

## Глава XII. Примеры решения задач повышенной трудности

Experimentia est optima rerum magistra!

—Латинская пословица

Программированию нельзя научить при помощи выполняемых под команду массовых вольных упражнений... В нем нужно упражняться индивидуально, как в хождении на лыжах и вождении автомобиля.

—Ф.Л.Бауэр, Р.Гнац, У.Хилл

Преподавание программирования — дело почти безнадёжное, а его изучение — непосильный труд. Преподаватель может всячески возиться со студентами, читать лекции, делать критические замечания, направлять по верному пути. Студент может все тщательно записывать, запоминать, читать, сдавать зачёты, дискутировать хоть до двух ночи.

**Но все усилия тщетны, если студент не будет практиковаться в написании программ, поскольку навык программирования даётся только практикой.**

Более того, учиться надо на «настоящих» программах, а не на упрощённых примерах, вроде тех, которыми изобилует большинство руководств по языкам программирования.

Перечислим те способности, которые жизненно необходимы *всякому программисту* [17].

1. Способность читать и понимать описание поставленной задачи, улавливать пожелания того, кто её ставит (что не всегда легко, так как и задачи, и те, кто их ставит, часто отличаются именно неуловимостью). Учтите третий закон Чизхолма: «Любую цель люди понимают иначе, чем человек, её указующий».
2. Способность чётко видеть действительные трудности и отбрасывать все, не относящееся к делу.
3. Способность выявить все случаи, где можно применить теорию, самостоятельно решиться на её применение или обратиться за советом к специалисту.
4. Способность разбить задачу на ряд обозримых независимых частей и понять взаимосвязи этих частей.
5. Способность оценивать эффективность предлагаемых решений с точки зрения затрат на программирование, машинных ресурсов и удовлетворения потребностей пользователя и находить приемлемый компромисс между этими видами эффективности.
6. Способность объединять множество частных решений воедино, получая при этом чёткое и изящное решение всей задачи.
7. Способность выражать решения на простом и понятном языке. Естественный это язык или искусственный — роли не играет, важно лишь, чтобы правильность решения была ясна и людям, и компьютеру.
8. И наконец, способность при неудаче подавить самолюбие и поискать другой подход (или даже другую задачу).

Способности эти, как видим, столь сложны и многообразны, что приобрести их можно *только на практике*. Накапливая опыт, студент постепенно приобретает качества, необходимые программисту.

Академик А.П.Ершов говорил, что «программист должен обладать способностью первоклассного математика к абстрактному и логическому мышлению, в сочетании с эдисоновским талантом соорудить все, что угодно из нуля и единицы. Он должен сочетать аккуратность бухгалтера с проникательностью разведчика, фантазию автора детективных романов с трезвой практичностью экономиста. А кроме того, программист должен иметь вкус к коллективной работе, понимать интересы пользователя и многое другое».

### XII.1. Задачи

Задачи 12–40 представляют собою задачи «со звёздочкой» из школьного учебника [11]. Остальные задачи (41–48) — это задачи олимпиадного типа по информатике. Программы составлены на диалоговом языке программирования **MSX BASIC**. Часть из них использует алгоритмы, приведённые в [41], [42], [43].

## Задача 12

12-12.bas

Дан целочисленный массив A[N]. Проверьте, есть ли в нем элементы, равные нулю. Если есть, найдите номер первого из них, т.е. наименьшее I, при котором A[I]=0.

```
NEW
Ok
10 DEFINT A-Z:INPUT N:IF N<=0 THEN CLS:GOTO 10 ELSE DIM A(N)
20 FOR I=1 TO N:A(I)=INT(2*RND(-TIME)):PRINT A(I);:NEXT I:PRINT
30 FOR I=1 TO N:IF A(I)=0 THEN PRINT"Первый нулевой элемент:";I:END EL
SE NEXT I:PRINT "Нулевого элемента нет!":END
```

## Задача 13

12-13.bas

Дан целочисленный массив A[N]. Проверьте, есть ли в нем отрицательные элементы. Если есть, найдите наибольшее I, при котором A[I]<0.

```
NEW
Ok
10 INPUT N:DIM A(N):FOR I=1 TO N:A(I)=INT(2*RND(-TIME)-1):PRINT A(I);:NEXT I:PRINT
20 FOR I=1 TO N:IF A(I)<0 THEN B=I:NEXT I ELSE NEXT I
30 IF B=0 THEN PRINT"Отрицательного элемента в массиве нет" ELSE PRINT"Номер последнего отрицательного
элемента:";B
```

## Задача 14

12-14.bas

Проверьте, образуют ли элементы двумерного массива A[N,N] «магический квадрат» (это значит, что суммы чисел во всех её вертикалях, всех горизонталях и двух диагоналях одинаковы).

```
NEW
Ok
10 INPUT "Сторона квадрата";N:DIM A(N,N):FOR I=1 TO N:FOR J=1 TO N:PRINT
"Ввести";J;"элемент";I;"строки";:INPUT A(I,J):NEXT J:NEXT I
20 DIM S(N):FOR I=1 TO N:FOR J=1 TO N:S(I)=S(I)+A(J,I):NEXT J:NEXT I
30 FOR I=1 TO N-1:IF S(I)<>S(I+1) THEN PRINT "Элементы массива не образуют магический квадрат!":END
40 DIM C(N):FOR J=1 TO N:FOR I=1 TO N:C(J)=C(J)+A(J,I):NEXT I:NEXT J
50 FOR J=1 TO N-1:IF C(J)<>C(J+1) THEN PRINT "Элементы массива не образуют магический квадрат!":END
60 Q1=0:FOR I=1 TO N:FOR J=1 TO N:IF J=I THEN Q1=Q1+A(I,J):NEXT I ELSE NEXT J:NEXT I
70 IF Q1<>C(1) THEN PRINT "Не является!":END
80 Q2=0:FOR I=N TO 1 STEP -1:FOR J=N TO 1 STEP -1:IF J=I THEN Q2=Q2+A(I,J):NEXT I ELSE NEXT
J:NEXT I
90 IF Q1=Q2 THEN PRINT "Магический квадрат есть!":END
```

## Задача 14.1

12-141.bas

Постройте «магический квадрат» нечётного порядка.

```
NEW
Ok
80 INPUT N:DIM S(N,N)
90 FOR I=1 TO N:FOR J=1 TO N
```

```

100 B=J-I+(N-1)\2:C=2*J-I
110 IF B>=N THEN B=B-N ELSE IF B<0 THEN B=B+N
120 IF C>N THEN C=C-N ELSE IF C<=0 THEN C=C+N
130 S(I,J)=B*N+C:NEXT J,I
140 FOR I=1 TO N:FOR J=1 TO N
150 PRINTUSING"####";S(I,J);:NEXTJ:PRINT:NEXTI

```

## Задача 14.2

[12-142.bas](#)

Постройте «магический» квадрат чётного порядка.

```

NEW
Ok
10 DEFINT A-Z:GOTO 120
20 FOR I=P1 TO Q1:H1=NOT H1
30 IF H1><0 THEN X[I,A1]=NN-A1*N+1+N-1ELSEX[I,A1]=A1*N-N+I
40 NEXT:RETURN '→
50 FOR I=P2 TO Q2:H2=NOT H2
60 IF H2><0 THEN X[I,A2]=A2*N+1-I ELSE X[I,A2]=NN-A*N+I
70 NEXT:RETURN '→
80 FOR I=P3 TO Q3:H3=NOT H3
90 IFH3><0THENX[I,A3]=A3*N+1-I ELSE X[I,A3]=NN-A3*N+N-I+1
100 NEXT:RETURN '→
110 '***** Основная программа *****
120 INPUT N:IF N=2 THEN PRINT"Подобный квадрат Вы легко составите самостоятельно! Good luck!":END
ELSE DIM X[N,N]
130 N2=N\2:NN=N*N:P=(N=(N\4)*4):Q=P::R=-1
140 FOR A=1 TO N2-1
150 P2=1:Q2=A-1:A2=A:H2=R:GOSUB 50:P1=A:Q1=N2-1:A1=A:H1=-1:GOSUB 20
160 IF Q THEN X[N2,A]=NN-A*N+N2+1 ELSE X[N2,A]=NN-A*N+N2
170 P1=N2+1:Q1=N:A1=A:H1=NOT Q:GOSUB 20:Q=NOT Q:R=NOT R:NEXT
180 P1=1:Q1=N2-1:A1=N2:H1=NOT P:GOSUB 20:P1=N2+2:Q1=N:A1=N2:H1=0:GOSUB 20
190 P3=1:Q3=N2-1:A3=N2+1:H3=P:GOSUB 80:P3=N2+2:Q3=N:A3=N2+1:H3=-1:GOSUB 80
200 Q=P:R=-1
210 FOR A=N2+2 TO N
220 P2=1:Q2=N-A:A2=A:H2=Q:GOSUB 50
230 X[N-A+1,A]=A*N-A+1
240 P2=N-A+2:Q2=N2-1:A2=A:H2=-1:GOSUB 50
250 IF NOT R THEN::X[N2,A]=NN-A*N+N2:X[N2+1,A]=A*N-N2+1::ELSE::FOR B=N2 TO N2+1:X[B,A]=NN-
A*N+N-B+1:NEXT B
260 P2=N2+2:Q2=A-1:A2=A:H2=NOT R:GOSUB 50:P1=A:Q1=N:A1=A:H1=-1:GOSUB20
270 Q=NOT Q:R=NOT R:NEXT A
280 FOR A=N2 TO N2+1:FOR B=N2 TO N2+1
290 IF P<>0 THEN X[B,A]=A*N-N+B ELSE X[B,A]=NN-A*N+N-B+1
300 NEXT B,A
310 IF NOT P THEN::FOR A=N2 TO N2+1:X[N2-1,A]=A*N-N2+2:NEXT::FOR B=N2 TO
N2+1:X[B,N2+2]=N*N2-2*N+B:NEXT
320 FOR F1=1 TO N:FOR F2=1 TO N:PRINTUSING"####";X[F1,F2];:NEXT F2:PRINT :NEXT F1
330 END
run
? 8
1 16 41 40 32 17 56 57
63 10 23 26 39 47 50 2
3 54 19 38 30 43 11 62
61 52 45 28 36 21 12 5
60 13 44 29 37 20 53 4
6 51 22 35 27 46 14 59
58 15 42 31 34 18 55 7
8 49 24 33 25 48 9 64
Ok

```

## Задача 15

12-15.bas

Дан целочисленный массив A(N). Подсчитайте наибольшее число одинаковых идущих в нем подряд элементов.

```
NEW
Ok
10 Z=RND(-TIME):INPUT"Число элементов";N:DIM A(N)
20 FOR T=1 TO N:
    A(T)=INT(2*RND(1)):PRINT A(T);:
NEXT T:K=1:F=0:PRINT
30 FOR I=1 TO N-1
40     IF A(I)=A(I+1)
        THEN K=K+1:M=K
        ELSE M=K:K=1
50     IF M>F THEN F=M
60 NEXT I:PRINT"Максимальная длина:"F
up
Число элементов? 9
 0 1 0 1 1 0 0 1 0
Максимальная длина: 2
Ok
```

## Задача 16

12-16.bas

Подсчитайте количество различных чисел, встречающихся в целочисленном массиве A(N). Повторяющиеся числа учитывайте один раз.

```
NEW
Ok
10 DEFINT N,I,Y,L:INPUT N:DIM A(N):X=RND(-TIME)
20 FOR I=1 TO N:A(I)=INT(20*RND(1)-10):PRINTA(I);:NEXT?:IF N=1 THEN PRINT "Один-
одиношенек!":END
70 FOR Y=1 TO N-1:FOR L=Y+1 TO N
80 IF A(Y)<A(L) THEN 90 ELSE SWAP A(Y),A(L)
90 NEXT:NEXT
95 U=1
100 FOR I=1 TO N-1
110 IF A(I)<>A(I+1) THEN U=U+1:NEXT I ELSE NEXT I
120 PRINT "Количество различных элементов";U
```

## Задача 17

12-17.bas

Дан целочисленный массив A(N). Постройте массив B(N), который состоит из тех же элементов, что и массив A, но в котором все отрицательные элементы предшествуют всем неотрицательным.

```
NEW
Ok
30 INPUT"Число элементов";N
40 DIM A(N),B(N):Z=RND(-TIME)
50 INPUT"Чему равен максимальный элемент";F
60 PRINT "Это - данная таблица:"
70 FOR I=1 TO N:A(I)=INT(2*F*RND(1)-F):PRINT A(I);:NEXT:?
80 I=1:J=1:IF I>N THEN 120
90 FOR K=1 TO 0 STEP 1
```

```

100 IF A(I)<0 THEN B(J)=A(I):J=J+1
110 I=I+1:K=(I<=N):NEXT K
120 I=1:IF I>N THEN 170
130 FOR K=1 TO 0 STEP 1
140 IF A(I)>=0 THEN B(J)=A(I):J=J+1
150 I=I+1:K=(I<=N):NEXT K
160 PRINT "А это - искомая таблица:"
170 FOR J=1 TO N:PRINT B(J);:NEXT
    run
Число элементов? 10
Чему равен максимальный элемент? 9
Это - данная таблица:
6 -4  1  6  2  7  7 -9 -5 -5
А это - искомая таблица:
-4 -9 -5 -5  6  1  6  2  7  7
Ok

```

## Задача 18

[12-18.bas](#)

Даны целочисленные массивы A(N), B(M), причём

- $A(1) \leq A(2) \leq \dots \leq A(N)$ ,
- $B(1) \leq B(2) \leq \dots \leq B(M)$ .

