите клавишу FO — на индикаторе наименьший элемент, повторно FO — номер этого элемента.

### Программа ПОРЯДОК-2

|    |        |    |    |      |    |    |      |    |
|----|--------|----|----|------|----|----|------|----|
| 00 | CX     | 0Г | 06 | —    | 11 | 11 | FL0  | 5Г |
| 01 | 1      | 01 | 07 | FX≥0 | 59 | 12 | 04   | 04 |
| 02 | 2      | 02 | 08 | 11   | 11 | 13 | П→ХД | 6Г |
| 03 | X→П0   | 40 | 09 | FBX  | 0  | 14 | Fп   | 20 |
| 04 | П→ХД   | 6Г | 10 | X→ПД | 4Г | 15 | C/P  | 50 |
| 05 | KП→ХВ↑ | ГЕ |    |      |    |    |      |    |

Таблица из  $n \leq 13$  элементов заранее введена в регистры Д, С, В, ... Ввод:  $n$  B/O C/P.

Для ПМК «Электроника» МК-61, МК-52, в языке которых нет команды KП→Х В↑, программа должна быть модифицирована:

|    |       |     |
|----|-------|-----|
| 04 | ПП    | 53  |
| 05 | 16    | 16  |
| 16 | ...   | ... |
| 17 | П→ХД  | 6Г  |
| 18 | П→Х0  | 60  |
| 19 | X→ПЕ  | 4Е  |
| 20 | FO    | 25  |
| 21 | KП→ХЕ | ГЕ  |
|    | B/O   | 52  |

### 5. n B/O C/P

|    |      |    |    |       |    |    |      |    |
|----|------|----|----|-------|----|----|------|----|
| 00 | X→П0 | 40 | 05 | X→П1  | 41 | 10 | FL0  | 5Г |
| 01 | CX   | 0Г | 06 | KП→Х1 | Г1 | 11 | 06   | 06 |
| 02 | X→П4 | 44 | 07 | FX≠0  | 57 | 12 | П→Х4 | 64 |
| 03 | 1    | 01 | 08 | —     | 10 | 10 | C/P  | 50 |
| 04 | 4    | 04 | 09 | KП→Х4 | Г4 |    |      |    |

Контрольный пример.  $n=7$ , 2 X→П7 1 /—/ X→П8 0 X→П9 П→ПА.  
3 X→ПВ Fп X→ПС 7,5 X→ПД 7 B/O C/P «5».

### 6. n B/O C/P

|    |      |    |    |                 |    |    |       |    |
|----|------|----|----|-----------------|----|----|-------|----|
| 00 | X→П0 | 40 | 06 | 7               | 07 | 12 | 14    | 14 |
| 01 | CX   | 0Г | 07 | KП→Х1           | Г1 | 13 | KП→Х4 | Г4 |
| 02 | X→П4 | 44 | 08 | FX <sup>2</sup> | 22 | 14 | FL0   | 5Г |
| 03 | 1    | 01 | 09 | FV              | 21 | 15 | 06    | 06 |
| 04 | 4    | 04 | 10 | —               | 11 | 16 | C/P   | 50 |
| 05 | X→П1 | 41 | 11 | FX<0            | 5C |    |       |    |

Контрольный пример. Тот же, что в программе 5. 7 B/O C/P П→Х4 «1».

### 7. n B/O C/P

|    |       |    |    |      |    |    |     |    |
|----|-------|----|----|------|----|----|-----|----|
| 00 | X→П0  | 40 | 05 | 7    | 07 | 10 | C/P | 50 |
| 01 | 1     | 01 | 06 | —    | 11 | 11 | FL0 | 5Г |
| 02 | 4     | 04 | 07 | FX=0 | 5E | 12 | 04  | 04 |
| 03 | X→П1  | 41 | 08 | 11   | 11 | 13 | CX  | 0Г |
| 04 | KП→Х1 | Г1 | 09 | 1    | 01 | 14 | C/P | 50 |

### 8. n B/O C/P

|    |       |    |    |      |    |    |      |    |
|----|-------|----|----|------|----|----|------|----|
| 00 | X→П0  | 40 | 10 | FX<0 | 5C | 20 | FX≥0 | 59 |
| 01 | 1     | 01 | 11 | 16   | 16 | 21 | 24   | 24 |
| 02 | 3     | 03 | 12 | FBX  | 0  | 22 | FBX  | 0  |
| 03 | X→П3  | 43 | 13 | X→П1 | 41 | 23 | X→П2 | 42 |
| 04 | П→ХД  | 6Г | 14 | БП   | 51 | 24 | FL0  | 5Г |
| 05 | X→П1  | 41 | 15 | 24   | 24 | 25 | 07   | 07 |
| 06 | X→П2  | 42 | 16 | FBX  | 0  | 26 | П→Х1 | 61 |
| 07 | П→Х1  | 61 | 17 | П→Х2 | 62 | 27 | П→Х2 | 62 |
| 08 | KП→Х3 | Г3 | 18 | —    | 14 | 28 | —    | 11 |
| 09 | —     | 11 | 19 | —    | 11 | 29 | C/P  | 50 |

Контрольный пример.  $n=6$ : 3 /—/ X→П8 6 X→П9 7 /—/ X→ПА 8 X→ПВ 7 X→ПС Fп X→ПД 6 B/O C/P (25 c) «15».

### 9. Данные ( $n \leq 9$ ) вводятся в РД, РС, РВ, РА, ..., n B/O C/P.

|    |       |    |    |      |    |    |        |    |
|----|-------|----|----|------|----|----|--------|----|
| 00 | X→П0  | 40 | 09 | FL0  | 5Г | 18 | KП→Х2  | Г2 |
| 01 | X→П1  | 41 | 10 | 07   | 07 | 19 | —      | 11 |
| 02 | 1     | 01 | 11 | П→Х1 | 61 | 20 | FX < 0 | 5C |
| 03 | 4     | 04 | 12 | :    | 13 | 21 | 23     | 23 |
| 04 | X→П2  | 42 | 13 | X→П3 | 43 | 22 | KП→Х4  | Г4 |
| 05 | CX    | 0Г | 14 | 1    | 01 | 23 | FL1    | 5L |
| 06 | X→П4  | 44 | 15 | 4    | 04 | 24 | 17     | 17 |
| 07 | KП→Х2 | Г2 | 16 | X→П2 | 42 | 25 | П→Х4   | 64 |
| 08 | +     | 10 | 17 | П→Х3 | 63 | 26 | C/P    | 50 |

### 10. Данные вводятся в РД, РС, РВ, РА, ..., n B/O C/P.

|    |       |    |    |      |    |    |        |    |
|----|-------|----|----|------|----|----|--------|----|
| 00 | X→П0  | 40 | 11 | —    | 11 | 22 | KП→Х2  | Г2 |
| 01 | X→П1  | 41 | 12 | FX<0 | 5C | 23 | —      | 11 |
| 02 | 1     | 01 | 13 | 16   | 16 | 24 | FX = 0 | 5E |
| 03 | 4     | 04 | 14 | FBX  | 0  | 25 | 27     | 27 |
| 04 | X→П2  | 42 | 15 | X→П3 | 43 | 26 | KП→Х4  | Г4 |
| 05 | CX    | 0Г | 16 | FL0  | 5Г | 27 | FL1    | 5L |
| 06 | X→П4  | 44 | 17 | 09   | 09 | 28 | 21     | 21 |
| 07 | KП→Х2 | Г2 | 18 | 1    | 01 | 29 | П→Х4   | 64 |
| 08 | X→П3  | 43 | 19 | 4    | 04 | 30 | C/P    | 50 |
| 09 | П→Х3  | 63 | 20 | X→П2 | 42 |    |        |    |
| 10 | KП→Х2 | Г2 | 21 | П→Х3 | 63 |    |        |    |

11. n В/О С/П ( $n \leq 10$ ).

|    |      |    |    |       |    |    |      |    |
|----|------|----|----|-------|----|----|------|----|
| 00 | X→P0 | 40 | 09 | CX    | 0Г | 17 | 11   | 11 |
| 01 | 1    | 01 | 10 | X→P3  | 43 | 18 | P→X3 | 63 |
| 02 | 3    | 03 | 11 | KП→X2 | Г2 | 19 | FX=0 | 5E |
| 03 | ↔    | 14 | 12 | FX=0  | 5E | 20 | 22   | 22 |
| 04 | —    | 11 | 13 | 16    | 16 | 21 | C/П  | 50 |
| 05 | X→P1 | 41 | 14 | P→X2  | 62 | 22 | P→X1 | 61 |
| 06 | 1    | 01 | 15 | X→P3  | 43 | 23 | —    | 11 |
| 07 | 4    | 04 | 16 | FL0   | 5Г | 24 | C/П  | 50 |
| 08 | X→P2 | 42 | —  | —     | —  | —  | —    | —  |

Контрольный пример. Массив 1, 0, 4, 5, 5. Вводим 1 X→P8 CX X→P9  
X→PВ 4 X→PA 5 X→PC X→PD 6 В/О С/П (15 с) «2».

12. n В/О С/П ( $n \leq 10$ ).

|    |      |    |    |        |    |    |      |    |
|----|------|----|----|--------|----|----|------|----|
| 00 | X→P0 | 40 | 09 | CX     | 0Г | 18 | FL0  | 5Г |
| 01 | 1    | 01 | 10 | X→P3   | 43 | 19 | 11   | 11 |
| 02 | 3    | 03 | 11 | KП→X2  | Г2 | 20 | P→X3 | 63 |
| 03 | ↔    | 14 | 12 | FX < 0 | 5C | 21 | FX=0 | 5E |
| 04 | —    | 11 | 13 | 18     | 18 | 22 | 24   | 24 |
| 05 | X→P1 | 41 | 14 | P→X2   | 62 | 23 | C/П  | 50 |
| 06 | 1    | 01 | 15 | X→P3   | 43 | 24 | P→X1 | 61 |
| 07 | 4    | 04 | 16 | БП     | 51 | 25 | —    | 11 |
| 08 | X→P2 | 42 | 17 | 20     | 20 | 26 | C/П  | 50 |

Контрольный пример.  $n=4$ ; массив 1, -2, -4, 5; 1 X→PA 2 /-/ X→PB  
4 /-/ X→PC 5 X→PD 4 В/О С/П (7 с) «3».

## 13. Программа СОВЕРШЕННЫЕ ЧИСЛА-1.

Распределение памяти

РА — значения переменной, ее исходное значение — данное число  $a$ ;

РВ — константа, данное число, используемое в качестве делимого;

Р4 — переменный делитель: 2, 3, 4, ...;

Р5 — сумма всех делителей  $a$ , вначале 0.

|    |       |    |    |       |    |    |        |    |
|----|-------|----|----|-------|----|----|--------|----|
| 00 | X→PA  | 4— | 14 | FX=0  | 5E | 27 | —      | 11 |
| 01 | X→PB  | 4L | 15 | 22    | 22 | 28 | FX ≥ 0 | 59 |
| 02 | CX    | 0Г | 16 | P→X4  | 64 | 29 | 06     | 06 |
| 03 | X→P5  | 45 | 17 | P→X7  | 67 | 30 | KП→X5  | Г5 |
| 04 | 2     | 02 | 18 | +     | 10 | 31 | P→X5   | 65 |
| 05 | X→P4  | 44 | 19 | P→X5  | 65 | 32 | P→XB   | 6L |
| 06 | P→XB  | 6L | 20 | +     | 10 | 33 | —      | 11 |
| 07 | P→X4  | 64 | 21 | X→P5  | 45 | 34 | FX=0   | 5E |
| 08 | :     | 13 | 22 | P→X7  | 67 | 35 | 38     | 38 |
| 09 | X→P7  | 47 | 23 | X→PA  | 4— | 36 | 1      | 01 |
| 10 | KП→X7 | Г7 | 24 | KП→X4 | Г4 | 37 | C/П    | 50 |
| 11 | F0    | 25 | 25 | P→X4  | 64 | 38 | 0      | 00 |
| 12 | P→X7  | 67 | 26 | P→XA  | 6— | 39 | C/П    | 50 |
| 13 | —     | 11 | —  | —     | —  | —  | —      | —  |

Инструкция. a ВО С/П. Индикация «1» означает, что число  $a$  совершенное, «0», что число несовершенное.

Контрольные примеры. 6 В/О С/П (12 с) «1», 28 В/О С/П (33 с) «1», 126 В/О С/П (1 мин 15 с) «0». За 1 ч работы ПМК не обнаружил совершенных чисел, кроме чисел 6 и 28, в пределах 100.

## Программа СОВЕРШЕННЫЕ ЧИСЛА-2

В программе «Совершенные числа-1» использован систематический перебор. Его можно ограничить, если воспользоваться теоремой Евклида: Если  $p=1+2+4+\dots+2^n=2^{n+1}-1$  простое, то  $2^np$  совершенное. Кстати, числа, имеющие много делителей, античные математики называли избыточными, а числа, имеющие мало делителей — недостаточными. В качестве меры использовалась сумма собственных делителей, которая сравнивалась с данным числом. Если сумма оказывалась меньше числа, то в числе видели «недостаток», больше — «избыток». В случае равенства число считалось совершенным. Есть данные о том, что совершенные числа были известны в древнем Вавилоне и Египте. Первая теорема о совершенных числах содержится в «Началах» Евклида (примерно 300 г. до н. э.). На ее основе составлена приведенная ниже программа.

|    |       |    |    |        |    |    |        |    |
|----|-------|----|----|--------|----|----|--------|----|
| 00 | X→P4  | 44 | 14 | KП→X1  | Г1 | 28 | 1      | 01 |
| 01 | KП→X4 | Г4 | 15 | P→X1   | 61 | 29 | 8      | 08 |
| 02 | P→X4  | 64 | 16 | F V    | 21 | 30 | 0      | 00 |
| 03 | X→P0  | 40 | 17 | X→P2   | 42 | 31 | X      | 12 |
| 04 | 1     | 01 | 18 | KП→X2  | Г2 | 32 | F sin  | 1C |
| 05 | B↑    | 0E | 19 | P→X2   | 62 | 33 | FX ≠ 0 | 57 |
| 06 | 2     | 02 | 20 | FX ≠ 0 | 57 | 34 | 01     | 01 |
| 07 | ×     | 12 | 21 | 37     | 37 | 35 | FL2    | 58 |
| 08 | FL0   | 5Г | 22 | 1      | 01 | 36 | 24     | 24 |
| 09 | 06    | 06 | 23 | X→P5   | 45 | 37 | P→XA   | 6— |
| 10 | X→PA  | 4— | 24 | KП→X5  | Г5 | 38 | P→X1   | 61 |
| 11 | 2     | 02 | 25 | P→X1   | 61 | 39 | X      | 12 |
| 12 | ×     | 12 | 26 | P→X5   | 65 | 40 | C/П    | 50 |
| 13 | X→P1  | 41 | 27 | :      | 13 | 41 | B/O    | 52 |

Инструкция. CX В/О С/П. Переключатель углов в положении Г. Индицируются только совершенные числа. Нажав клавишу С/П, продолжаем вычисления.

Контрольный пример. CX В/О С/П (8 с) «6», СР (15 с) «28», СР (50 с) «496», С/П (1 мин 30 с) «8128», С/П (13 мин) «33550336». Скорость вычислений повысилась на порядок по сравнению с предыдущей программой.

## 14. Программа ПОРЯДОК-1.

|    |       |    |    |        |    |    |       |    |
|----|-------|----|----|--------|----|----|-------|----|
| 00 | 1     | 01 | 11 | P→X1   | 61 | 22 | KП→XС | ГС |
| 01 | 1     | 01 | 12 | X→PD   | 4Г | 23 | X→PB  | 4L |
| 02 | X→P1  | 41 | 13 | P→XB   | 6L | 24 | KХ→PD | LG |
| 03 | P→X0  | 60 | 14 | KП→X3  | Г3 | 25 | ↔     | 14 |
| 04 | X→P2  | 42 | 15 | P→X3   | 63 | 26 | KХ→PС | LC |
| 05 | P→X1  | 61 | 16 | X→PС   | 4C | 27 | FL2   | 58 |
| 06 | 1     | 01 | 17 | FO     | 25 | 28 | 13    | 13 |
| 07 | —     | 11 | 18 | —      | 11 | 29 | FL0   | 5Г |
| 08 | X→P3  | 43 | 19 | FX < 0 | 5C | 30 | 03    | 03 |
| 09 | KП→X1 | Г1 | 20 | 27     | 27 | 31 | F π   | 20 |
| 10 | X→PB  | 4L | 21 | KП→ХД  | ГГ | 32 | C/П   | 50 |

Инструкция.  $n-1$  X→P0 В/О С/П «п». После окончания решения элементы упорядоченного массива извлекаются из регистров А, 9, 8, ...

Контрольный пример. Массив 4, 8, 2, 1, 7, 5, 3;  $n=7$ .  
4 X→P4, 8 X→P5, 2 X→P6, 1 X→P7, 7 X→P8, 5 X→P9, 3 X→PA,  
6 X→P0 В/О С/П (2 мин) «п».

Считываем упорядоченный массив: P→X4, «1», P→X5 «2», P→X6 «3»,  
P→X7 «4», P→X8 «5», P→X9 «7», P→XA «8».