Постройте целочисленный массив C(N+M), содержащий все элементы массивов A и B, в котором  $C(1) \leq C(2) \leq \dots \leq C(N+M)$ .

Эта важная задача принципиально должна быть выполнена за N+M действий. Возьмём из A и B по первому элементу. Меньший из них занесём в C и заменим следующим из его же массива. Снова выберем меньший из двух, занесём в C и т.д. После каждого сравнения в C добавляется элемент — значит, сравнений будет меньше, чем N+M. Нужно только позаботиться о том, чтобы программа работала верно и при исчерпании одного из массивов.

```

NEW
Ok
10 INPUT "Введите число элементов в массиве A, затем - в массиве B";N,M:
FF=RND(-TIME)
20 DIM A(N),B(M),C(N+M)
30 FOR K=1 TO N:A(K)=INT(7*RND(1)):NEXT
40 FOR L=1 TO M:B(L)=INT(9*RND(1)):NEXT
41 FOR I=1 TO N-1:FOR J=I+1 TO N:IF A(I)>A(J) THEN SWAP A(I),A(J)
42 NEXT J,I
43 FOR I=1 TO N:PRINT A(I);:NEXT:PRINT
45 FOR I=1 TO M-1:FOR J=I+1 TO M:IF B(I)>B(J) THEN SWAP B(I),B(J)
46 NEXT J,I
47 FOR I=1 TO M:PRINT B(I);:NEXT:PRINT
50 P=0:Q=0
60 FOR I=1 TO N+M
70 IF P=N THEN Q=Q+1:C(I)=B(Q):NEXT:GOTO 110
80 IF Q=M THEN P=P+1:C(I)=A(P):NEXT:GOTO 110
90 IF A(P+1)<B(Q+1) THEN P=P+1:C(I)=A(P) ELSE Q=Q+1:C(I)=B(Q)
100 NEXT
110 FOR I=1 TO N+M:PRINT C(I);:NEXT

```

## Задача 19

[12-19.bas](#)

Дан целочисленный массив A(N,M). Найдите количество тех чисел I от 1 до N, для которых  $A(I,J)=0$  при некотором J от 1 до 50. Определить количество нулевых элементов в каждой строке.

```

NEW
Ok
10 INPUT "Введите значения N и M";N,M:IF M>50 THEN 10
20 DIM A(N,M):G=RND(-TIME)
30 FOR I=1 TO N:FOR J=1 TO M:A(I,J)=INT(10*RND(1)-5):PRINT A(I,J);:NEXT J:PRINT:NEXT I
40 DIM P(N):FOR I=1 TO N:FOR J=1 TO M
50 IF A(I,J)=0 THEN C=C+1:NEXT J ELSE NEXT J
60 P(I)=C:C=0:NEXT I
70 PRINT "Количество нулевых элементов по строкам: ":FOR K=1 TO N:? P(K);:NEXT
80 FOR K=1 TO N:IF P(K)<>0 THEN S=S+1:NEXT K ELSE NEXT K
90 PRINT:PRINT "Количество строк с нулевыми элементами: ";S:END

```

## Задача 20

[12-20.bas](#)

Дан целочисленный массив L(N,M). Найдите наименьшее число K, обладающее таким свойством: хотя бы в одной строке массива все элементы не превосходят K.

```

NEW
Ok
20 INPUT "Введите значения N и M";N,M:IF N=1 OR M=1 THEN 20
30 DIM L(N,M):G=RND(-TIME)
40 FOR I=1 TO N:FOR J=1 TO M:L(I,J)=INT(10*RND(1)):PRINT L(I,J);:NEXT J:PRINT:NEXT I
50 FOR I=1 TO N
60   GOSUB 130
70 NEXT I
80 FOR I=1 TO N-1:FOR C=I+1 TO N
90 IF L(I,1)<L(C,1) THEN NEXT C:B=L(I,1) ELSE NEXT I:B=L(C,1)
100 PRINT
110 FOR I=1 TO N
120 NEXT:PRINT:PRINT "K=";B:END
130 'Упорядочение элементов в строке
140 FOR J=1 TO M-1:FOR K=J+1 TO M
150 IF L(I,J)>L(I,K) THEN GOTO 160 ELSE SWAP L(I,J),L(I,K)
160 NEXT K:NEXT J:PRINT
170 FOR J=1 TO M:PRINT L(I,J);:NEXT J
180 RETURN

```

## Задача 21

Аналогичный случай был...

—Бравый солдат Швейк

[12-21.bas](#)

Дан двумерный целочисленный массив A(M,N). Найти наибольшее целое число K, обладающее таким свойством: в любой строке таблицы есть элемент, больший или равный K.

```

NEW
Ok
7 CLS
10 INPUT "Введите значения M и N";M,N
15 IF M<=1 OR N<=1 THEN 7
20 DIM A(M,N):X=RND(-TIME)
30 FOR I=1 TO M:FOR J=1 TO N:A(I,J)=INT(21*RND(1)):PRINT A(I,J);:NEXT J:PRINT:NEXT I
40 DIM B(M)
50 FOR I=1 TO M:FOR J=1 TO N-1:FOR L=J+1 TO N

```

```

60 IF A(I,J)>A(I,L) THEN NEXT L:B(I)=A(I,J) ELSE NEXT J:B(I)=A(I,L)
70 NEXT I
80 FOR I=1 TO M-1:FOR J=I+1 TO M
90 IF B(I)<=B(J) THEN NEXT J:PRINT"Число,удовлетворяющее условию:"B(I) ELSE NEXT
I:PRINT"Число,удовлетворяющее условию:"B(J)

```

По сути дела условия задач 20 и 21 одинаковы, если, разумеется, массивы A и L совпадают!

## Задача 22

[12-22.bas](#)

Постройте алгоритм разложения натуральных чисел на простые множители. В какой форме будут представлены результаты работы этого алгоритма.

```

NEW
Ok
560 INPUT"Натуральное число";S:N=S:W$=""
570 I=2
580 IF N/I=FIX(N/I) THEN N=N/I:W$=W$+STR$(I) ELSE I=I+1:GOTO 580
590 IF I<N THEN 570 ELSE W$=W$+STR$(N)
600 FOR L=2 TO LEN(W$)
610 IF MID$(W$,L,1)=CHR$(32) THEN MID$(W$,L,1)="*"
620 NEXT:PRINT S;"=";W$
630 END

```

Совершенные числа красивы. Но известно, что красивые вещи редки и немногочисленны, безобразные же встречаются в изобилии.

—Никомах из Герасы (ок. 100 н.э.)

## Задача 23

[12-23.bas](#)

Натуральное число называют *совершенным*, если оно равно сумме всех своих делителей, не считая его самого (например,  $6=1+2+3$  — совершенное число). Напишите алгоритм, проверяющий, является ли заданное число совершенным.

```

NEW
Ok
100 INPUT"Введите натуральное N:";N:IF N=1 THEN 140
120 FOR J=1 TO N-1
130 IF N/J=FIX(N/J) THEN K=K+J:NEXT J:ELSE NEXT J
140 IF K=N THEN PRINT N"-совершенное число" ELSE PRINT N"не является совершенным"

```

## Задача 23\*

[12-230.bas](#)

Написать программу, находящую все совершенные числа, принадлежащие отрезку  $[1, N]$ .

```

NEW
Ok
10 INPUT"Введите натуральное N:";N
15 PRINT"Совершенные числа на промежутке от 1 до"N": "
20 FOR I=2 TO N:FOR J=1 TO I-1

```

```

30 IF I/J=FIX(I/J) THEN K=K+J:NEXTJ:ELSE NEXTJ
40 IF K=I THEN PRINT I:K=0:L=1:NEXT I ELSE K=0:NEXTI
50 IF L<1 THEN PRINT "Таких чисел нет"
run
Введите натуральное N? 1000
Совершенные числа на промежутке от 1 до 1000:
6
28
496
Ok

```

## Задача 24

[12-24.bas](#)

Напечатайте в порядке возрастания первые 1000 чисел, которые не имеют простых делителей, кроме 2, 3 и 5. (Начало списка: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, ...).

```

NEW
Ok
20 CLEAR 1000:I=2:N=1:PRINT 1;SPC(2);"1=1"
30 IF NOT I<=1000 THEN 90
40 FOR K=1 TO 0 STEP 1:GOSUB 100
50 FOR T=1 TO LEN(V$) STEP 2
60 IF (MID$(V$,T,2)="1*" OR MID$(V$,T,2)="2*") OR (MID$(V$,T,2)="3*" OR MID$(V$,T,2)="5*") THEN
H$="good" ELSE H$="not":T=LEN(V$)+1
70 NEXT T:IF H$="good" THEN PRINT I;SPC(2);MID$(STR$(N),2);"=";LEFT$(V$,LEN(V$)-1):I=I+1
80 N=N+1:K=(I<=1000):NEXT K
90 PRINT :PRINT "Выбор закончен":END
100 V$="":P=2:W=N
110 D=W/P:IF D=INT(D) THEN V$=V$+MID$(STR$(P),2)+"*":W=D:GOTO 110
120 P=P+1:IF W>=P THEN 110
130 RETURN
run
1 1=1
2 2=2
...
85 972=2*2*3*3*3*3*3
86 1000=2*2*2*5*5*5
Ok

```

## Задача 25

[12-25.bas](#)

Придумайте способ нахождения самой лёгкой и самой тяжёлой из 100 монет различной массы, если можно сделать не более 150 взвешиваний на чашечных весах без гирь. (На каждую чашку весов помещается по одной монете; весы показывают, какая из них тяжелее.)

```

NEW
Ok
5 'Автор программы: Поляков Сергей (9 класс), 02.01.90
10 WIDTH80:CLS:INPUT "Введите число монет(продолжение сообщений по нажатию клавиши 'Пробел') ";N:IF
N/2=FIX(N/2) THEN DIM A(N):X=RND(-TIME) ELSE CLS:RUN 10
20 PRINT "В Вашем кошельке хранятся следующие монеты:"
30 PRINT "_____ "
40 FOR I=1 TO N:A(I)=INT(100*RND(1)+1):PRINT A(I);:NEXT:R1=A(1)
50 PRINT:PRINT "_____ ":H$=INPUT$(1)
60 FOR I=1 TO N-1 STEP 2
70 IF A(I)<A(I+1) THEN R=A(I):D=A(I+1) ELSE R=A(I+1):D=A(I)
80 G=G+1:PRINT " И так, "G" - ОЕ ВЗВЕШИВАНИЕ":PRINT

```



```

90      PRINT " Вес  "I"-ой монеты  - "A(I)" граммов      "I+1"-ой монеты-      "A(I+1)" граммов
":PRINT
100     PRINT " Самая меньшая по весу из предыдущих монет - "R1" граммов":PRINT
110     G=G+1:IF R<R1 THEN R1=R
120     PRINT " Меньшая из 2 только что взятых монет - "R"граммов; после сравнения этих монет ОСТАЕТСЯ
монета весом "R1"граммов":PRINT
125 PRINT "      ("G"-ОЕ ВЗВЕШИВАНИЕ)":PRINT
130     PRINT " Самая большая по весу из предыдущих монет - "D1" граммов":PRINT
140     G=G+1:IF D>D1 THEN D1=D
150     PRINT " Большая из 2 только что взятых монет - "D"граммов; после сравнения этих монет ОСТАЕТСЯ монета
весом "D1" граммов":PRINT
155 PRINT "      ("G"-ОЕ ВЗВЕШИВАНИЕ)":PRINT
160     H$=INPUT$(1):GOSUB 200
170 C=C+3:NEXT I
180     PRINT:PRINT"После долгих стараний: ("C" взвешиваний) мы нашли,что наибольший вес: ":PRINT
D1"граммов":PRINT
190     PRINT " Минимальный вес: "R1"граммов":END
200 LOCATE 0,4:FOR F=1 TO 17:FOR J=1 TO 80:PRINT " ";:NEXT:NEXT:LOCATE 0,4:RETURN

```

## Задача 26

[12-26.bas](#)

Имеется 1000 монет, из которых одна фальшивая (она легче других). Докажите, что нельзя придумать способ, гарантирующий нахождение фальшивой монеты за 6 взвешиваний на чашечных весах без гирь. Придумайте способ, гарантирующий нахождение фальшивой монеты за 7 взвешиваний.

```

NEW
Ok
20 B=1000:D=1:Q=1:PRINT "Мы имеем";B;"монет":GOTO 110
30 I=I+1:PRINT I;"ВЗВЕШИВАНИЕ."
40 A=INT(B/3):PRINT "Разделим все оставшиеся монеты на 4 кучки по";A;" ";A;" ";A;" ";A;"и";B-3*A-Q+D;"монет."
50 PRINT "Взвесим 2 кучки по";A;"монет."
60 PRINT "Если одна из них легче, то берем эту кучку монет и еще";B-3*A-Q+D;"монеты."
70 PRINT "Остальные монеты не являются фальшивыми."
80 PRINT "Для получения дальнейшей информации нажмите любую клавишу."
90 IF NOT B/3=A THEN Q=Q-1
100 B=A:P$=INPUT$(1):RETURN
110 FOR T=1 TO 6:GOSUB 30:NEXT
120 B=3:D=0:Q=0:GOSUB 30
130 PRINT "Итак, Вы нашли фальшивую монету."