## Структура программы

- 00—02 : запись в Р1 начального значения переменной, необходимой для перебора элементов во внешнем цикле;  
 03—08 : формирование начальных данных для внутреннего цикла;  
 09—30 : построение внешнего цикла;  
 13—28 : построение внутреннего цикла, повторяющегося несколько раз для каждого значения параметра внешнего цикла.

### 15. Программа ПОРЯДОК-2.

|    |       |    |    |        |    |    |       |    |
|----|-------|----|----|--------|----|----|-------|----|
| 00 | X→P3  | 43 | 25 | P→XA   | 6— | 50 | X→P1  | 41 |
| 01 | KП→X3 | G3 | 26 | —      | 11 | 51 | X→P2  | 42 |
| 02 | 1     | 01 | 27 | 1      | 01 | 52 | KП→X1 | G1 |
| 03 | 0     | 00 | 28 | 0      | 00 | 53 | 9     | 09 |
| 04 | X→P1  | 41 | 29 | X      | 12 | 54 | X→P0  | 40 |
| 05 | P→X1  | 61 | 30 | X→PС   | 4C | 55 | P→X9  | 69 |
| 06 | 1     | 01 | 31 | P→ХД   | 6Г | 56 | X→PС  | 4C |
| 07 | —     | 11 | 32 | P→ХС   | 6C | 57 | KП→X0 | G0 |
| 08 | X→P0  | 40 | 33 | —      | 11 | 58 | X→PA  | 4— |
| 09 | P→X3  | 63 | 34 | FX≥0   | 59 | 59 | KП→XA | G— |
| 10 | X→P2  | 42 | 35 | 45     | 45 | 60 | FO    | 25 |
| 11 | KП→X1 | G1 | 36 | P→XB   | 6L | 61 | P→XA  | 6— |
| 12 | X→PВ  | 4L | 37 | P→XA   | 6— | 62 | P→XC  | 6C |
| 13 | KП→XB | ГL | 38 | —      | 11 | 63 | ↔     | 14 |
| 14 | FO    | 25 | 39 | FX≥0   | 59 | 64 | —     | 11 |
| 15 | P→XB  | 6L | 40 | 45     | 45 | 65 | FX<0  | 5C |
| 16 | —     | 11 | 41 | KП→XB↑ | ГЕ | 66 | 69    | 69 |
| 17 | 1     | 01 | 42 | 1      | 01 | 67 | FBX   | 0  |
| 18 | 0     | 00 | 43 | +      | 10 | 68 | X→PС  | 4G |
| 19 | X     | 12 | 44 | KХ→PВ↑ | LE | 69 | FL1   | 5L |
| 20 | X→PД  | 4Г | 45 | FL2    | 58 | 70 | 57    | 57 |
| 21 | KП→X0 | G0 | 46 | 21     | 21 | 71 | P→X2  | 62 |
| 22 | X→PA  | 4— | 47 | FL3    | 5— | 72 | P→XC  | 6C |
| 23 | KП→XA | G— | 48 | 05     | 05 | 73 | —     | 11 |
| 24 | FO    | 25 | 49 | C/P    | 50 | 74 | C/P   | 50 |

Изменения, необходимые для ПМК «Электроника» МК-61, МК-52:

|    |      |    |    |       |    |    |       |    |
|----|------|----|----|-------|----|----|-------|----|
| 39 | PП   | 53 | 44 | KХ→PЕ | LE | 77 | P→X0  | 60 |
| 40 | 75   | 75 | —  | —     | —  | 78 | X→PЕ  | 4E |
| 41 | КНОП | 54 | 75 | FX≥0  | 59 | 79 | KП→XE | ГЕ |
| —  | —    | —  | 76 | 45    | 45 | 80 | B/O   | 52 |

**Инструкция.** Подготовьте двузначные числа вида  $\overline{1a_i}$ , где  $a_i$  — элементы данной таблицы,  $i \leq 6$  (верхняя черта означает, что имеются в виду именно двузначные числа, а не произведения). Введите их, разделяя цифры запятыми: 1,  $a_1$  X→P9, 1,  $a_2$  X→P8, ..., n B/O C/P. Примерно через 2 мин ПМК остановится. Наберите число n и, нажав C/P, запустите продолжение вычислений. Через 20 с на индикаторе появится результат — минимальное число элементов таблицы, которые необходимо исключить, чтобы она образовала возрастающую (неубывающую) последовательность.

**Контрольный пример.** Дано A: 6, 3, 1, 5, 7, 2 n=6.

1.6 X→P4, 1.3 X→P5, 1.1 X→P6, 1.5 X→P7, 1.7 X→P8, 1.2 X→P9 6 B/O C/P (2 мин). Нажмите 6 C/P (20 с) «3» — это число элементов, которое нужно исключить, чтобы образовалась возрастающая (неубывающая) последовательность. Итак, сохранить можно 6—3=3 числа; остаются 1, 5, 7; высчитываются 6, 3, 2.

## Структура программы

- 00—04 : формирование начальных данных для организации цикла по i;  
 05—10 : формирование цикла по j;  
 11—20 : в регистрах В и Д запоминаются цифры числа a<sub>i</sub>;  
 21—30 : в регистрах А и С запоминаются цифры числа a<sub>j</sub>;  
 31—48 : формирование и запись в регистры 9, 8, 7, ..., разделенных запятыми двузначных чисел: вторые цифры — это по-прежнему элементы данной таблицы, первые — число элементов таблицы;  
 49 : останов для ввода числа n;  
 50—69 : вычисление и запись в РС максимальной первой цифры в таблице K, т. е. наибольшего числа элементов таблицы, которые удается сохранить;  
 70—73 : вычисление и индикация требуемого в задаче числа элементов, которые исключаются из таблицы.

### 16. Программа РАССТАНОВКА ФЕРЗЕЙ для полей 5×5 и 7×7.

#### Распределение памяти

P4, P5 — текущие значения x и y — координаты клетки, занимаемой ферзем. Позиция ферза выводится на индикатор двузначным числом 10x+y.

|    |       |    |    |       |    |    |       |    |
|----|-------|----|----|-------|----|----|-------|----|
| 00 | X→PA  | 4— | 17 | +     | 10 | 34 | X→P4  | 44 |
| 01 | X→P0  | 40 | 18 | X→P7  | 47 | 35 | P→X4  | 64 |
| 02 | X→P3  | 43 | 19 | 0     | 00 | 36 | ВП    | 0C |
| 03 | X→P6  | 46 | 20 | X→P5  | 45 | 37 | 1     | 01 |
| 04 | KП→X3 | G3 | 21 | P→X4  | 64 | 38 | P→X5  | 65 |
| 05 | KП→X6 | G6 | 22 | 2     | 02 | 39 | +     | 10 |
| 06 | P→X7  | 67 | 23 | +     | 10 | 40 | C/P   | 50 |
| 07 | P→XA  | 6— | 24 | X→P4  | 44 | 41 | P→X7  | 67 |
| 08 | —     | 11 | 25 | KП→X5 | G5 | 42 | P→X3  | 63 |
| 09 | FX≥0  | 59 | 26 | P→X4  | 64 | 43 | —     | 11 |
| 10 | 12    | 12 | 27 | P→X6  | 66 | 44 | FX≥0  | 59 |
| 11 | K×    | 28 | 28 | —     | 11 | 45 | 47    | 47 |
| 12 | P→X7  | 67 | 29 | FX≥0  | 59 | 46 | KП→X4 | G4 |
| 13 | 1     | 01 | 30 | 35    | 35 | 47 | FLO   | 5Г |
| 14 | —     | 11 | 31 | P→X4  | 64 | 48 | 21    | 21 |
| 15 | X→P4  | 44 | 32 | P→XA  | 6— | 49 | Fп    | 20 |
| 16 | 2     | 02 | 33 | —     | 11 | 50 | C/P   | 50 |

**Инструкция.** m X→P7 n B/O C/P. n=5 или n=7; m+1 — абсолютная координата клетки, занимаемой ферзем на первой горизонтали; m=0, 1, ..., 4 для поля 5×5 и m=0, 1, ..., 6 для поля 7×7. В Р7 вводится любое целое число m от 0 до n—1. Через 12 с после запуска вычислений по командам n B/O C/P индицируется двузначное число: первая цифра — абсолютная координата клетки, занимаемой ферзем, вторая — ордината. Повторное нажатие C/P и ПМК продолжает вычисления, через 10 с на индикаторе появляются координаты следующей клетки в виде двузначного числа. После индикации координат n клеток, в которых размещаются ферзи, на индикаторе число n, что означает завершение расчета данного варианта расположения ферзей.

Наберите другое значение m из заданного промежутка: m X→P7 n B/O C/P и т. д. Если окажется, что m≥n, то ПМК через несколько секунд выдаст на индикатор ЕГГОГ, что означает ошибочный ход.