```

## Задача 27

[12-27.bas](#)

Среди  $2 \cdot N + 1$  различных по массе монет нужно найти среднюю (т.е. такую, которая тяжелее  $N$  монет и легче  $N$  других монет). Придумайте способ, позволяющий сделать это не более чем за  $100 \cdot N$  взвешиваний на чашечных весах без гирь.

```

5 'Автор программы: Шарковский Игорь (9 класс), 02.01.90
10 KEY OFF:CLS:COLOR 15,4
20 INPUT"Количество монет (нечетное число, не большее 13)";N
30 IF N/2=FIX(N/2) OR N>13 THEN CLS :RUN20 ELSE X=RND(-TIME):DIM A(N)
40 PRINT "В вашем кошельке лежат монеты:"
50 FOR I=1 TO N:PRINT "┌───┐";:NEXT:PRINT:
60 FOR I=1 TO N
70 A(I)=INT(20*(RND(1))+10):PRINT "┌─"A(I)─┐";
80 NEXT :PRINT:FOR I=1 TO N:PRINT "┌───┐";:NEXT:FOR I=1 TO 1000:NEXT:PRINT:
90 PRINT "Приступим к упорядочению монет по возрастанию ":FOR I=1 TO 1000:NEXT
100 PRINT :IF N=1 THEN PRINT "Всего-то?":END

```

```

110 FOR I=1 TO N-1:FOR J=I+1 TO N
120 IF A(I)>=A(J) THEN SWAP A(I),A(J):K=K+1:NEXTJ:NEXT I ELSE K=K+1:NEXT J:NEXT I
130 PRINT "После упорядочения монеты в Вашем кошельке расположились так:"
140 FOR I=1 TO N:PRINT"┌───┐";NEXT:FOR I=1 TO 1000:NEXT:PRINT
150 FOR I=1 TO N :PRINT "│"A(I)"│";NEXT
160 PRINT:FOR I=1 TO N:PRINT"└───┘";:NEXT:FOR I=1 TO 1000:NEXT:FOR I=1 TO 4:PRINT:NEXT
170 PRINT "Теперь (наконец-то) приступим к нахождению средней из них (по массе)":FOR I=1 TO 1000:NEXT:PRINT
190 PRINT "Средняя по массе монета в Вашем кошельке - "A(FIX(N/2)+1)"г":PRINT"Число взвешиваний - "K
:FOR I=1 TO 1000:NEXT
220 LOCATE 4+FIX(N/2-1)*6,11:PRINT "  ████ "
230 LOCATE 4+FIX(N/2-1)*6,12:PRINT "  "
235 LOCATE 4+FIX(N/2-1)*6,13:PRINT "  BOT ОНА"
240 LOCATE 0,24

```

## Задача 28

[12-28.bas](#)

Последовательность  $A(1), A(2), A(3), \dots$  определяется так  $A(1)=1$ ,  $A(2N)=A(N)$ ,  $A(2N+1)=A(N)+A(N+1)$ . Напишите программу, вычисляющую  $A(N)$  по известному  $N$ .

```

NEW
Ok
490 INPUT"Введите N";N:DIM A(N)
500 A(1)=1:PRINT A(1);:FOR I=2 TO N
510 IF I/2=INT(I/2) THEN A(I)=A(I/2)ELSE A(I)=A(INT(I/2))+A(INT(I/2)+1)
520 PRINT A(I);:NEXT:PRINT:PRINT "An=";A(N)
530 END

```

Каждая истинная работа имеет свою красоту.

—Н.К.Перих

## Задача 29

[12-29.bas](#)

Дан целочисленный массив  $A(N)$ . Найдите наименьшее число  $K$  элементов, которые нужно «выкинуть» из последовательности  $A(1), A(2), \dots, A(N)$ , чтобы осталась возрастающая последовательность.

```

10 DEFINT Q,K,I,J,N,D:INPUT "Укажите количество элементов";N:DIM A(N),Q(N),B(N):X=RND(-TIME)
15 PRINT"Исходная последовательность:"
20 FOR I=1 TO N:A(I)=INT(9*RND(1)):PRINT A(I);:NEXTI
30 FOR I=1 TO N:Q(I)=N-1:NEXT I
40 FOR I=1 TO N-1:FOR J=I+1 TO N
50 IF A(I)<A(J) THEN IF Q(I)<Q(J)+1 THEN Q(J)=Q(I)-1
55 NEXT J,I
60 K=Q(1):FOR I=2 TO N
80 IF K>Q(I) THEN K=Q(I)
90 NEXT I:D=K
100 PRINT:PRINT"Нужно выбросить: ";K;"элемента"
120 FOR I=N TO 1 STEP(-1)
130 IF Q(I)=K THEN B(K)=A(I):K=K+1:NEXT I ELSE NEXTI
135 PRINT"Одна из самых длинных возрастающих подпоследовательностей:"
140 FOR J=N-1 TO D STEP (-1):PRINT B(J);:NEXTJ
гип
Укажите количество элементов? 10
Исходная последовательность:

```

```
·7··4··2··4··7··8··2··6··8··3
```

Нужно выбросить: 6 элемента

Одна из самых длинных возрастающих подпоследовательностей:

```
·2··4··6··8
```

Ok

В этой задаче программными строками 10–20 описываем массив A и заполняем его псевдослучайными числами; это делается для формирования тестовых примеров без утомительного набора их на клавиатуре. Строки с 30 по 100 позволяют с помощью дополнительного массива Q найти наименьшее число K элементов, которые нужно «выкинуть» из массива, чтобы осталась возрастающая подпоследовательность. И, наконец, в строках 120–140 происходит выделение этой максимальной по длине подпоследовательности.

## Задача 30

[12-30.bas](#)

Дано три целочисленных массива A(N), B(N), C(N). Известно, что существуют целые числа, встречающиеся во всех трёх таблицах. Найдите одно из таких чисел.

```
NEW
Ok
20 INPUT "Введите размер массивов"; N
30 Z=RND(-TIME)
40 PRINT "Первый массив"
50 DIM S(N)
60 FOR I=1 TO N: X=INT(N*RND(1)): S(I)=X: PRINT S(I); : NEXT: PRINT
90 PRINT "Второй массив"
100 DIM K(N)
110 FOR I=1 TO N: X=INT(N*RND(1)): K(I)=X: PRINT K(I); : NEXT: PRINT
130 PRINT "Третий массив"
140 DIM P(N): FOR I=1 TO N: X=INT(N*RND(1)): P(I)=X: PRINT P(I);
160 NEXT: PRINT
170 DIM U(N,3): Y$="а его нет!"
180 FOR I=1 TO N: M=1: U(I,M)=S(I): NEXT
210 FOR I=1 TO N: M=2: U(I,M)=K(I): NEXT
240 FOR I=1 TO N: M=3: U(I,M)=P(I): NEXT
270 FOR I=1 TO 3: FOR J=1 TO N: FOR R=1 TO 3: FOR K=1 TO N
310 IF U(J,I)=U(K,R) THEN NEXT R: Y$=STR$(U(J,I)) ELSE NEXT K,J,I
320 PRINT
330 PRINT "Общий элемент: "; Y$: END
```

## Задача 30\*

[12-300.bas](#)

Найдём теперь все числа, обладающие свойством, описанным в задаче 30.

```
NEW
Ok
10 INPUT "Сколько чисел в каждой таблице"; N: DIM A1(N), A2(N), A3(N)
20 INPUT "Чему равен максимальный элемент"; F
30 FOR I=1 TO N: A1(I)=INT(2*F*RND(1)-F): PRINT A1(I); : NEXT: PRINT: PRINT
40 FOR I=1 TO N: A2(I)=INT(2*F*RND(1)-F): PRINT A2(I); : NEXT: PRINT: PRINT
50 FOR I=1 TO N: A3(I)=INT(2*F*RND(1)-F): PRINT A3(I); : NEXT: PRINT: PRINT
60 PRINT "А вот те числа, которые встречаются в трех таблицах сразу:"
70 FOR I=1 TO N: FOR J=1 TO N: FOR K=1 TO N
80 IF A1(I)=A2(J) AND A1(I)=A3(K) THEN PRINT A1(I); : NEXT I ELSE NEXT K,J,I
90 PRINT : PRINT "Увы, но все числа, встречающиеся в трех таблицах сразу, уже выведены на экран."
```

## Задача 31

12-31.bas

Даны два слова X\$ и Y\$. Проверить, можно ли из символов, входящих в слово X\$, составить слово Y\$. Символы можно переставлять, но каждый символ можно использовать не более одного раза!

```
NEW
Ok
10 INPUT X$,Y$
15 IF X$="" OR Y$="" THEN PRINT "Нет":GOTO 10
20 FOR I=1 TO LEN(Y$):FOR J=1 TO LEN(X$)
30 IF MID$(Y$,I,1)=MID$(X$,J,1) THEN X$=LEFT$(X$,J-1)+RIGHT$(X$,LEN(X$)-J):A$="Да":NEXT I ELSE
A$="Нет":NEXT J
50 PRINT A$:END
run                A затем...    Ok                Подумайте,почему
? молоко,комоло    print LEN(x$)    x$="" ?
Да                0
Ok                Ok
```

## Задача 32

12-32.bas

Составьте алгоритм, находящий по целым величинам A, B и C такие целые числа X и Y, что  $A \cdot X + B \cdot Y = C$  (если такие X и Y есть).

```
NEW
Ok
30 PRINT "Ax+By=C (Диофантовы уравнения)"
40 INPUT "Чему равны A,B,C";A,B,C
50 A1=A:B1=B:C1=C
60 'Нам нужно найти число M такое, которое в 5 раз больше числа цифр в максимальном по модулю из чисел A, B, C (строки 60-80)
70 CJ=ABS(A):IF ABS(B)>CJ THEN CJ=ABS(B)
80 IF ABS(C)>CJ THEN CJ=ABS(C)
90 CJ$=MID$(STR$(CJ),2):M=5*LEN(CJ$)
100 DIM Q(M)
110 N=0:I=0:D=ABS(A):S=ABS(A):R=ABS(B)
120 I=I+1:IF NOT R<>0 THEN 160
130 FOR W=1 TO 0 STEP 1 '**** WHILE ****
140 N=I:Q[I]=FIX(S/R):D=R-R*Q[I]:S=D
150 I=I+1:W=R<>0:NEXT W '**** WEND ****
160 IF D=0 THEN:::IF C=0 THEN PRINT "Неопределенность":END ELSE PRINT "Нет решения":END:::ELSE
170 IF FIX(C/D)*D<>C THEN PRINT "Нет решения":END
180 A=A/D:B=B/D:C=C/D 'Уравнение приведем к виду, когда A и B взаимно простые.
190 IF N=0 THEN V=0:U=1 ELSE V=1:U=0:::FOR I=N-1 TO 1 STEP -1:S=V:V=U-V*Q[I]:U=S:NEXT I
200 X=C*U*SGN(A):Y=C*V*SGN(B):C=FIX(X/B):X=X-C*B:Y=Y+C*A 'Частные решения
210 PRINT "Получено частное решение":PRINT "уравнения: (";STR$(A1);")*x+(";STR$(B1);")*y=";C1
220 PRINT "X0=";STR$(X);SPC(2);"Y0=";STR$(Y)
230 PRINT "Общее решение:":PRINT "Xi=";STR$(X);"+i*(";STR$(B1);") и Yi=";STR$(Y);"-i*(";STR$(A1);"),"
240 PRINT "где i- целое число."
250 ' Таблица тестовых примеров
260 '
270 '
280 '
290 '
300 '
310 '
320 '
330 '


| A    | B    | C     | X                | Y    |
|------|------|-------|------------------|------|
| 1000 | 23   | 1046  | -22              | 1002 |
| 0    | 0    | 0     | Неопределенность |      |
| 57   | -103 | 47009 | 73               | -416 |
| 10   | 12   | 578   | -1               | 49   |
| 10   | 12   | 97    | Нет решения      |      |


```

### Задача 33

12-33.bas

А вы, друзья, как не садитесь,  
Всё в музыканты не годитесь.

—И.А.Крылов

Перестановкой  $N$  чисел называется последовательность  $A(1), A(2), \dots, A(N)$ , в которой встречаются по одному разу все числа от 1 до  $N$ . (Например, перестановками трёх чисел являются: (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1) и других перестановок трёх чисел нет.) Постройте алгоритм, печатающий все перестановки  $N$  чисел.

Этой проблеме посвящено много публикаций. Она имеет давнюю историю, её возникновение можно отнести к началу XVII века, когда в Англии зародилось особое искусство колокольного боя, основанного, если говорить упрощённо, на выбивании на  $N$  различных колоколах всех  $n!$  перестановок. Перестановки эти следовало выбивать «по памяти», что способствовало разработке сторонниками этого искусства первых простых методов систематического перечисления всех перестановок (без повторений). Некоторые из этих незаслуженно забытых методов были переоткрыты в настоящее время в связи с появлением цифровых машин. Это искусство просуществовало до наших дней, поскольку знаменитая «Книга рекордов Гиннеса» содержит упоминание о выбивании всех  $8! = 40320$  перестановок на 8 колоколах в 1963 году; установление этого рекорда потребовало 17 часов 58.5 минут! Конечно, использование цифровых машин позволяет генерировать перестановки значительно быстрее, однако разница не так уж велика, как можно было бы подумать — за 18 часов даже самый быстрый компьютер не сможет получить все перестановки  $n$ -элементного множества, если  $n > 13$  (с другой стороны, перестановки 8-элементного множества можно получить в течении доли секунды). Это простое следствие того факта, что  $n!$  растёт весьма быстро с ростом  $n$ .

Укажем алгоритм получения всех перестановок из  $N$  элементов. Пусть имеется множество элементов

$$A = \{A_1, A_2, \dots, A_N\}.$$

Обозначим две разные перестановки этих элементов

$$A_i = [A_1^i, A_2^i, \dots, A_N^i], \quad A_j = [A_1^j, A_2^j, \dots, A_N^j].$$

**Определение.** Две перестановки называются *упорядоченными в лексикографическом смысле*,  $A_i > A_j$ , если:

$$1. \quad A_1^i > A_1^j \quad \text{или}$$

$$2. \quad \text{существует целое } k \text{ из } [2, N] \text{ такое, что } A_l^i = A_l^j \text{ для всех } l \text{ из } [1, k-1] \text{ и } A_k^i > A_k^j.$$

**Пример 1.** Пусть  $A = \{1, 2, 3, 4\}$ ;  $A_i = [4, 3, 2, 1]$ ;  $A_j = [4, 2, 1, 3]$ .

Перестановки  $A_i$  и  $A_j$  упорядочены лексикографически,  $A_i > A_j$ , так как выполняются условия  $A_1^i = A_1^j$  и  $A_2^i > A_2^j$ .

**Пример 2.** Лексикографически упорядочены фамилии владельцев телефонов в телефонном справочнике.

Таким образом, если мы научимся генерировать перестановки в лексикографическом порядке, то сможем получать все возможные варианты — начиная с «самой маленькой» и кончая «самой большой». Пусть исходное множество, для которого нужно получить все перестановки, есть множество натуральных чисел  $1, 2, \dots, N$ . Положим, что исходная перестановка  $1, 2, \dots, N$ . Текущую перестановку обозначим  $A = [A_1, A_2, \dots, A_N]$ .

Предлагается следующая последовательность действий (можно доказать, что она всегда приводит к получению лексикографически упорядоченных перестановок).

1. Просматриваем  $A$  справа налево в поисках самого правого  $i$ , для которого  $A_i < A_{i+1}$ . Другими словами, просматриваем перестановку с «хвоста», отыскиваем первый элемент, меньший своего соседа справа, и запоминаем его номер  $i$ . Если такого элемента не окажется, т.е.  $i$  будет равно 0, значит, эта перестановка последняя и следует закончить выполнение алгоритма. Заметим, что все элементы, стоящие справа от  $A_i$ , упорядочены по убыванию.
2. Просматриваем  $A$  слева направо, начиная с  $A_{i+1}$ , в поисках элементов  $A_j$  таких, что  $i < j$  и  $A_i < A_j$ . Номер (обозначим его  $j$ ) наименьшего среди этих элементов запоминаем. Иначе говоря, начиная с номера  $i+1$  выделяем все элементы, большие найденного  $A_i$ , и среди них отыскиваем наименьший.
3. Меняем местами элементы  $A_i$  и  $A_j$ .
4. Переворачиваем «хвост» перестановки от  $A_{i+1}$  до  $A_N$ , т.е. элемент  $A_{i+1}$  меняем местами с  $A_N$ ,  $A_{i+2}$  — с  $A_{N-1}$  и т.д.

```
Ok
5 DEFINT N,K,J,I:INPUT N:DIM A(N)
10 FOR L=1TO N:A(L)=L:NEXT L
20 I=N
30 GOTO 110
40 FOR K=1 TO N:PRINTA(K);:NEXT K:PRINT:J=N-1
50 IF A(J)>A(J+1)ANDJ>0 THEN J=J-1:GOTO 50
60 I=J
70 IF I>0 THEN J=N:GOTO 80 ELSE GOTO 110
80 IF A(I)>A(J) THEN J=J-1:GOTO 80
90 SWAP A(I),A(J):P1=I+1:P2=N
100 IF P1<P2 THEN SWAP A(P1),A(P2):P1=P1+1:P2=P2-1:GOTO100
110 IF I>0 THEN GOTO 40
run
? 3
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1
Ok
```

Очевидно, программа остаётся работоспособной и тогда, когда элементами исходного множества являются любые числа (не только натуральные), между которыми установлено соотношение порядка.

Кстати, какие изменения нужно внести в программу, чтобы она строила все перестановки множества букв латинского алфавита? Может ли она проделать те же преобразования с буквами русского алфавита?

**Задача.** Написать все предложения, которые можно составить из слов «Ваши прекрасные глаза», «прекрасная маркиза», «от любви», «сулят», «мне», «смерть» путём их всевозможных перестановок (данная ситуация обыгрывается в пьесе Мольера «Мещанин во дворянстве»).

## Задача 33.1

[12-331.bas](#)

Перестановки с повторениями.

```
NEW
Ok
5 DEFINT N,K,J,I:INPUT"Число элементов";N:DIM A(N)
10 W=0:FOR L=1 TO N:INPUT A(L):NEXT L
12 FOR L1=1 TO N-1:FOR L2=L1 TO N
14 IF A(L1)>A(L2) THEN SWAP A(L1),A(L2)
16 NEXT:NEXT
20 I=N
30 GOTO 110
40 PRINT W;SPC(5);:FOR K=1 TO N:PRINT A(K);:NEXT K:PRINT:J=N-1
50 IF A(J)>=A(J+1)AND J>0 THEN J=J-1:GOTO 50
60 I=J
70 IF I>0 THEN J=N:GOTO80 ELSE GOTO 110
80 IF A(I)>=A(J) THEN J=J-1:GOTO 80
90 SWAP A(I),A(J):P1=I+1:P2=N
100 IF P1<P2 THEN SWAP A(P1),A(P2):P1=P1+1:P2=P2-1:GOTO 100
110 IF I>0 THEN W=W+1:GOTO 40
гип
Число элементов? 4
? 1
? 2
? 2
? 2
·1·····1··2··2··2
·2·····2··1··2··2
·3·····2··2··1··2
·4·····2··2··2··1
Ok
```

## Задача 34

[12-34.bas](#)

Дан целочисленный массив T(N). Найдите наименьшее натуральное число X, не обладающее следующим свойством: «можно так вычеркнуть некоторые числа таблицы T, чтобы сумма оставшихся равнялась X.

```
NEW
Ok
10 INPUT"Введите количество элементов в массиве(больше чем 1):";N:DIM T(N)
20 PRINT:PRINT" Натуральные элементы массива перед Вами: ":PRINT"_____ "
30 FOR I=1 TO N:T(I)=INT(3*RND(-TIME))+1:PRINT T(I);:NEXT:PRINT
40 FOR I=1 TO N-1:FOR J=I+1 TO N
50 IF T(I)>=T(J) THEN SWAP T(I),T(J)
60 NEXT J,I:PRINT"_____ "
70 PRINT"Искомое число:";
80 IF T(1)=1 THEN GOSUB 100:PRINT X:END ELSE X=1:PRINT X:END
90 ' *****
100 S=T(1):A$="Продолжить":K=2
110 IF NOT(K<=N AND A$="Продолжить") THEN X=S+1:PRINT X:END
120 FOR I=1 TO 0 STEP 1
130 IF T(K)<=S+1 THEN S=S+T(K):K=K+1 ELSE A$="Закончить"
140 I=(K<=N AND A$="Продолжить"):NEXT
150 X=S+1:RETURN
```

## Задача 35

12-35.bas

Составьте алгоритм подсчёта числа способов, которыми можно уплатить K рублей купюрами в 1, 3, 5, 10, 25, 50, 100 рублей.

```
NEW
Ok
10 INPUT N
20 PRINT"Могут уплатить так:"
30 PRINT"100 50 25 10 5 3 1"
40 PRINT"_____ "
50 FOR A=0 TO N\100:FOR B=0 TO N\50:FOR C=0 TO N\25:FOR D=0 TO N\10:FOR F=0 TO N\5:FOR I=0 TO N\3:FOR J=0 TO N
60 IF 100*A+50*B+25*C+10*D+5*F+3*I+J=N THEN L=L+1:PRINT A;B;C;D;F;I;J
70 NEXT J,I,F,D,C,B,A
80 PRINT"_____ "
90 PRINT"Число способов: ";L
```

## Задача 36

12-36.bas

В стране имеется N городов, соединённых авиационным сообщением. Стоимость билета из I-го города в J-й записана в массиве A[N,N]. Составьте алгоритм, находящий стоимость перелёта из I-го города в J-й по самому дешёвому маршруту (возможно с пересадками).

```
NEW
Ok
10 CLS:INPUT"Число городов";N
20 DIM CC[N,N],ST[N,N],G[N]
30 FOR I=1 TO N:FOR J=1 TO N
40 PRINT"Из" I " -го города в" J " -ый: ";:INPUT CC[I,J]
50 NEXT J,I
60 PRINT"Это - данная таблица:"
70 FOR I=1 TO N:FOR J=1 TO N:PRINTCC[I,J];TAB(4*J);:NEXTJ:PRINT:NEXTI
80 PRINT"Таблица минимальных цен:"
90 FOR I=1 TO N
100 FOR J=1 TO N:G[J]=J:NEXT J
110 G[1]=I:G[I]=1:S=1
120 FOR J=1 TO N:ST[I,J]=CC[I,J]:NEXT J
130 IF S>=N THEN 210 ELSE FOR R=1 TO 0
140 MIN=ST[I,G[S+1]]:JM=S+1
150 IF S+2>N THEN 170 ELSE FOR J=S+2 TO N
160 IF ST[I,G[J]]<MIN THEN MIN=ST[I,G[J]]:JM=J:NEXT J ELSE NEXT J
170 SWAP G[S+1],G[JM]:S=S+1
180 IF S+1>N THEN 200 ELSE FOR J=S+1 TO N
190 IF ST[I,G[S]]+CC[G[S],G[J]]<ST[I,G[J]] THEN ST[I,G[J]]=ST[I,G[S]]+CC[G[S],G[J]]:NEXT J ELSE
NEXT J
200 R=(S<N):NEXT R
210 NEXT I
220 FOR I=1 TO N:FOR J=1 TO N 'Вывод таблицы цен
230 PRINT ST[I,J];TAB(4*J);:NEXT J:PRINT:NEXT I
```

## Задача 37

12-37.bas



Вся шахматная партия — это один замаскированный ход конём.

—С.Тартаковер

Трава не растёт там, где ступил мой конь.