**Контрольный пример.** n=5, m=0: 0 X→P7 5 B/O C/P (12 с) «11», C/P (8 с) «32», C/P (8 с) «53», C/P (8 с) «24», C/P (8 с) «45», C/P (3 с) «». Возможны пять различных расстановок ферзей для m=0, 1, 2, 3, 4.

Пусть n=7, m=5: 5 X→P7, 7 B/O C/P «61», C/P «22», C/P «53», C/P «14», C/P «45», C/P «76», C/P «37», C/P «». Задача имеет семь решений для m=0, 1, ..., 6.

# Структура программы

- 00—25 : формирование счетчика циклов генерирования позиций ферзей в данной расстановке (начальное значение  $n$ ); константы  $n$ ,  $n-1$ ,  $n+1$  записываются в регистры А, З, Б, начальное значение  $m$  — абсциссы первого хода — в Р7. Если  $m \geq n$ , индицируется ЕГГОГ, в Р5 записывается 1 — ферзь устанавливается на первой горизонтали, с каждым новым ферзем содержимое Р5 увеличивается на единицу.
- 26—34 : если номер столбца, вычисленный по программе, т. е. значение  $x$ , хранящееся в Р4, больше  $n$ , то из номера (содержимого Р4) вычитается  $n$ .
- 35—40 : вычисление и индикация позиции очередного ферзя в данной расстановке.
- 42—46 : увеличение длины шага  $h_x$  на единицу, если содержимое Р7 не меньше  $n-1$ .
- 47—50 : проверка окончания цикла — если в данном варианте расставлено меньше чем  $n$  ферзей, то генерирование ходов продолжается; в противном случае индицируется число  $\pi$ .

алг ФЕРЗИ (нат  $n$ ,  $m$ , лит А, лит таб В [ $1:n$ ,  $1:n$ ])

```

    арг  $n$ ,  $m$ 
    рес А, В
    нач nat  $h$ ,  $i$ ,  $x$ ,  $y$ 
    если  $m > n$ 
        то А := «нет решения»
        иначе А := «есть решение»
        если  $m \leq n-2$ 
            то  $h := 2$ 
            иначе  $h := 3$ 
        все
         $x := m$ 
        для  $i$  от 1 до  $n$ 
        иц
             $y := i$ ; В [ $x$ ,  $y$ ] := «»
             $x := x + h$ 
            если  $x > n$ 
                то  $x := x - n$ 
            все
        кц
    все
кон

```

## Программа ФЕРЗИ 8×8

|    |      |    |    |      |    |    |      |    |
|----|------|----|----|------|----|----|------|----|
| 00 | CX   | 0Г | 11 | П→XA | 6— | 22 | 11   | 11 |
| 01 | 2    | 02 | 12 | П→XB | 6L | 23 | FL1  | 5L |
| 02 | 3    | 03 | 13 | +    | 10 | 24 | 27   | 27 |
| 03 | X→PA | 4— | 14 | X→PA | 4— | 25 | Fπ   | 20 |
| 04 | 1    | 01 | 15 | C/P  | 50 | 26 | C/P  | 50 |
| 05 | 9    | 09 | 16 | 9    | 09 | 27 | 1    | 01 |
| 06 | X→PB | 4L | 17 | 9    | 09 | 28 | 2    | 02 |
| 07 | 2    | 02 | 18 | ↔    | 14 | 29 | X→PB | 4L |
| 08 | X→P1 | 41 | 19 | —    | 11 | 30 | BП   | 51 |
| 09 | 2    | 02 | 20 | C/P  | 50 | 31 | 09   | 09 |
| 10 | X→P0 | 40 | 21 | FL0  | 5Г |    |      |    |

Инструкция. Через 5 с после В/О С/П индицируется 42 — ферзь занимает позицию Ф (4; 2); С/П (2 с) «57» и т. д. Индикация  $\pi$  означает, что задача решена

Контрольный пример. В/О С/П «42», С/П «57», С/П «61», С/П «38», С/П «73», С/П «26», С/П «85», С/П «14», С/П «π» (рис. 21).

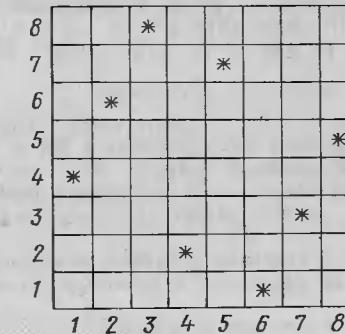


Рис. 21

Постройте самостоятельно программу, которая на основе заданных позиций ферзей 12, 24, 53, 61 генерировала бы полную расстановку для доски 8×8. Каждому числу  $a$ , описывающему позицию ферзя, в данной расстановке соответствует число  $24+a$ , также являющееся позицией ферзя. Запишите алгоритм в обозначениях УАЯ для программы «Ферзи 8×8».

## 17. Программа РЕКУРСИВНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

|    |        |    |    |        |    |    |        |    |
|----|--------|----|----|--------|----|----|--------|----|
| 00 | B↑     | 0E | 32 | 2      | 02 | 63 | X→P2   | 42 |
| 01 | FO     | 25 | 33 | :      | 13 | 64 | ↔      | 14 |
| 02 | X→PA   | 4— | 34 | X→PB   | 4L | 65 | X→P3   | 43 |
| 03 | 2      | 02 | 35 | Π→XA   | 6— | 66 | Π→X2   | 62 |
| 04 | :      | 13 | 36 | Π→XB   | 6L | 67 | X→PA   | 4— |
| 05 | B↑     | 0E | 37 | X      | 12 | 68 | Π→XB   | 6L |
| 06 | B↑     | 0E | 38 | 1      | 01 | 69 | 2      | 02 |
| 07 | 1      | 01 | 39 | —      | 11 | 70 | :      | 13 |
| 08 | 8      | 08 | 40 | FX = 0 | 5E | 71 | 9      | 09 |
| 09 | 0      | 00 | 41 | 46     | 46 | 72 | 0      | 00 |
| 10 | ×      | 12 | 42 | Π→X4   | 64 | 73 | ×      | 12 |
| 11 | F sin  | 1C | 43 | Π→X5   | 65 | 74 | F sin  | 1C |
| 12 | FX ≠ 0 | 57 | 44 | +      | 10 | 75 | FX = 0 | 5E |
| 13 | 01     | 01 | 45 | C/P    | 50 | 76 | 85     | 85 |
| 14 | 1      | 01 | 46 | Π→XA   | 6— | 77 | Π→X3   | 63 |
| 15 | X→P4   | 44 | 47 | 1      | 01 | 78 | X→PB   | 4L |
| 16 | CX     | 0Г | 48 | —      | 11 | 79 | Π→X5   | 65 |
| 17 | X→P5   | 45 | 49 | 2      | 02 | 80 | Π→X4   | 64 |
| 18 | X→PB   | 4L | 50 | :      | 13 | 81 | +      | 10 |
| 19 | Π→XA   | 6— | 51 | X→P2   | 42 | 82 | X→P5   | 45 |
| 20 | 1      | 01 | 52 | 1      | 01 | 83 | БП     | 51 |
| 21 | —      | 11 | 53 | +      | 10 | 84 | 26     | 26 |
| 22 | FX = 0 | 5E | 54 | X→P3   | 43 | 85 | Π→X3   | 63 |
| 23 | 26     | 26 | 55 | 9      | 09 | 86 | Π→XB   | 6L |
| 24 | 1      | 01 | 56 | 0      | 00 | 87 | Π→X5   | 65 |
| 25 | C/P    | 50 | 57 | ×      | 12 | 88 | Π→X4   | 64 |
| 26 | Π→XA   | 6— | 58 | F sin  | 1C | 89 | X→P5   | 45 |
| 27 | Π→XB   | 6L | 59 | FX ≠ 0 | 57 | 90 | +      | 10 |
| 28 | —      | 11 | 60 | 66     | 66 | 91 | X→P4   | 44 |
| 29 | FX < 0 | 5C | 61 | Π→X2   | 62 | 92 | БП     | 51 |
| 30 | 46     | 46 | 62 | Π→X3   | 63 | 93 | 26     | 26 |
| 31 | Π→XB   | 6L |    |        |    |    |        |    |

Инструкция. Переключатель углов в положении Г. п В/О С/П.  
 Контрольный пример. 10 В/О С/П (45 с) «3», 124 В/О С/П (1 мин 25 с) «5», 10890 В/О С/П (2 мин 50 с) «76», 1024 В/О С/П (45 с) «1».

### Структура программы

00—18: вычисление и запись в РА нечетного числа, полученного делением  $a_n$  на целую степень двойки, одновременно в РВ и Р5 заносятся нули в качестве начальных значений четного числа и его коэффициента, в Р4 — единица (коэффициент числа с нечетным номером).

19—25: проверка окончания расчета — если  $n$  — целая степень двойки, индируется единица.

26—30: сравнение слагаемых с нечетным и четным номерами.

31—34: анализ случая, когда слагаемое с нечетным номером меньше другого слагаемого.