—Аттила — вождь гуннов

Существует способ обойти шахматным конём шахматную доску, побывав на каждом поле по одному разу. Составьте алгоритм отыскания такого способа.

```
NEW
Ok
10 COLOR 1,15,8:SCREEN 2,2:OPEN"grp:"FOR OUTPUT AS#1
20 DEFINT A-Z:K=8:E=8:X8=55:Y8=25
30 '***** Рисунок шахматной доски *****
40 FOR R0=Y8 TO Y8+64 STEP 64:FOR I=X8 TO X8+32*3 STEP 32:LINE(I+1,R0)-
STEP(15,15),,BF:LINE(I+17,R0+16)-STEP(15,15),,BF:LINE(I+1,R0+32)-
STEP(15,15),,BF:LINE(I+17,R0+48)-STEP(15,15),,BF:NEXTI,R0
50 LINE(X8,Y8-1)-(X8+1+16*8,Y8-1)
60 LINE(X8,Y8+16*8)-(X8+16*8+1,Y8+16*8)
70 LINE(X8,Y8-1)-(X8,Y8+16*8)
80 LINE(X8+1+16*8,Y8-1)-(X8+16*8+1,Y8+16*8)
90 PRESET(X8+5,Y8-11):PRINT #1,"1 2 3 4 5 6 7 8"
100 PRESET(X8-7,Y8+5):PRINT #1,"1"
110 PRESET(X8-7,Y8+21):PRINT #1,"2"
120 PRESET(X8-7,Y8+37):PRINT #1,"3"
130 PRESET(X8-7,Y8+53):PRINT #1,"4"
140 PRESET(X8-7,Y8+69):PRINT #1,"5"
150 PRESET(X8-7,Y8+85):PRINT #1,"6"
160 PRESET(X8-7,Y8+101):PRINT #1,"7"
170 PRESET(X8-7,Y8+117):PRINT #1,"8"
180 PRESET(20,165):PRINT#1,"Ваш ход?:"
190 '***** Ввод начальной позиции коня *****
200 EL=100
210 P$=INPUT$(1):IF P$="" THEN 210
220 IF M$<>""THEN 240
230 PRESET(EL,165):PRINT #1,P$+", "
240 M$=M$+P$:PRESET(EL,165):PRINT #1,P$:EL=EL+16
250 IF LEN(M$)<2 THEN 210
260 M=VAL(LEFT$(M$,1)):N=VAL(RIGHT$(M$,1)):IF M<1 OR M>K OR N<1 OR N>E THEN CLS:P$="":M$="":GOTO
20
270 P$="":M$="":EL=0
280 DATA 1,1,1,2,7,10,16,23,13,1,1,2,7,15,0,0,0,192,240,112,56,24,24,24,56,48,16,8,252,254,0,0
290 FOR TI=1 TO 32:READ R:X0$=X0$+CHR$(R):NEXT
300 SPRITE$(1)=X0$
310 DIM A(K,E) 'Основная программа
320 G=8:I=M:J=N:A(I,J)=1:XX=(J-1)*16+X8:YY=(I-1)*16+Y8:PUT SPRITE 1,(XX,YY),10,1
330 FOR V=-2 TO 2:FOR H=-2 TO 2'Перебор всевозможных вариантов хода коня
340 IF V=0 OR H=0 OR ABS(V)=ABS(H) THEN 350 ELSE GOSUB 500 'Переход на подпрограмму, определяющую
оптимальный вариант хода коня.
350 NEXT H:NEXT V
360 COLOR15,1,8:PSET(190,45):PRINT #1,SED:COLOR1,15,8
370 COLOR15,1,8:PSET(190,60):PRINT #1,I;",";J:COLOR 1,15,8
380 SED=SED+1 'Число ходов
390 I=I+V2:J=J+H2:XX=(J-1)*16+X8:YY=(I-1)*16+Y8:A(I,J)=1
400 PUT SPRITE 1,(XX,YY),10,1 'Перемещение спрайта
410 IF (I-V2+J-H2)MOD2<>0 THEN GOTO 430
420 COLOR15,1,8:PRESET((J-H2-1)*16+X8+5,(I-V2-1)*16+Y8+5):PRINT #1,"*":COLOR1,15,8:GOTO 440
430 PRESET((J-H2-1)*16+X8+5,(I-V2-1)*16+Y8+5):PRINT #1,"*" 'Заполнение пройденных полей
```

```

440 PRESET(190,45):PRINT #1,SED
450 PRESET(215,45):PRINT #1,"ход"
460 PRESET(190,60):PRINT #1,I;"",";J
470 FOR II=1 TO K:FOR JJ=1 TO E
480 IF A(II,JJ)=1 THEN NEXT JJ:NEXT II:GOTO 610 ELSE IF G>0 THEN G=8:GOTO 330
490 'Подпрограмма, считающая число возможных ходов с тех полей,на которые конь мог бы пойти, и выделение из них
минимального.
500 IF I+V=>1 AND I+V=<K AND J+H=>1 AND J+H=<E THEN 510 ELSE RETURN
510 IF A(I+V,J+H)=0 THEN I=I+V:J=J+H ELSE RETURN
520 FOR V1=-2 TO 2:FOR H1=-2 TO 2
530 IF V1=0 OR H1=0 OR ABS(V1)=ABS(H1) THEN 560
540 IF I+V1>=1 AND I+V1<=K AND J+H1>=1 AND J+H1<=E THEN 550 ELSE 560
550 IF A(I+V1,J+H1)=0 THEN L=L+1:GOTO 560 ELSE 560
560 NEXT H1,V1
570 IF(M=4 AND N=1)OR(M=4 AND N=4)OR(M=7AND N=2) THEN 590 ELSE 580
580 IF L<G THEN G=L:V2=V:H2=H:GOTO 600 ELSE GOTO 600
590 IF L=<G THEN G=L:V2=V:H2=H
600 L=0:I=I-V:J=J-H:RETURN 'Конец подпрограммы
610 COLOR 15,1,8:PSET(20,165):PRINT #1,"":COLOR1,15,8
620 PRINT#1,"Продолжим наши игры?(д/н):"
630 P$=INPUT$(1):IF P$="" THEN 630
640 ON INSTR("ддДДууУуИиНнНН",P$) GOTO 660,660,660,660,660,660,660,660,670,670,670,670
650 P$="":GOTO 610
660 SCREEN0:RUN
670 CLS:PRESET(60,90):PRINT #1,"G O O D B Y E !":PUT SPRITE 1,(116,110),8,1
680 FOR T=0 TO 10000:NEXT:END

```

## Игрушка 37.1

[12-371.bas](#)

А теперь приглашаем Вас немного поиграть...

```

NEW
OK
10 COLOR 1,15,8:SCREEN 2,2
30 LINE(0,180)-STEP(255,12),8,BF
50 OPEN"grp:"FOR OUTPUT AS#1
70 DEFINT A-Z:OL=35:K=8:E=8
90 Y8=25:X8=70
110 FOR R0=Y8 TO Y8+64 STEP 64:FOR I=X8 TO X8+32*3 STEP 32:LINE(I+1,R0)-
STEP(15,15),,BF:LINE(I+17,R0+16)-STEP(15,15),,BF:LINE(I+1,R0+32)-
STEP(15,15),,BF:LINE(I+17,R0+48)-STEP(15,15),,BF:NEXTI,R0
130 LINE(X8,Y8-1)-(X8+1+16*8,Y8-1)
150 LINE(X8,Y8+16*8)-(X8+16*8+1,Y8+16*8)
170 LINE(X8,Y8-1)-(X8,Y8+16*8)
190 LINE(X8+1+16*8,Y8-1)-(X8+1+16*8,Y8+16*8)
210 PRESET(X8+5,Y8-11):PRINT #1,"1 2 3 4 5 6 7 8"
230 PRESET(X8-7,Y8+5):PRINT #1,"1"
250 PRESET(X8-7,Y8+21):PRINT #1,"2"
270 PRESET(X8-7,Y8+37):PRINT #1,"3"
290 PRESET(X8-7,Y8+53):PRINT #1,"4"
310 PRESET(X8-7,Y8+69):PRINT #1,"5"
330 PRESET(X8-7,Y8+85):PRINT #1,"6"
350 PRESET(X8-7,Y8+101):PRINT #1,"7"
370 PRESET(X8-7,Y8+117):PRINT #1,"8"
390 COLOR10,8,8:PRESET(5,182):PRINT #1,"SCORE:"
410 PRESET(6,182):PRINT #1,"SCORE:":COLOR1,15,8
430 COLOR10,8,8:PRESET(165,182):PRINT #1,"HISCORE:"
450 PRESET(166,182):PRINT #1,"HISCORE:":COLOR1,15,8
470 DATA 1,1,1,2,7,10,16,23,13,1,1,2,7,15,0,0,0,192,240,112,56,24,24,24,56,48,16,8,252,254,0,0
475 RESTORE 470:FOR TI=1 TO 32:READ R:X0$=X0$+CHR$(R):NEXT
490 SPRITE$(20)=X0$:X0$=""

```

```

491 RESTORE 1280:FOR TI=1 TO 32:READ R:X1$=X1$+CHR$(R):NEXT
492 SPRITE$(1)=X1$:X1$=""
493 RESTORE 1290:FOR TI=1 TO 32:READ R:X2$=X2$+CHR$(R):NEXT
494 SPRITE$(2)=X2$:X2$=""
495 RESTORE 1300:FOR TI=1 TO 32:READ R:X3$=X3$+CHR$(R):NEXT
496 SPRITE$(3)=X3$:X3$=""
497 RESTORE 1310:FOR TI=1 TO 32:READ R:X4$=X4$+CHR$(R):NEXT
498 SPRITE$(4)=X4$:X4$=""
499 RESTORE 1320:FOR TI=1 TO 32:READ R:X5$=X5$+CHR$(R):NEXT
500 SPRITE$(5)=X5$:X5$=""
501 SPRITE$(6)=SPRITE$(4)
502 SPRITE$(7)=SPRITE$(3)
503 SPRITE$(8)=SPRITE$(2)
504 RESTORE 1330:FOR TI=1 TO 32:READ R:X9$=X9$+CHR$(R):NEXT
505 SPRITE$(9)=X9$:X9$=""
506 RESTORE 1340:FOR TI=1 TO 32:READ R:X10$=X10$+CHR$(R):NEXT
507 SPRITE$(10)=X10$:X10$=""
508 SPRITE$(11)=SPRITE$(3)
509 RESTORE 1350:FOR TI=1 TO 32:READ R:X12$=X12$+CHR$(R):NEXT
510 SPRITE$(12)=X12$:X12$=""
511 RESTORE 1360:FOR TI=1 TO 32:READ R:X13$=X13$+CHR$(R):NEXT
512 SPRITE$(13)=X13$:X13$=""
513 SPRITE$(14)=SPRITE$(12)
514 SPRITE$(15)=SPRITE$(3)
515 SPRITE$(16)=SPRITE$(10)
530 DIM A(K,E):I=1:J=1:X5=X8:Y5=Y8:X6=X8+16*8:Y6=Y8+16*8:D0=16
550 XX=(J-1)*D0+X8:YY=(I-1)*D0+Y8:PUT SPRITE 5,(XX,YY),10,20
570 A$=INPUT$(1):IF A$="" THEN 570
590 Z$=CHR$(30)+CHR$(31)+CHR$(28)+CHR$(29)+CHR$(32)
610 W=INSTR(Z$,A$):ON W GOTO 650,690,710,730,750
630 GOTO 570
650 IF YY-D0<Y5 THEN A$="":GOTO 570
670 YY=YY-D0:I=I-1:PUT SPRITE5,(XX,YY),10,20:GOTO 570
690 IF YY+D0>=Y6 THEN A$="":GOTO 570 ELSE YY=YY+D0:I=I+1:PUT SPRITE5,(XX,YY),10,20:GOTO 570
710 IF XX+D0>=X6 THEN A$="":GOTO 570 ELSE XX=XX+D0:J=J+1:PUT SPRITE5,(XX,YY),10,20:GOTO 570
730 IF XX-D0<X5 THEN A$="":GOTO 570 ELSE XX=XX-D0:J=J-1:PUT SPRITE5,(XX,YY),10,20:GOTO 570
750 IF F>=1 THEN 790 ELSE A(I,J)=1:GOSUB1150:Q=5:GOSUB1010:IF(I+J)MOD2=0 THEN
COLOR15,1,8:PRESET(XX+5,YY+5):PRINT #1,"*":COLOR 1,15,8 ELSE PRESET(XX+5,YY+5):PRINT #1,"*"
770 F=1:K1=I:K2=J:GOTO 570
790 FOR V4=-2 TO 2:FOR H4=-2 TO 2
810 IF V4=0 OR H4=0 OR ABS(V4)=ABS(H4) THEN NEXT H4,V4 ELSE IF I=K1+V4 AND J=K2+H4 AND A(I,J)<>1
THEN 850 ELSE NEXT H4,V4:GOTO 570
830 GOTO 570
850 K1=I:K2=J:A(I,J)=1:GOSUB1150:Q=Q+5:GOSUB1010:IF (I+J)MOD2=0 THEN
COLOR15,1,8:PRESET(XX+5,YY+5):PRINT #1,"*":COLOR 1,15,8 ELSE PRESET(XX+5,YY+5):PRINT #1,"*"
870 FOR V5=-2 TO 2:FOR H5=-2 TO 2
890 IF V5=0 OR H5=0 OR ABS(V5)=ABS(H5) THEN 950
910 IF I+V5>K OR I+V5<1 OR J+H5>E OR J+H5<1 THEN 950 ELSE 930
930 IF A(I+V5,J+H5)=1 THEN 950 ELSE BOB=BOB+1
950 NEXT H5,V5
970 IF BOB<>0 THEN BOB=0:GOTO 570
990 GOTO 1050
1010 COLOR 15,8,8:PRESET(45,182):PRINT#1,Q
1030 PRESET(46,182):PRINT#1,Q:COLOR1,15,8:RETURN
1050 FOR L1=1 TO 8:PUT SPRITE 1,(119,79),8,L1:PUTSPRITE2,(135,79),8,L1+8:FORC=0 TO 35:NEXTC:NEXT
1070 XIT$=INKEY$:IFXIT$="" THEN 1050 ELSE PUT SPRITE1,(0,0),15,L1:PUT SPRITE 2,(16,0),15,L1+8
1090 FOR I1=1 TO 8:FORJ1=1 TO 8
1110 IF(I1+J1)MOD2=0 THEN NEXTJ1,I1 ELSECOLOR15,1,8:PSET((I1-1)*16+X8+5,(J1-1)*16+Y8+5):PRINT
#1,"*":COLOR1,15,8:NEXTJ1,I1
1130 GOSUB 1150:GOTO 1190
1150 COLOR 8,15,8:PSET(45,182):PRINT #1,Q
1170 PSET(46,182):PRINT #1,Q:COLOR 1,15,8:RETURN
1190 IF Q<=M1 THEN 1270
1210 COLOR 8,5,8:PSET(225,182):PRINT #1,M1:PSET(226,182):PRINT #1,M1:COLOR1,15,8

```

```

1230 M1=Q
1250 COLOR 5,8,8:PRESET(225,182):PRINT #1,M1:PRESET(226,182):PRINT #1,M1:COLOR1,15,8
1270 ERASE A:F=0:Q=0:CLOSE:GOTO 50
1280 DATA
0,0,127,128,128,190,179,179,190,179,179,179,190,128,128,127,0,0,255,0,0,51,51,51,51,30,12,12,12,
0,0,255
1290 DATA
0,0,0,0,0,127,128,191,190,179,190,128,127,0,0,0,0,0,0,0,255,0,51,51,30,12,0,255,0,0,0
1300 DATA 0,0,0,0,0,0,0,0,0,255,0,0,0,0,0,0,0,0,0,0,0,255,0,0,0,0,0
1310 DATA
0,0,0,0,0,127,128,190,179,190,191,128,127,0,0,0,0,0,0,0,255,0,12,30,51,51,0,255,0,0,0
1320 DATA
0,0,127,128,128,190,179,179,179,190,179,179,190,128,128,127,0,0,255,0,0,12,12,12,30,51,51,51,51,
0,0,255
1330 DATA 0,0,255,0,0,63,51,53,60,52,49,51,63,0,0,255,0,0,254,1,1,25,25,25,25,25,1,25,25,1,1,254
1340 DATA 0,0,0,0,0,255,0,63,60,53,63,0,255,0,0,0,0,0,0,0,254,1,25,25,1,25,1,254,0,0,0
1350 DATA 0,0,0,0,0,255,0,63,53,60,63,0,255,0,0,0,0,0,0,0,254,1,25,1,25,25,1,254,0,0,0
1360 DATA 0,0,255,0,0,63,51,49,52,60,53,51,63,0,0,255,0,0,254,1,1,25,25,1,25,25,25,25,25,1,1,254

```

## Задача 38

[12-38.bas](#)

Если в задаче меньше трёх переменных, это не задача;  
если больше восьми — она неразрешима.