35—45: проверка окончания, индикация результата.

46—65: случай, когда меньше слагаемое с нечетным номером, слагаемое преобразуется по формуле (3), его компоненты запоминаются в Р2 и Р3.

66—93: приведение подобных и возврат к началу цикла.

Проверка окончания проводится по равенству единице содержимого РА и РВ (команды 35—45). Оказывается, эти числа не независимы: когда в РА находится единица, то в РВ — двойка. Так как  $a_2 = a_1 = 1$ , то условие окончания можно упростить — достаточно, чтобы содержимое Р1 было равно единице. Это позволяет исключить команды 36, 37. Преобразуйте программу соответствующим образом.

### 18. Программа СУММЫ ЦИФР

|    |        |    |    |        |    |    |       |    |
|----|--------|----|----|--------|----|----|-------|----|
| 00 | 9      | 09 | 28 | P→X4   | 64 | 56 | X→P1  | 41 |
| 01 | X→P4   | 44 | 29 | 3      | 03 | 57 | БП    | 51 |
| 02 | P→X4   | 64 | 30 | F10*   | 15 | 58 | 61    | 61 |
| 03 | 9      | 09 | 31 | —      | 11 | 59 | 4     | 04 |
| 04 | +      | 10 | 32 | FX ≥ 0 | 59 | 60 | X→P1  | 41 |
| 05 | X→P4   | 44 | 33 | 02     | 02 | 61 | P→X3  | 63 |
| 06 | ПП     | 53 | 34 | Fπ     | 20 | 62 | 1     | 01 |
| 07 | 36     | 36 | 35 | C/П    | 50 | 63 | 0     | 00 |
| 08 | X→P4   | 4— | 36 | X→P3   | 43 | 64 | :     | 13 |
| 09 | 1      | 01 | 37 | CX     | 0Г | 65 | !     | 01 |
| 10 | 0      | 00 | 38 | X→PС   | 4C | 66 | +     | 10 |
| 11 | X→P2   | 42 | 39 | P→X3   | 63 | 67 | X→P3  | 43 |
| 12 | 8      | 08 | 40 | 2      | 02 | 68 | KП→X3 | Г3 |
| 13 | X→P0   | 40 | 41 | F10*   | 15 | 69 | F0    | 25 |
| 14 | KП→X2  | Г2 | 42 | —      | 11 | 70 | P→X3  | 63 |
| 15 | P→X2   | 62 | 43 | FX < 0 | 5C | 71 | —     | 11 |
| 16 | P→X4   | 64 | 44 | 49     | 49 | 72 | 1     | 01 |
| 17 | ×      | 12 | 45 | 2      | 02 | 73 | 1     | 11 |
| 18 | ПП     | 53 | 46 | X→P1   | 41 | 74 | 1     | 01 |
| 19 | 36     | 36 | 47 | БП     | 51 | 75 | 0     | 00 |
| 20 | P→XA   | 6— | 48 | 61     | 61 | 76 | ×     | 12 |
| 21 | —      | 11 | 49 | P→X3   | 63 | 77 | P→XC  | 6C |
| 22 | FX = 0 | 5E | 50 | 3      | 03 | 78 | +     | 10 |
| 23 | 28     | 28 | 51 | F10*   | 15 | 79 | X→PС  | 4C |
| 24 | FE0    | 5Г | 52 | —      | 11 | 80 | FL1   | 5L |
| 25 | 14     | 14 | 53 | FX < 0 | 5C | 81 | 61    | 61 |
| 26 | P→X4   | 64 | 54 | 59     | 59 | 82 | B/O   | 52 |
| 27 | C/П    | 50 | 55 | 3      | 03 |    |       |    |

Контрольный пример. CX В/О С/П (4 мин) «18», С/П (9 мин) «45», С/П (9 мин) «90», С/П (5 мин) «99», С/П (20 мин) «180», С/П (6 мин) «198», С/П (25 мин) «297», С/П (21 мин) «396», С/П и т. д.

### Структура программы

00—08: вычисление суммы цифр очередного числа, запоминается в РА.  
 09—35: подсчет суммы цифр произведения числа и  $k$  ( $k=2, 3, \dots, 9$ ), сравнение с содержимым РА.  
 36—82: подпрограмма вычисления суммы цифр числа, находящегося в РХ.

21. Ответ: нуль. Замените  $\sqrt[3]{2} + \sqrt[3]{20} - \sqrt[3]{25}$  корнем  $\sqrt[3]{(\sqrt[3]{2} + \sqrt[3]{20})^2 - \sqrt[3]{25}}$ .

22. Ответ: единица. При непосредственном вычислении на ПМК уменьшаемое выходит за пределы разрядной сетки, и результат округляется. В связи с этим необходимо предварительно преобразовать выражение.

23. Ответы: 3, 51, 6554.  $10^{n-1} \leq a^b \leq 10^n$ , после логарифмирования получаем  $n = [b \lg a] + 1$ .

### 24. Программа СУММА. n В/О С/П.

|    |      |    |    |      |    |    |     |    |
|----|------|----|----|------|----|----|-----|----|
| 00 | X→P0 | 40 | 03 | +    | 10 | 06 | FL0 | 5Г |
| 01 | CX   | 0Г | 04 | P→X0 | 60 | 07 | 02  | 02 |
| 02 | 1    | 01 | 05 | :    | 13 | 08 | C/П | 50 |

### 25. Программа ПОСЛЕДОВАТЕЛЬНОСТЬ. n В/О С/П

|    |      |    |    |     |    |    |     |    |
|----|------|----|----|-----|----|----|-----|----|
| 00 | B↑   | 0E | 06 | 2   | 02 | 12 | FL0 | 5Г |
| 01 | 1    | 01 | 07 | ×   | 12 | 13 | 06  | 06 |
| 02 | —    | 11 | 08 | 3   | 03 | 14 | Fπ  | 20 |
| 03 | X→P0 | 40 | 09 | +   | 10 | 15 | C/П | 50 |
| 04 | 1    | 01 | 10 | FV  | 21 |    |     |    |
| 05 | B↑   | 0E | 11 | C/П | 50 |    |     |    |

$x_2 = 2,2360679$ ,  $x_3 = 2,7335207$  и т. д.

После каждого останова, нажав С/П, запускаем продолжение вычислений. Появление на индикаторе π означает, что задача решена.

Программа упрощается, если требуется генерировать последовательность, не задаваясь заранее определенным числом ее членов.

### 1 В/О С/П

|    |    |    |    |    |    |    |     |    |
|----|----|----|----|----|----|----|-----|----|
| 00 | B↑ | 0E | 03 | 3  | 03 | 06 | C/П | 50 |
| 01 | 2  | 02 | 04 | +  | 10 | 07 | B/O | 52 |
| 02 | ×  | 12 | 05 | FV | 21 |    |     |    |

### 26а. 1 В/О С/П

|    |    |    |    |      |    |    |     |    |
|----|----|----|----|------|----|----|-----|----|
| 00 | B↑ | 0E | 03 | F1/X | 23 | 06 | C/П | 50 |
| 01 | 1  | 01 | 04 | 1    | 01 | 07 | B/O | 52 |
| 02 | +  | 10 | 05 | +    | 10 |    |     |    |

### 26б. 1 В/О С/П

|    |      |    |    |     |    |    |    |    |
|----|------|----|----|-----|----|----|----|----|
| 00 | F1/X | 23 | 03 | :   | 13 | 06 | БП | 51 |
| 01 | FBX  | 0  | 04 | +   | 10 | 07 | 00 | 00 |
| 02 | 2    | 02 | 05 | C/П | 50 |    |    |    |

### 27. 1 В/О С/П

|    |                 |    |    |                 |    |    |      |    |
|----|-----------------|----|----|-----------------|----|----|------|----|
| 00 | X→PС            | 4C | 07 | P→X4            | 64 | 14 | P→XB | 6L |
| 01 | X→P4            | 44 | 08 | FX <sup>2</sup> | 22 | 15 | ↔    | 14 |
| 02 | Fπ              | 20 | 09 | FX <sup>2</sup> | 22 | 16 | ;    | 13 |
| 03 | FX <sup>2</sup> | 22 | 10 | F1/X            | 23 | 17 | C/П  | 50 |
| 04 | FX <sup>2</sup> | 22 | 11 | P→XC            | 6C | 18 | БП   | 51 |
| 05 | X→PВ            | 4L | 12 | +               | 10 | 19 | 06   | 06 |
| 06 | KП→X4           | 4Г | 13 | X→PС            | 4C |    |      |    |

Возможности ПМК позволяют решить еще немало задач, но объем книги заставляет нас остановиться. Последняя задача посвящена использованию ПМК в эксперименте.