—Закон Мэрфи

Существуют способы расстановки 8 ферзей на шахматной доске так, чтобы они не били друг друга. Составьте алгоритм, отыскивающий один из таких способов.

```

NEW
Ok
10 'Расстановка 8 ферзей. Минимальное время работы - 14 минут.
30 COLOR 1,15,8:SCREEN 2,1:VV=0:OPEN"grp:" AS#1
40 DATA a,b,c,d,e,f,g,h
50 A$=CHR$(0)+CHR$(16)+CHR$(146)+CHR$(84)+CHR$(40)+CHR$(124)+CHR$(0)+CHR$(0) ' "Внешний облик"
ферзей.
60 FOR J1=20 TO 160 STEP 20:FOR J2=20 TO 160 STEP 20
70 LINE(J2,J1)-(J2+20,J1+20),2,B
80 NEXT J2:PRESET(6,J1+6):PRINT #1,J1/20:NEXT J1
90 FOR I=1 TO 8:READ C$:PRESET (20*I+6,182):PRINT #1,C$:NEXT I 'Нарисовано поле.
100 SPRITE$(1)=A$:N1=1:N2=1:N3=1:N4=1:N5=1:N6=1:N7=1:N8=1:GOSUB 500
110 PRESET(184,100):PRINT #1,"Восемь":PRESET(184,110):PRINT #1,"ферзей!":GOSUB 470:SPRITE$(1)=A$
120 'Проверка условий (строки 120-400).
130 FOR N1=1 TO 8:FOR N2=1 TO 9:IF N2=N1 OR N1-N2=1 OR N2-N1=1 THEN 420
140 IF N2=9 THEN N2=1:GOTO 430
150 FOR N3=1 TO 9:IF N3=N1 OR N3=N2 OR N1-N3=2 OR N2-N3=1 OR N3-N1=2 OR N3-N2=1 THEN 410
160 IF N3=9 THEN N3=1:GOTO 420
170 FOR N4=1 TO 9:IF N4=N1 OR N4=N2 OR N4=N3 THEN 400
180 IF N1-N4=3 OR N2-N4=2 OR N3-N4=1 OR N4-N1=3 OR N4-N2=2 OR N4-N3=1 THEN 400
190 IF N4=9 THEN N4=1:GOTO 410
200 FOR N5=1 TO 9:IF N5=N1 OR N5=N2 OR N5=N3 OR N5=N4 THEN 390
210 IF N1-N5=4 OR N2-N5=3 OR N3-N5=2 OR N4-N5=1 OR N5-N1=4 OR N5-N2=3 OR N5-N3=2 OR N5-N4=1 THEN
390
220 IF N5=9 THEN N5=1:GOTO 400
230 FOR N6=1 TO 9:IF N6=N1 OR N6=N2 OR N6=N3 OR N6=N4 OR N6=N5 THEN 380
240 IF N1-N6=5 OR N2-N6=4 OR N3-N6=3 OR N4-N6=2 OR N5-N6=1 THEN 380
250 IF N6-N1=5 OR N6-N2=4 OR N6-N3=3 OR N6-N4=2 OR N6-N5=1 THEN 380
260 IF N6=9 THEN N6=1:GOTO 390

```

```

270 FOR N7=1 TO 9:IF N7=N1 OR N7=N2 OR N7=N3 OR N7=N4 OR N7=N5 OR N7=N6 THEN 370
280 IF N1-N7=6 OR N2-N7=5 OR N3-N7=4 OR N4-N7=3 OR N5-N7=2 OR N6-N7=1 THEN 370
290 IF N7-N1=6 OR N7-N2=5 OR N7-N3=4 OR N7-N4=3 OR N7-N5=2 OR N7-N6=1 THEN 370
300 IF N7=9 THEN N7=1:GOTO 380
310 FOR N8=1 TO 9:IF N8=N1 OR N8=N2 OR N8=N3 OR N8=N4 OR N8=N5 OR N8=N6 OR N8=N7 THEN 360
320 IF N1-N8=7 OR N2-N8=6 OR N3-N8=5 OR N4-N8=4 OR N5-N8=3 OR N6-N8=2 OR N7-N8=1 THEN 360
330 IF N8-N1=7 OR N8-N2=6 OR N8-N3=5 OR N8-N4=4 OR N8-N5=3 OR N8-N6=2 OR N8-N7=1 THEN 360
340 IF N8=9 THEN N8=1:GOTO 370
350 GOSUB 500:GOSUB 460
360 NEXT N8
370 NEXT N7
380 NEXT N6
390 NEXT N5
400 NEXT N4
410 NEXT N3
420 NEXT N2
430 NEXT N1
440 SCREEN3:PRESET(20,10):PRINT #1,"ВСЕГО":PRESET(5,80):PRINT #1,"ХОРОШЕГО"
450 GOTO 450 'Конец программы.
460 VV=VV+1:PRESET(182,2):PRINT #1,"Способ":PRESET(182,11):PRINT #1,"номер ";VV
470 PRESET(190,160):PRINT #1,"Нажмите":PRESET(190,170):PRINT #1,"любую":PRESET(190,180):PRINT
#1,"клавишу"
480 F$=INPUT$(1):LINE(182,0)-(256,192),15,BF:RETURN
490 'Расстановка 8 ферзей.
500 PUT SPRITE 1,(20*N1+2,20*1+2),8,1
510 PUT SPRITE 2,(20*N2+2,20*2+2),8,1
520 PUT SPRITE 3,(20*N3+2,20*3+2),8,1
530 PUT SPRITE 4,(20*N4+2,20*4+2),8,1
540 PUT SPRITE 5,(20*N5+2,20*5+2),8,1
550 PUT SPRITE 6,(20*N6+2,20*6+2),8,1
560 PUT SPRITE 7,(20*N7+2,20*7+2),8,1
570 PUT SPRITE 8,(20*N8+2,20*8+2),8,1
580 RETURN

```

## Задача 39

[12-39.bas](#)

Некоторые натуральные числа могут быть представлены в виде суммы кубов целых неотрицательных чисел: например,  $9=2^3+1^3$ ,  $27=3^3+0^3$ . Составьте алгоритм, отыскивающий наименьшее натуральное число, имеющее два разных таких представления. (Представления  $9=2^3+1^3 = 1^3+2^3$  считаются одинаковыми.)

```

NEW
Ok
5 PRINT "В каком диапазоне натуральных чисел будем решать задачу?"
10 INPUT P:INPUT N:FOR L=P TO N
20 FOR I=FIX(L^(1/3)+.5) TO 0 STEP(-1):FOR J=0 TO I
30 IF I^3+J^3=L THEN K=K+1:NEXT I ELSE NEXT J,I
40 IF K>=2 THEN GOSUB 100:RA=1:K=0:NEXT L ELSE K=0:NEXT L
50 IF RA<1 THEN PRINT "Таких чисел нет!"
60 END
100 FOR I=FIX(L^(1/3)+.5) TO 0 STEP(-1):FOR J=0 TO I
110 IF I^3+J^3=L THEN PRINT STR$(L);"=";MID$(STR$(I),2);"^3+";MID$(STR$(J),2);"^3":NEXT I ELSE
NEXT J,I
120 RETURN
run
В каком диапазоне натуральных чисел будем решать задачу?
? 1720
? 1740
1729=12^3+1^3
1729=10^3+9^3
Ok

```

## Задача 40

[12-40.bas](#)

Составьте алгоритм, вычисляющий по заданным вещественным числам A, B, C, D величины  $A \cdot C + B \cdot D$  и  $A \cdot D - B \cdot C$ . Этот алгоритм может использовать промежуточные величины, операции сложения, вычитания и умножения, причём умножение должно выполняться не более трёх раз.

```
NEW
Ok
50 INPUT "Чему равны A,B,C,D";A,B,C,D
60 P0=(A+B)*(C+D):P1=A*D:P2=B*C:S1=P0-(P1+P2):S2=P1-P2
80 PRINT "AC+BD=";S1,"AD-BC=";S2
90 END
```

## Задача 41

[12-41.bas](#)

Подсчитайте, сколько символов слова X используется при написании слова Y.

```
NEW
Ok
10 R=0:P$="":INPUT "Введите первое слово";X$
20 INPUT "Введите второе слово";Y$
25 IF LEN(X$)=0 OR LEN(Y$)=0 THEN PRINT "Не балуйтесь!":RUN
30 L=LEN(X$):U=LEN(Y$)
40 FOR I=1 TO L-1:FOR J=I+1 TO L
50 IF MID$(X$,I,1)=MID$(X$,J,1) THEN MID$(X$,J,1)=CHR$(0)
60 NEXT J,I
70 FOR I=1 TO L:FOR J=1 TO U
80 IF MID$(X$,I,1)=MID$(Y$,J,1) AND MID$(X$,I,1)<>CHR$(0) THEN
P$=P$+MID$(X$,I,1):R=R+1:N$=MID$(X$,I,1):NEXT I
90 IF N$=MID$(X$,L,1) THEN 120
100 NEXT J
110 NEXT I:X$="":Y$=""
120 PRINT "Из первого слова используется";R;"букв для написания второго"
125 PRINT
130 IF LEN(P$)>=2 THEN PRINT "Эти буквы: ";P$
140 IF LEN(P$)=1 THEN PRINT "Эта буква: ";P$
150 RUN 10
```

## Задача 42

[12-42.bas](#)

Цифры 1, 2, 3, 4, 5, 6, 7, 8, 9 соединить знаками действий +, -, \*, / так, чтобы в результате выполнения действий слева направо (без учёта приоритета!) получилось заданное число M.

```
NEW
Ok
22 CLS:PRINT:PRINT:INPUT "Число M ";M:DIM A$(17)
50 FOR I=1 TO 4:FOR J=1 TO 4:FOR K=1 TO 4:FOR L=1 TO 4:FOR N=1 TO 4:FOR S=1 TO 4:FOR P=1 TO 4:FOR R=1 TO 4:A=1
70 IF I=1 THEN A=A+2:A$(2)="+" ' *** В строках 70-390 осуществляется
80 IF I=2 THEN A=A-2:A$(2)="-" ' полный перебор возможных комбинаций
90 IF I=3 THEN A=A*2:A$(2)="*" ' расположения символов +,-,*,/ меж-
100 IF I=4 THEN A=A/2:A$(2)="/" ' ду цифрами 1,2,3,4,5,6,7,8,9 ***
110 IF J=1 THEN A=A+3:A$(4)="+"
120 IF J=2 THEN A=A-3:A$(4)="-"
```

```

130 IF J=3 THEN A=A*3:A$(4)="*"
140 IF J=4 THEN A=A/3:A$(4)="/"
150 IF K=1 THEN A=A+4:A$(6)="+"
160 IF K=2 THEN A=A-4:A$(6)="-"
170 IF K=3 THEN A=A*4:A$(6)="*"
180 IF K=4 THEN A=A/4:A$(6)="/"
190 IF L=1 THEN A=A+5:A$(8)="+"
200 IF L=2 THEN A=A-5:A$(8)="-"
210 IF L=3 THEN A=A*5:A$(8)="*"
220 IF L=4 THEN A=A/5:A$(8)="/"
230 IF N=1 THEN A=A+6:A$(10)="+"
240 IF N=2 THEN A=A-6:A$(10)="-"
250 IF N=3 THEN A=A*6:A$(10)="*"
260 IF N=4 THEN A=A/6:A$(10)="/"
270 IF S=1 THEN A=A+7:A$(12)="+"
280 IF S=2 THEN A=A-7:A$(12)="-"
290 IF S=3 THEN A=A*7:A$(12)="*"
300 IF S=4 THEN A=A/7:A$(12)="/"
310 IF P=1 THEN A=A+8:A$(14)="+"
320 IF P=2 THEN A=A-8:A$(14)="-"
330 IF P=3 THEN A=A*8:A$(14)="*"
340 IF P=4 THEN A=A/8:A$(14)="/"
350 IF R=1 THEN A=A+9:A$(16)="+"
360 IF R=2 THEN A=A-9:A$(16)="-"
370 IF R=3 THEN A=A*9:A$(16)="*"
380 IF R=4 THEN A=A/9:A$(16)="/"
390 IF A=M THEN G=G+1:GOTO 420
395 LOCATE 16,7:PRINTI;J;K;L;N;S;P;R;
400 NEXT R,P,S,N,L,K,J,I
410 IF W<=4THENPRINT"УБЫ.....":END ELSE END
420 FOR Y=1 TO 17 STEP 2:A$(Y)=STR$((Y+1)/2):NEXTY
430 LOCATE0,10+G:FOR Y=1 TO 17
440 PRINTA$(Y);:NEXT Y
450 IF I=4ANDR=4 THEN ?:"BCE.":END ELSE W=4:GOTO 400
run
Число M ? 45
  1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9
  1 + 2 + 3 - 4 * 5 - 6 - 7 + 8 * 9
  1 + 2 + 3 - 4 * 5 - 6 * 7 + 8 + 9
  1 + 2 + 3 * 4 + 5 + 6 - 7 + 8 + 9
  ...
Ok

```

## Задача 43

Составить программу, определяющую количество «счастливых» автобусных билетов (номер билета шестизначный). Билет считается «счастливым», если сумма трёх старших цифр его номера совпадает с суммой трёх младших цифр.