В одной школе собрали данные о росте и весе учеников четвертых классов (табл. 12). Используя таблицу, нетрудно вычислить средние значения этих по-

Таблица 12

| № п/п. | 4а   |      | 4б   |      | 4в   |      | 4г  |      |      |     |
|--------|------|------|------|------|------|------|-----|------|------|-----|
|        | Рост | Вес  | Рост | Вес  | Рост | Вес  | Пол | Рост | Вес  | Пол |
| 1      | 145  | 33,4 | 135  | 36,2 | 143  | 36,6 | ж   | 142  | 32,6 | ж   |
| 2      | 134  | 35,2 | 141  | 32,4 | 139  | 34,1 | ж   | 127  | 25,6 | м   |
| 3      | 127  | 26,9 | 136  | 32,9 | 130  | 33,4 | м   | 133  | 29,9 | м   |
| 4      | 142  | 30,2 | 140  | 36,2 | 142  | 32,1 | ж   | 142  | 35,6 | ж   |
| 5      | 137  | 32,9 | 136  | 37,9 | 139  | 34,2 | ж   | 133  | 30,2 | ж   |
| 6      | 139  | 32,6 | 145  | 36,4 | 138  | 33,1 | ж   | 143  | 44,8 | м   |
| 7      | 142  | 32,0 | 135  | 27,4 | 139  | 36,4 | м   | 129  | 30,6 | м   |
| 8      | 143  | 33,2 | 136  | 33,2 | 128  | 27,4 | м   | 133  | 30,9 | ж   |
| 9      | 150  | 32,2 | 151  | 41,0 | 134  | 32,3 | м   | 134  | 29,4 | м   |
| 10     | 137  | 29,9 | 134  | 30,1 | 132  | 27,4 | м   | 138  | 32,4 | м   |
| 11     | 134  | 29,9 | 138  | 32,8 | 135  | 34,2 | ж   | 136  | 30,5 | м   |
| 12     | 140  | 32,0 | 142  | 36,0 | 139  | 34,6 | м   | 138  | 42,0 | ж   |
| 13     | 143  | 42,3 | 139  | 37,4 | 151  | 39,0 | ж   | 136  | 30,2 | м   |
| 14     | 142  | 37,2 | 126  | 27,2 | 130  | 29,8 | ж   | 132  | 28,4 | м   |
| 15     | 136  | 31,4 | 140  | 35,4 | 137  | 32,1 | ж   | 147  | 38,2 | ж   |
| 16     | 142  | 34,2 | 134  | 28,6 | 138  | 36,4 | м   | 137  | 25,4 | м   |
| 17     | 137  | 31,9 | 142  | 33,2 | 133  | 30,1 | ж   | 139  | 34,6 | м   |
| 18     | 140  | 33,0 | 136  | 32,4 | 132  | 30,4 | ж   | 135  | 32,2 | м   |
| 19     | 132  | 32,2 | 144  | 32,8 | 133  | 26,7 | м   | 144  | 42,4 | м   |
| 20     | 134  | 30,0 | 132  | 28,2 | 134  | 32,9 | ж   | 136  | 31,2 | ж   |
| 21     | 141  | 34,0 | 129  | 26,4 | 135  | 34,1 | ж   | 133  | 29,8 | м   |
| 22     | 135  | 31,6 | 133  | 26,9 | 156  | 51,0 | м   | 140  | 34,6 | м   |
| 23     | 132  | 34,4 | 136  | 32,8 | 134  | 32,2 | м   | 138  | 32,6 | ж   |
| 24     | 141  | 33,2 | 137  | 30,8 | 132  | 27,4 | м   | 137  | 29,9 | ж   |
| 25     | 134  | 28,0 | 132  | 31,8 | 144  | 46,3 | ж   | 135  | 30,9 | ж   |
| 26     | 138  | 31,8 | 133  | 30,4 | 144  | 46,3 | ж   | 140  | 33,6 | ж   |
| 27     | 132  | 29,4 | 135  | 30,1 | 130  | 27,9 | ж   | 137  | 32,4 | ж   |
| 28     | 138  | 39,1 | 139  | 29,9 | 134  | 35,1 | м   | 135  | 37,4 | м   |
| 29     | 146  | 38,0 | 145  | 46,8 | 133  | 35,0 | м   | 140  | 38,9 | м   |
| 30     | 134  | 27,2 | 135  | 31,2 | 145  | 40,0 | ж   | 135  | 29,4 | ж   |
| 31     | 129  | 30,3 | 132  | 33,2 |      |      |     | 134  | 32,4 | м   |

казателей. Однако на практике не менее важно знать, например, какова вероятность попадания роста школьника в некоторую окрестность среднего значения. Эти данные пригодились бы прежде всего швейникам — они смогли бы оценить потребность в одежде различных размеров. Эти задачи легко решить с помощью ПМК.

Достаточно точно описывает распределение вероятностей случайной величины гауссовский закон распределения  $\Phi(x) = (\sigma\sqrt{2\pi}) \exp[-(x-\bar{x})^2/(2\sigma^2)]$ , где среднее значение переменной  $\bar{x} = \sum_{i=1}^n x_i/n$  ( $n$  — число значений случайной величи-

ны  $X=x_1, x_2, \dots, x_n$ );  $\sigma^2 = D = (\sum_{i=1}^n x_i^2 - n\bar{x}^2)/(n-1)$ . Эта величина является нижней границей вероятности того, что  $\bar{x}$  принадлежит интервалу  $(x, x+1)$  для

$x < \bar{x}$  или  $(x-1, x)$  для  $x > \bar{x}$ . Площадь фигуры, ограниченной кривой  $\Phi(x)$  (гауссианой), равна 1, т. е.  $\Phi(x) < 1$  для всех действительных  $x$ , за исключением  $x = \bar{x}$ . Чтобы проверить, насколько точно гауссовский закон описывает реальное распределение, значение  $\Phi$  сравнивают с величиной  $\sqrt{\pi/(2n(n-1))} \sum_{i=1}^n |x_i - \bar{x}|$ .

Итак, прежде всего необходимо вычислить  $\Phi(x)$ . Для этого будем вводить элементы массива данных  $X$  в цикле, одновременно вычисляя и накапливая в памяти суммы  $\Sigma x_i$  и  $\Sigma x^2_i$ . После того как ввод данных завершен, нетрудно подсчитать  $\bar{x}$ ,  $\sigma^2$ ,  $\sigma\sqrt{2\pi}$ , необходимые для вычисления  $\Phi(x)$  при данном значении  $x$ . Остается лишь отыскать нижнюю границу вероятности того, что случайная величина попадет в некоторый интервал  $(\bar{x}-l, \bar{x}+l)$ , симметричный относительно  $\bar{x}$ . Для этого достаточно подсчитать сумму

$$\sum_{i=0}^{l-1} \Phi(\bar{x}-l+i) = \Phi(\bar{x}-l) + \Phi(\bar{x}-l+1) + \Phi(\bar{x}-l+2) + \dots + \Phi(\bar{x}-1).$$

Поскольку  $\Phi(x)$  симметрична относительно  $x=\bar{x}$ , полученную сумму умножают на 2. Составим программу для решения задачи на ПМК.

### Программа ДОВЕРИТЕЛЬНЫЙ ИНТЕРВАЛ $(\bar{x}-l, \bar{x}+l)$

#### Распределение памяти

P0 — счетчик для организованного ввода элементов массива  $X$ , начальное значение  $n$ ;

P4 —  $l$  — радиус интервала;

P8 —  $2\sigma^2$ ;

P9 — накопитель величины  $\Sigma\Phi(x)$ , начальное значение 0;

PA —  $\sqrt{2\pi\sigma^2}$ ;

PB — на первом этапе  $\Sigma x^2_i$ , на втором  $\sigma^2$ , начальное значение 0;

PC — на первом этапе  $\Sigma x_i$ , на втором  $\bar{x}$ , начальное значение 0;

PD —  $n$  — константа.

|    |                 |    |    |      |    |    |                 |    |
|----|-----------------|----|----|------|----|----|-----------------|----|
| 00 | X→P0            | 40 | 25 | —    | 11 | 50 | +               | 10 |
| 01 | X→PД            | 4Г | 26 | П→ХД | 6Г | 51 | X→P4            | 44 |
| 02 | СХ              | 0Г | 27 | 1    | 01 | 52 | П→ХС            | 6C |
| 03 | X→ПВ            | 4L | 28 | —    | 11 | 53 | —               | 11 |
| 04 | X→ПС            | 4C | 29 | :    | 13 | 54 | FX=0            | 5E |
| 05 | C/P             | 50 | 30 | X→ПВ | 4L | 55 | 64              | 64 |
| 06 | П→ХС            | 6C | 31 | 2    | 02 | 56 | П→X9            | 69 |
| 07 | ↔               | 14 | 32 | ×    | 12 | 57 | П→ХА            | 6— |
| 08 | +               | 10 | 33 | X→П8 | 48 | 58 | :               | 13 |
| 09 | X→ПС            | 4C | 34 | Fπ   | 20 | 59 | 2               | 02 |
| 10 | FBX             | 0  | 35 | ×    | 12 | 60 | ×               | 12 |
| 11 | FX <sup>2</sup> | 22 | 36 | Fγ   | 21 | 61 | C/P             | 50 |
| 12 | П→ХВ            | 6L | 37 | X→ПА | 4— | 62 | П→ХС            | 6C |
| 13 | +               | 10 | 38 | Fπ   | 20 | 63 | C/P             | 50 |
| 14 | X→ПВ            | 4L | 39 | C/P  | 50 | 64 | FX <sup>2</sup> | 22 |
| 15 | FL0             | 5Г | 40 | П→ХС | 6C | 65 | П→X8            | 68 |
| 16 | 05              | 05 | 41 | ↔    | 14 | 66 | :               | 13 |
| 17 | П→ХВ            | 6L | 42 | —    | 11 | 67 | /—              | 0L |
| 18 | П→ХС            | 6C | 43 | 1    | 01 | 68 | Fe <sup>x</sup> | 16 |
| 19 | П→ХД            | 6Г | 44 | —    | 11 | 69 | П→X9            | 69 |
| 20 | ·               | 13 | 45 | X→П4 | 44 | 70 | +               | 10 |
| 21 | X→ПС            | 4C | 46 | CX   | 0Г | 71 | X→П9            | 49 |
| 22 | FX <sup>2</sup> | 22 | 47 | X→П9 | 49 | 72 | БП              | 51 |
| 23 | П→ХД            | 6Г | 48 | П→Х4 | 64 | 73 | 48              | 48 |
| 24 | X               | 12 | 49 | 1    | 01 |    |                 |    |

Инструкция. Ввод:  $n$  В/О С/П (2 с) «0»,  $x_1$  С/П (4 с),  $x_2$  С/П..., пока не будут введены все  $n$  значений,  $x_n$  С/П (12 с) «л» — это означает, что  $\bar{x}$ ,  $\sigma^2$ ,  $\sigma\sqrt{2\pi}$  вычислены и хранятся в памяти. Введите радиус интервала  $l$  С/П и ожидайте, пока ПМК не рассчитает искомую вероятность того, что случайная величина попадет в интервал  $(\bar{x}-l, \bar{x}+l)$ . Нажмите С/П и на индикаторе появится  $\bar{x}$ . Теперь уже легко записать интервал, для которого вычислена вероятность попадания случайной величины.

Контрольные примеры. По данным табл. 12 для разных классов, а также порознь для мальчиков и девочек найдем нижнюю границу попадания значений случайной величины Рост в интервал данного радиуса.

Рассмотрим объединенный массив Рост для всех четвероклассников:  $n=32+31+30+35=128$  ( $\sigma_p=5.10$ ;  $\sigma_k=5.20$  — гауссовское распределение).

1.  $l=10$ ; 128 В/О С/П (2 с) «0» 145 С/П (4 с) «210225» 134 С/П (4 с) «38981» и т. д. — вводятся все значения  $x_i$ . После ввода  $x_{128}=139$  С/П (12 с) «л» 10 С/П (1 мин 25 с) «0.8807586» С/П (1 с) « $\bar{x}=137.14062$ ». Результат: не менее 88% четвероклассников имеют рост от  $\bar{x}-l=137-10=127$  (см) до  $\bar{x}+l=137+10=147$  (см).

2.  $l=6$ . Так как состав учащихся не изменился, нет нужды заново формировать значения  $\bar{x}$  и  $\sigma^2$ . В связи с этим обратимся к той части программы, которая следует за индикацией «л». 6 БП 40 С/П (53 с) «0.71448056» С/П (1 с) «137» — примерно у 70% четвероклассников рост находится в пределах 131—143 (см).

### Структура программы

00—04 : ввод значения  $n$  и очистка регистров, в которых будут накапливаться суммы;  
 05 : останов для ввода  $x_i$ ;  
 06—14 : вычисление  $\Sigma x_i$  и  $\Sigma x_i^2$  и запись их в РС и РВ;  
 15, 16 : проверка окончания ввода данных, возврат на повторение цикла ввода;  
 17—30 : вычисление  $\bar{x}$  и  $\sigma^2$  и запись их в РС и РВ;  
 31—37 : вычисление  $2\sigma^2$ ,  $\sigma\sqrt{2\pi}$  и запись их в Р8 и РА;  
 38, 39 : индикация л означает, что все вспомогательные величины рассчитаны и можно ввести радиус интервала, вероятность попадания в который мы ищем;  
 40—45 : в Р4 записывается  $\bar{x}-l-1$ ;  
 46, 47 : очистка накопителя вероятностей — Р9;  
 48—63 : проверка окончания цикла, вывод результатов;  
 64—73 : вычисление  $\exp[-(x-\bar{x})^2/2\sigma^2]$ , возврат на повторение цикла суммирования вероятностей.

Покажите, что более половины четвероклассников имеют рост до 141 см, 88% мальчиков 126—146 см, 90% девочек 128—148 см. Проведите аналогичное исследование по параметру Вес. На основе полученных результатов сформулируйте рекомендации швейной фабрике. Обработайте данные учащихся вашего класса для параметров Рост и Вес, используя программу.

### РЕКОМЕНДУЕМЫЙ СПИСОК ЛИТЕРАТУРЫ

- Белый Ю. А. Считывающая микроэлектроника. — М.: Наука, 1984. — 119 с.
- Белый Ю. А. Электронные микрокалькуляторы и техника вычислений. — М.: Знание, 1981.
- Блох А. Ш., Павловский А. Н., Пенкрант В. В. Программирование на микрокалькуляторах. — Минск: Вышэйшая школа, 1981. — 192 с.
- Данилов И. Д. Секреты программируемого микрокалькулятора. — М.: Наука, 1986. — 160 с. — (Б-ка «Квант», вып. 55).
- Дьяконов В. П. Справочник по расчетам на микрокалькуляторах. — М.: Наука, 1985. — 224 с.
- Еланечников В. А. Прикладные программы для расчета специальных функций на микрокалькуляторе «Электроника Б3-34». Препринт, М.: 1982. — 31 с. — ИРЭ, АН СССР, № 27.
- Жамкоган В. М., Тер-Исаакян Н. Л. Численное интегрирование на программируемых микрокалькуляторах. — Препринт. — Ереван, 1984. — 16 с. — (Ереван, физ. ин-т, № 688 (3)—84).
- Кройль Г. Что умеет мой микрокалькулятор? — М.: Мир, 1981.
- Кузнецов Е. Ю., Острецов Б. В., Минкин А. К., Егорова Ю. Н. Микрокалькуляторы: Технические и конструктивные характеристики. — М.: Радио и связь, 1984. — 127 с. — (Массовая библиотека инженера. Вып. 40.).
- Микрокалькуляторы в играх и задачах. — М.: Наука, 1986. — 160 с. — (Сер. Кибернетика — Неограниченные возможности и возможные ограничения).
- Оксман В. М. Применение микрокалькуляторов при обучении математике в средних ПТУ: Методическое пособие. — М.: Высшая школа, 1984.
- Основы информатики и вычислительной техники: Пробное учебное пособие для средних учебных заведений. — В 2-х ч./Под ред. А. П. Ершова и В. М. Монахова. — М.: Просвещение. — Ч. I, 1985; Ч. II, 1986.
- Поснов М. Ф., Поснова Н. Н. Микрокалькуляторы с программным управлением в учебной лаборатории. — Минск: Гос. ун-т, 1985.
- Нужнаев Ю. В., Данилов И. Д. Микрокалькуляторы для всех. — М.: Знание, 1986. — 192 с.
- Романовский Т. Б. Микрокалькуляторы в рассказах и играх. — Рига: Зиннатне, 1984. — 120 с.
- Славин Г. В. Прикладные программы для программируемого микрокалькулятора «Электроника Б3-34». — Тарту: ТГУ, 1983. — 86 с.
- Славин Г. В. Программирование на программируемых микрокалькуляторах типа «Электроника Б3-34». — Таллин: Валгус, 1984. — 129 с.
- Трохименко Я. К. Игры с микроЭВМ. — Киев: Техника, 1986. — 120 с.
- Трохименко Я. К., Любич Ф. Д. Инженерные расчеты на программируемых микрокалькуляторах. — Киев: Техника, 1985. — 328 с. — (Б-ка инженера).
- Трохименко Я. К., Любич Ф. Д. Микрокалькулятор, ваш ход! — М.: Радио и связь, 1985. — 224 с.
- Трохименко Я. К. Программирование микрокалькуляторов «Электроника МК-52» и «Электроника МК-61». — Киев, Техника, 1987.
- Трохименко Я. К., Любич Ф. Л. Радиотехнические расчеты на микрокалькуляторах: Справочник. — 2-е изд., перераб. и доп. — М.: Радио и связь, 1988.
- Францевич Л. Н. Обработка результатов биологических экспериментов на микроЭВМ «Электроника Б3-21». — Киев: Наукова думка, 1979. — 91 с.