*Первый способ.*

[12-431.bas](#)

```

10 INPUT N,M
20 FOR T=INT(N/1000) TO INT(M/1000)
21 I=T
30 S=(I-INT(I/10)*10)+(INT(I/10)-INT(I/100)*10)+INT(I/100):L=0:P=9
40 IF S<10 THEN P=S ELSE IF S>=19 THEN L=S-18 ELSE
50 FOR J=L TO P:FOR K=L TO P:FOR N=L TO P
60 IF S=J+K+N THEN C=((I*10)+J)*10+K:IF C<=M AND C>=N THEN PRINT C;:W=W+1
80 NEXT N,K,J,T:PRINT W
run
? 0,1000000

```

```
0 001001 ... 999999 55252
```

Ok

Компьютер считает  $\approx 4.5$  ч!

Второй способ.

[12-432.bas](#)

```
10 TIME=0:DIM A%(27)
20 FOR I=0 TO 9:FOR J=0 TO 9:FOR K=0 TO 9
30 A%(I+J+K)=A%(I+J+K)+1:NEXT K,J,I
40 N=1:FOR I=1 TO 27:A%(I)=A%(I)^2
50 N=N+A%(I):NEXT I
60 PRINT"Счастливых билетов: ";N:PRINT"Время работы программы: ";INT(TIME/60);"с"
гип
Счастливых билетов: 55252
Время работы программы: 14 с
Ok
```

Отметим, что количество  $C_{2n}^{2 \times n}$  счастливых  $2 \times n$ -значных чисел (ясно, что под этим понимается: например, 00351 20115 — счастливое десятизначное число) определяется явной формулой:

$$C_{2n}^{2 \times n} = \frac{1}{\pi} \times \int_0^{\pi} \left[ \frac{\sin(10 \times x)}{\sin(x)} \right]^{2 \times n} dx$$

Интересно отметить, что количество счастливых  $2n$ -значных чисел в системе счисления по основанию  $k$ :

$$C_{2n}^k = \frac{1}{\pi} \times \int_0^{\pi} \left[ \frac{\sin(k \times x)}{\sin(x)} \right]^{2 \times n} dx$$

## Задача 44

[12-44.bas](#)

Написать программу, реализующую римскую арифметику.

Римская система счисления относится к непозиционным системам. Например, запишем число 88 римскими знаками — LXXXVIII. В этой записи смысл каждого знака (символа) не зависит от того места, на котором он стоит. Здесь, например, цифра X повторяется 3 раза и каждый раз означает одну и ту же величину — десять единиц.

```
NEW
Ok
20 DIM C$,B,B$,W,A,A$,N$,N,T$,L$,K$,T,L,X$
30 PRINT "Римская арифметика"
40 PRINT "До начала работы нажмите клавишу"+CHR$(34)+"CAPS"+CHR$(34)
50 PRINT "ВНИМАНИЕ!"
60 PRINT "У древних римлян не было цифры 0!"
70 PRINT "Действия производились только над положительными целыми числами."
80 PRINT"При делении за результат принимается целая часть частного."
90 GOTO 240
100 ' ***** From ARABIC to ITALIC *****
110 DATA 1000,M,900,CM,500,D,400,CD,100,C,90,XC,50,L,40,XL,10,X,9,IX,5,V,4,IV,1,I
120 RESTORE 110:C$=""
130 READ B,B$
```



```

140 IF B<=W THEN C$=C$+B$:W=W-B:GOTO 140
150 IF W>0 GOTO 130
160 RETURN
170 '          ***** From ITALIC to ARABIC *****
180 DATA 1000,M,900,CM,500,D,400,CD,100,C,90,XC,50,L,40,XL,10,X,9,IX,5,V,4,IV,1,I
190 RESTORE 180:N=0
200 READ A,A$
210 IF A$=LEFT$(N$,LEN(A$))THEN N$=RIGHT$(N$,LEN(N$)-LEN(A$)):N=N+A:GOTO 210
220 IF N$>" " GOTO 200
230 RETURN
240 LINEINPUT "Первое число:";T$
250 LINEINPUT "Второе число:";L$
260 LINEINPUT "Знак операции (+, -, *, /, ^):";K$
270 Y$=INKEY$
280 INPUT "А,может,Вы желаете извлечь корень? (Да/Нет) ";Y$
290 IF Y$="Д" OR Y$="д" THEN N$=T$:GOSUB 180:T=N:W=INT(SQR(T)):GOSUB 110:? "Корень из ";T$;"
равен: ";C$:GOTO 400
300 IF Y$="Н" OR Y$="н" THEN 310 ELSE 270
310 N$=T$:GOSUB 180:T=N:N$=L$:GOSUB 180:L=N
320 IF K$="+" THEN W=T+L:GOTO 370
330 IF K$="-" THEN W=T-L:GOTO 370
340 IF K$="*" THEN W=T*L:GOTO 370
350 IF K$="/" THEN W=INT(T/L):GOTO 370
360 IF K$="^" THEN W=T^L
370 IF W<0 THEN W=-W:X$="-" ELSE X$=""
380 GOSUB 110
390 PRINT T$;K$;L$;X$;C$
400 RUN 90

```

## Задача 45

[12-45.bas](#)

Написать программу, которая позволяет генерировать (получать) всевозможные сочетания подмножества  $\{1, 2, \dots, N\}$  по  $K$  элементов.

Итак, основным множеством является подмножество натуральных чисел  $\{1, 2, \dots, n\}$ .

Наиболее естественно рассматривать представление сочетания  $A_1, A_2, \dots, A_K$  в виде вектора

$C = (C_1, C_2, \dots, C_N)$ , у которого компоненты расположены в порядке возрастания слева направо (т.е.  $C_i < C_{i+1}$  для любого  $i$ ), и порождать эти векторы в лексикографическом порядке. Начинается порождение с сочетания  $(1, 2, \dots, K)$ .

Каждое следующее сочетание строится из предыдущего с помощью следующих простых действий: в ходе просмотра текущего сочетания справа налево в нем находится самый первый элемент, который ещё не достиг своего максимального значения; этот элемент увеличивается на единицу, а всем элементам справа от него последовательно присваиваются новые наименьшие возможные значения.

```

NEW
OK
10 INPUT"Укажите N и K";N,K:IF K>N OR K<1 THEN GOTO 10
30 DIM A(N),C(K):FOR I=1TO K:C(I)=I:NEXT
60 P=0:FOR I=1 TO K:IF C(I)=N-K+I THEN P=P+1:NEXT I ELSE NEXT I
80 IF P=K THEN FOR I=1 TO K:PRINT C(I);:NEXT I:END '→
90 FOR J=1 TO K:PRINT C(J);:NEXT J:PRINT
100 FOR J=K TO 1 STEP -1
110 IF C(J)=N-K+J THEN NEXT J:GOTO 60 ELSE C(J)=C(J)+1:IF J+1>K THEN GOTO 60 ELSE FOR I=J+1 TO

```

```

K:C(I)=C(I-1)+1:NEXT I:GOTO 60
run
Укажите N и K? 3,2
1 2
1 3
2 3
Ok

```

## Задача 46

12-46.bas

Что наша «Жизнь»? Игра!

—М.Гарднер

Написать программу, реализующую на текстовом экране алгоритм игры «Жизнь».

Игра «Жизнь» придумана сотрудником Кембриджского университета Дж. Конвеем. «Жизнь» — многоклеточное сообщество, населяющее пустыню, которая представляет собой прямоугольную решётку, каждая ячейка которой вмещает одну клетку Жизни. Мерой течения времени служит смена поколений Жизни, приносящая в колонию клеток смерть и рождение.

Чтобы проследить за историей развития колонии, разместим в пустыне клетки Жизни в их начальном положении. Смена поколений будет происходить по следующим правилам.

1. Соседями клетки считаются все клетки, находящиеся в восьми ячейках, расположенных рядом с данной по горизонтали, вертикали или диагонали.

M-1, K-1	M-1, K	M-1, K+1
M, K-1	M, K	M, K+1
M+1, K-1	M+1, K	M+1, K+1

2. Если у некоторой клетки меньше двух соседей, она погибает от одиночества. Если клетка имеет больше трёх соседей, она погибает от тесноты.
3. Если рядом с пустой ячейкой окажутся ровно три соседних клетки Жизни, то в этой ячейке рождается новая клетка.
4. Гибель и рождение происходят в момент смены поколений. Таким образом, гибнущая клетка может способствовать рождению новой, но рождающаяся клетка не может воскресить гибнущую, и гибель одной клетки, уменьшив локальную плотность населения, не может предотвратить гибель другой.

Колония может все время расти, непрерывно меняя своё расположение, форму или число клеток. Однако чаще колония становится в конце концов *стационарной*, начиная циклически повторять один и тот же конечный набор состояний. Длина цикла называется *периодом* колонии. (По этому определению период мёртвой и пустой колонии равен единице.)

Измените приведённую ниже программу так, чтобы она выявляла стационарные колонии и сообщала о них.

Можете ли Вы придумать хоть какой-нибудь алгоритм, не использующий запоминания всех предыдущих поколений, который мог бы распознать любую стационарную колонию?

История жизни колонии зачаровывает, но она будет ещё увлекательнее, если предстанет в цвете. Каждой клетке при рождении может быть приписан некоторый цвет, определяемый, возможно, её поколением или генами, переданными ей родителями. Циклические, но при этом движущиеся колонии (а таких немало!) великолепны в своём сверкающем многоцветном наряде.

Любая колония имеет преемника, но не у каждой есть предшественник. Такие изолированные колонии называются *садами Эдема*. Сад Эдема можно увидеть, только если поместить его на плоскость в качестве начальной конфигурации. Подумайте, как использовать программу для нахождения сада Эдема.

```

NEW
Ok
3 PRINT"Введите размеры поля:
5 INPUT"3≤ N ≤10";N
6 INPUT"3≤ M ≤14";M
10 K=N-1:L=M-1
20 DIM A(N,M),B(K,L),A1$(N,M),B1$(K,L)
30 W=RND(-TIME):CLS:PRINT"Игра";CHR$(34);"ЖИЗНЬ";CHR$(34):LOCATE 0,4,0
40 FOR I=1 TO K
50 FOR J=1 TO L
60 A(I,J)=INT(2*RND(1)):IF A(I,J)=1 THEN A1$(I,J)=CHR$(155) ELSE A1$(I,J)="*"
70 PRINT A1$(I,J);SPC(2);:NEXT J:PRINT:PRINT:NEXT I
80 P=0
90 FOR I=1 TO K:FOR J=1 TO L
100 P=A(I-1,J-1)+A(I-1,J)+A(I-1,J+1)+A(I,J-1)+A(I,J)+A(I+1,J-1)+A(I+1,J)+A(I+1,J+1)
110 IF A(I,J)=0 THEN IF P=3 THEN B(I,J)=1 ELSE B(I,J)=0 ELSE IF P=2 OR P=3 THEN B(I,J)=1 ELSE
B(I,J)=0
120 NEXT J:NEXT I
130 FOR I=1 TO K:FOR J=1 TO L:IF B(I,J)<>A(I,J)THEN 150 ELSE NEXT:NEXT
140 PRINT "Колония стационарна!"
145 GOTO 145
150 LOCATE0,4
160 FOR I=1 TO K:FOR J=1 TO L
170 A(I,J)=B(I,J)
180 IF B(I,J)=1 THEN B1$(I,J)=CHR$(155)ELSE B1$(I,J)="*"
190 PRINT B1$(I,J);SPC(2);:NEXT:PRINT:PRINT:NEXT
200 LOCATE0,0,0:GOTO80

```

## Задача 47

[12-47.bas](#)

Написать программу, реализующую проверку правильности расстановки круглых скобок в математических выражениях.

```

NEW
Ok
10 LINEINPUT "Введите строку, содержащую открывающие и закрывающие круглые скобки: ";A$
15 IF LEFT$(A$,1)<>"(" THEN GOSUB45:END
20 FOR I=1 TO LEN(A$)
25 IF MID$(A$,I,1)="(" THEN K=K+1 ELSE IF MID$(A$,I,1)=")" THEN K=K-1
30 IF K<0 THEN GOSUB45:END ELSE NEXT
35 IF K<>0 THEN GOSUB45 ELSE PRINT"Скобки расставлены правильно!"
40 END
45 PRINT"Скобки расставлены неправильно!":RETURN

```

Приведём алгоритм, который для любого слова в алфавите  $\{(,)\}$  позволяет выяснить, является ли это слово правильным скобочным выражением. Этот алгоритм называется способом *подсчёта скобок*. Сопоставим слову  $A$  в алфавите  $\{(,)\}$ , состоящему из  $n$  букв, числовой массив  $A_1, A_2, \dots, A_n$ , заменяя каждую левую скобку на 1, и каждую правую скобку — на -1. Этот массив назовём *изображением* слова  $A$ . Имеет место следующая

**Теорема.** Массив  $A_1, A_2, \dots, A_n$  где  $A_i = \pm 1, i = 1, 2, \dots, n$  является изображением правильного скобочного выражения тогда и только тогда, когда

$$\bullet \quad A_1 = 1;$$

- для каждого  $k$  такого, что  $1 < k < n$  выполнено  $\sum_{i=1}^k A_i \geq 0$  ;
- $\sum_{i=1}^n A_i = 0$  .