24. Цветков А. Н., Епанечников В. А. Прикладные программы для микроЭВМ «Электроника Б3-34», «Электроника МК-56», «Электроника МК-54». — М.: Финансы и статистика, 1984. — 175 с.
25. Чакань А. Что умеет карманный ЭВМ? Пер. с венгер. — М.: Радио и связь, 1982. — 114 с.
26. Шапиро С. И. Решение логических и игровых задач. Логико-психологические этюды. — М.: Радио и связь, 1984. —
27. Наука и жизнь. Цикл «Школа начинающего программиста». Начало 1985, № 6; «Человек с микрокалькулятором». Начало 1983, № 10.
28. Техника — молодежи. Рубрика «Для всех профессий». — Начало 1985, № 2. «Клуб электронных игр». Начало 1985, № 6.
29. Химия и жизнь. Цикл «МикроЭВМ для химиков». — 1984. — № 9—12..
30. Белый Ю. А. Работа с программируемым микрокалькулятором//Земля и Вселенная. — 1984. — № 2. — С. 101—106.
31. Богуславский А. А., Замаховский М. П., Симонова Г. А. Использование программируемого микрокалькулятора при обучении физике//Физика в школе. — 1985. — № 5. — С. 46—48.
32. Ершов А. Решение задач с применением программируемого микрокалькулятора «Электроника Б3-34»//Информатика и образование. — 1986. — № 3. — С. 61—72.
33. Ионов Г. Н. Микрокалькулятор как средство контроля и обучения//Математика в школе. — 1984. — № 3. — С. 51—54.
34. Сергеев В. К., Гасимов Р. К., Сергеев В. Ф. Настольный программируемый микрокалькулятор «Электроника МК-56»//Изв. вузов. Радиоэлектроника. — 1981. — Т. 24, № 12. — С. 79—80.
35. Финк Л. М. Целочисленные операции с помощью программируемых микрокалькуляторов//Изв. вузов. Радиоэлектроника. — 1984. — Т. 27, № 6. — С. 55—60.

## ОГЛАВЛЕНИЕ

|                                                                                             |    |
|---------------------------------------------------------------------------------------------|----|
| Предисловие . . . . .                                                                       | 3  |
| От авторов . . . . .                                                                        | 4  |
| Введение . . . . .                                                                          | 8  |
| <b>Часть I. ОСНОВЫ ПРОГРАММИРОВАНИЯ . . . . .</b>                                           |    |
| <b>Глава 1. Программируемый микрокалькулятор «Электроника МК-56» . . . . .</b>              | 13 |
| 1.1. Принцип работы . . . . .                                                               | 13 |
| 1.2. Ввод и представление чисел. Обращение к регистрам памяти . . . . .                     | 14 |
| 1.3. Организация вычислений в стеке . . . . .                                               | 16 |
| 1.4. Что нельзя вычислить на ПМК . . . . .                                                  | 20 |
| 1.5. Режим программирования. Ввод программы и ее отладка . . . . .                          | 21 |
| 1.6. Адресация в ПМК . . . . .                                                              | 25 |
| 1.7. Составные операторы . . . . .                                                          | 27 |
| 1.8. Программируемый микрокалькулятор и учебно-алгоритмический язык . . . . .               | 30 |
| <b>Глава 2. Методы программирования . . . . .</b>                                           | 38 |
| 2.1. Базовые программы . . . . .                                                            | 38 |
| 2.2. Организация циклов с помощью счетчиков и операторов условного перехода . . . . .       | 44 |
| 2.3. Объединение программ. Оптимизация составной программы . . . . .                        | 48 |
| 2.4. Циклы и массивы . . . . .                                                              | 50 |
| 2.5. Поэтапная обработка данных . . . . .                                                   | 55 |
| 2.6. Подпрограммы . . . . .                                                                 | 59 |
| 2.7. Оператор В/О в организации цикла. Вложенные циклы . . . . .                            | 64 |
| 2.8. Циклы, используемые при решении логических задач . . . . .                             | 66 |
| 2.9. Выборочное обращение. Вложенные подпрограммы. Пошаговое исполнение программы . . . . . | 68 |
| 2.10. Циклы и подпрограммы в программах численного интегрирования . . . . .                 | 72 |
| <b>Глава 3. В мире последовательностей . . . . .</b>                                        | 77 |
| 3.1. Предел последовательности . . . . .                                                    | 77 |
| 3.2. Рекуррентные последовательности и число π . . . . .                                    | 80 |
| 3.3. Последовательности, генерирующие π и не связанные с окружностью . . . . .              | 84 |
| <b>Глава 4. Современная микропроцессорная техника . . . . .</b>                             | 86 |
| 4.1. Что могут и чего не могут новые микрокалькуляторы . . . . .                            | 86 |
| 4.2. Работа с устройством памяти ПМК «Электроника МК-52» . . . . .                          | 89 |
| 4.3. МикроЭВМ «Электроника МК-85» . . . . .                                                 | 92 |
| 4.4. МикроЭВМ «Микроша» . . . . .                                                           | 95 |
| 4.5. МикроЭВМ «Электроника БК-0010» . . . . .                                               | 96 |

## Часть II. ПРОГРАММИРОВАНИЕ РЕШЕНИЯ ИГРОВЫХ И ЛОГИЧЕСКИХ ЗАДАЧ

98

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| Глава 1. Человек и машина в противостоянии и союзе . . . . .             | 98  |
| 1.1. Вращающийся бочонок . . . . .                                       | 98  |
| 1.2. Дорожка . . . . .                                                   | 106 |
| 1.3. Дорожка, модификация . . . . .                                      | 112 |
| 1.4. Один ферзь и два игрока . . . . .                                   | 118 |
| 1.5. Ханойская башня . . . . .                                           | 120 |
| Послесловие к главе . . . . .                                            | 124 |
| Глава 2. Решение задач в диалоговом режиме человек — ПМК . . . . .       | 124 |
| 2.1. Шестизначное число . . . . .                                        | 125 |
| 2.2. Человек и ПМК совместно расшифровывают числа . . . . .              | 130 |
| 2.3. Эксперименты с двухзначными числами . . . . .                       | 136 |
| 2.4. ПМК отгадывает цифры . . . . .                                      | 138 |
| 2.5. Быки и коровы . . . . .                                             | 142 |
| 2.6. Логика . . . . .                                                    | 146 |
| 2.7. Ловля блохи (Алгоритм против случая) . . . . .                      | 152 |
| 2.8. Рейтинг . . . . .                                                   | 154 |
| Послесловие к главе . . . . .                                            | 158 |
| Глава 3. Продолжение диалога . . . . .                                   | 159 |
| 3.1. Магические квадраты . . . . .                                       | 159 |
| 3.2. Крестики-нолики . . . . .                                           | 164 |
| 3.3. Крестики-нолики, вариация . . . . .                                 | 167 |
| 3.4. Тригекс . . . . .                                                   | 171 |
| 3.5. Бридж-ит . . . . .                                                  | 174 |
| 3.6. Игра «5» . . . . .                                                  | 180 |
| 3.7. Робот находит путь к цели . . . . .                                 | 184 |
| 3.8. Клетки на прямой . . . . .                                          | 187 |
| Послесловие к главе . . . . .                                            | 189 |
| Глава 4. Программируемые микрокалькуляторы в обучении . . . . .          | 191 |
| 4.1. Задачи двоичной логики (Обучение логическому рассуждению) . . . . . | 191 |
| 4.2. Урок биологии (Условный рефлекс) . . . . .                          | 197 |
| 4.3. Психологический эксперимент. Проверка и тренировка памяти . . . . . | 202 |
| 4.4. Контроль знаний . . . . .                                           | 205 |
| 4.5. Правописание приставок на з и с . . . . .                           | 208 |
| 4.6. Логические функции и программы контроля знаний . . . . .            | 211 |
| Глава 5. Занимательные задачи . . . . .                                  | 213 |
| 5.1. Размен без сдачи . . . . .                                          | 213 |
| 5.2. В многоквартирном доме . . . . .                                    | 219 |
| 5.3. Головоломка автомобилиста . . . . .                                 | 221 |
| 5.4. Выборы . . . . .                                                    | 223 |
| 5.5. День недели. Биоритмы . . . . .                                     | 226 |
| 5.6. Миссионеры и людоеды . . . . .                                      | 230 |
| Вместо заключения. К читателю . . . . .                                  | 234 |
| Приложение . . . . .                                                     | 234 |
| Рекомендуемый список литературы . . . . .                                | 253 |