Переменная K в программе первоначально приобретает значение 0, затем мы просматриваем слово слева направо и всякий раз, встретив левую скобку, прибавляем к K единицу, а встретив правую скобку — единицу вычитаем. Теорема утверждает, что мы имеем дело с правильным скобочным выражением тогда и только тогда, когда значение K все время неотрицательно и в конце просмотра становится равным нулю. Весь анализ проводится во время одного просмотра выражения.

## Задача 48

[12-48.bas](#)

В банке находятся белые и чёрные зерна кофе, причём количество их заранее неизвестно. Из банки наугад извлекаются два зерна, причём если они оказываются одинакового цвета, то чёрное зерно возвращается в банку, а если они разного цвета, то в банку возвращается белое зерно.

Требуется при помощи компьютера качественно описать множества белых и чёрных зёрен, первоначально находившихся в банке, при условии, что последнее зерно, оставшееся в банке — чёрное (соответственно — белое).

```
NEW
OK
10 L=INT(20*RND(-TIME)+5)
11 FOR I=1 TO L
12   IF RND(-TIME)<.5
      THEN P$=P$+"1"
      ELSE P$=P$+"0":S=S+1
13 NEXT
14 IF S MOD 2=1
      THEN PRINT"Осталось белое зерно!"
      ELSE PRINT"Осталось черное зерно!"
15 L=LEN(P$):R=INT(RND(-TIME)*L+1):R1=INT(RND(-TIME)*L+1)
16 IF R=R1 THEN GOTO 15
20 Q$="":S$=MID$(P$,R,1):S1$=MID$(P$,R1,1)
30 FOR K=1 TO LEN(P$)
40   IF K<>R AND K<>R1
      THEN Q$=Q$+MID$(P$,K,1)
50 NEXT
60 IF S$=S1$
      THEN Q$=Q$+"1"
      ELSE Q$=Q$+"0"
70 P$=Q$:IF LEN(P$)>1
      THEN 15
      ELSE PRINT"...а белых зерен в банке изначально было";S
```

## Диск с примерами

[Загрузить образ диска](#)

 [Открыть диск в WebMSX](#)

## XII.2. Задачи для самостоятельного решения

Один человек купил трёх коз и заплатил 3 рубля.

Спрашивается: по чему каждая коза пошла?

—Старинная задача-шутка (конец XVI века)

1. Для запоминания числа  $\pi$  иногда используют «магические» фразы, например: «это я знаю и помню прекрасно  $\pi$  многие знаки мне лишни напрасны» или «кто и шутя и скоро пожелает  $\pi$  узнать число уж знает». Число букв в каждом слове любой из данных фраз представляет собою некоторую цифру числа  $\pi$ : «это»—3, «я»—1, «знаю»—4 и т.д.

Вывести на печать число  $\pi$ , используя любую из указанных фраз.

2. В одной математической книге наборщик вместо выражения  $2^5 \cdot 9^5$  набрал 2592. Корректор эту серьёзную ошибку пропустил, и книга вышла в свет в таком виде. Эту книгу изучало много математиков, все предыдущие и последующие вычисления проверялись, а эту явную ошибку, пропущенную наборщиком и корректором, никто не заметил.

Как же это могло случиться? Дело в том, что  $2^5 \cdot 9^2 = 2592$ . Это, по-видимому, единственный в своём роде числовой курьёз! Проверьте.

3. Написать программу игры на ЭВМ под названием «быки и коровы». Правила игры чрезвычайно просты. Один из партнёров — ведущий. Он задумывает любое четырёхзначное число. Требуется лишь, чтобы ни одна из составляющих это число цифр не повторялась. Задача другого партнёра (назовём его просто игроком) — отгадать это число. Игрок называет пробные числа. Игра заканчивается, если названо число, задуманное ведущим. Чем меньше ходов-попыток потребуется для этого игроку, тем лучше он играет. Ведущий помогает игроку подсказками. Что это за подсказки, — рассмотрим на примере.

Предположим, задумано число 9480. Назовём его *кодом*. Как и предписывается условиями игры, все цифры в коде различны. Игрок называет своё пробное число 7984. Ведущий сравнивает его с кодом 9480. Так как числа не совпадают, то игра не закончена. Ведущий сообщает игроку количество «быков» и «коров», содержащихся в предложенном им пробном числе. «Быками» называют те его цифры, которые по значению и позиции совпадают с соответствующими цифрами задуманного кода. В нашем примере «бык» один — цифра 8. «Коровы» — это цифры, которые совпадают с цифрами задуманного кода, но находятся в иных позициях. В числе 7984 «коров» две: 9 и 4. Понятно, что отгадать код или назвать четыре «быка» — это одно и то же.

4. Написать программу определения соответствующего дня недели по известным целым числам:  $J$  — число,  $M$  — месяц,  $A$  — год, применяя метод Лемуара М., который заключается в следующем:

1) вычислить величину  $N$ :

- если месяц — январь или февраль високосного года, то  $N=1$ ;
- если месяц — январь или февраль обычного года, то  $N=2$ ;
- в других случаях  $N=0$ .

Чтобы узнать, является ли год високосным, можно действовать так:

- а) разложить год на две части  $A_1$  и  $A_2$ , причём:  $A_1$  содержит 2 старшие цифры года,  $A_2$  содержит 2 младшие цифры года;
- б) если  $A_2=0$  и  $A_1$  делится на 4, то год високосный;
- в) если  $A_2 \neq 0$ , то год високосный, если  $A_2$  делится на 4;

2) вычислить «код» дня  $C$  по формуле:  $C = [365.25 \cdot A_2] + [30.56 \cdot M] + J + N$ ;

3) вычислить остаток  $S$  от деления  $C$  на 7: если  $S=0$ , то день — среда, если  $S=1$ , то день — четверг, если  $S=2$ , то день — пятница, ..., если  $S=6$ , то день — вторник.

Незатейливые головоломки о целых числах веками служили источником обновления математики.

—Г.Д.Биркгоф

5. Гольдбахом было высказано предположение, что каждое чётное число, большее или равное 4, представимо в виде суммы двух простых.

Это предположение до сих пор не доказано и не опровергнуто. Написать программу проверки этой гипотезы для данного чётного числа. Результатом выполнения программы должен быть вывод самого числа, если не удалось найти пару простых слагаемых, и вывод пары соответствующих простых чисел, если таковая пара найдена.

6. Составить программу поиска среди чисел  $n, n+1, \dots, 2n$  так называемых *близнецов*, т.е. двух простых чисел, разность которых равна 2. Если близнецы в указанном диапазоне найдены, то вывести 1, если близнецов нет, то вывести 0.
7. Написать игровую программу «Игры на дорожке». Суть её в том, что два противника двигают фишки, находящиеся на противоположных сторонах дорожки, разбитой на  $n$  полей, например, на одной из вертикалей шахматной доски, где дорожка содержит 8 полей. Ходы делаются поочередно. За один ход каждый участник может подвинуть свою фишку не более чем на  $m$  полей вперёд или назад. При этом нельзя перепрыгивать через фишку партнёра и покидать пределы дорожки. Проигрывает тот, у которого не окажется ходов, т.е. чья фишка будет загнана в одно из двух крайних полей дорожки.

8. Написать программу, определяющую показатель, с которым данное простое  $p$  входит в произведение  $n!$ . Использовать

$$\alpha = \left[ \frac{n}{p} \right] + \left[ \frac{n}{p^2} \right] + \left[ \frac{n}{p^3} \right] + \dots + \left[ \frac{n}{p^l} \right],$$

формулу:

○  $[ ]$  — символ выделения целой части,

$$l = \left[ \frac{\lg(n)}{\lg(p)} \right].$$

○  $\alpha$  — искомый показатель,

Например, если  $n = 367$ ,  $p = 3$ , то  $\alpha = 180$ .

*Замечание.* Так можно разложить  $n!$  на простые множители, беря за  $p$  последовательно все простые числа, меньшие  $n$ .

9. Найти число делителей данного числа  $a$ . Известно, что если  $a = p_1^{\alpha_1} \times \dots \times p_k^{\alpha_k}$  — каноническое разложение числа  $a$ , и  $\sigma(a)$  — число делителей числа  $a$ , то  $\sigma(a) = (\alpha_1 + 1) \times (\alpha_2 + 1) \times \dots \times (\alpha_k + 1)$ .

Например:  $\sigma(720) = 30$ .

10. Найти сумму делителей данного числа  $a$ . Известно, что если  $a = p_1^{\alpha_1} \times \dots \times p_k^{\alpha_k}$  — каноническое разложение числа  $a$ , и  $S(a)$  — сумма делителей числа  $a$ , то  $S(a) = \frac{(p_1^{\alpha_1+1} - 1)}{p_1 - 1} \times \dots \times \frac{(p_k^{\alpha_k+1} - 1)}{p_k - 1}$ .
- Например:  $S(720) = 2418$ .

11. В разложении числа  $A$  на простые множители некоторые из них могут повторяться. Обозначая буквами  $p_1, p_2, \dots, p_k$  различные из них и буквами  $\alpha_1, \alpha_2, \dots, \alpha_k$  кратности их вхождения в  $A$ , получим так называемое *каноническое разложение* числа  $A$  на множители:  $a = p_1^{\alpha_1} \times p_2^{\alpha_2} \times \dots \times p_k^{\alpha_k}$ .

Написать программу нахождения канонического разложения данного числа  $A$ .

12. Задана последовательность из  $N$  чисел. Найти самую длинную подпоследовательность, обладающую следующим свойством:  $A(i) < A(i+1) > A(i+2) < A(i+3) > A(i+4) \dots$ , для некоторого  $i \in [1, N]$ .

13. *Дружественными* числами называется пара чисел, обладающих таким свойством: сумма собственных делителей первого из них равна второму числу, а сумма собственных делителей второго числа равна первому числу. Например, сумма делителей числа 220 равна  $1+2+4+5+10+11+20+22+44+55+110=284$ , а сумма делителей числа 284:  $1+2+4+71+142=220$ , поэтому числа 220 и 284 — дружественная пара.

Вторая дружественная пара (1184 и 1210) была найдена в 1867 году шестнадцатилетним итальянцем Б. Паганини.

Написать программу для нахождения дружественных чисел, используя способ, указанный ещё в 9-м веке арабским математиком Сабитом ибн Корра.

**Теорема Сабита.** Если все три числа  $p = 3 \times 2^{n-1} - 1$ ,  $q = 3 \times 2^n - 1$ ,  $r = 9 \times 2^{2n-1} - 1$  — простые, то числа  $A = 2^n p q$  и  $B = 2^n r$  — дружественные.

При  $n=2$  числа  $p=5$ ,  $q=11$  и  $r=71$  — простые, и получается пара чисел 220 и 284, найденная ещё Пифагором. Однако, теорема Сабита даёт дружественные числа и при других  $n$ , например:

$n=4$	$n=7$
A=17296	A=9363584
B=18416	B=9437056

В настоящее время известно, что этими тремя случаями исчерпываются все значения  $n \leq 20000$ , при которых указанный способ даёт дружественные числа.

14. Рассмотрим некоторое натуральное число  $n$ . Если это не палиндром, то изменим порядок его цифр на обратный и сложим исходное число с получившимся. Если сумма — не палиндром, то над ней повторяется то же действие и т.д., пока не получится палиндром. До настоящего времени *неизвестно*, завершается ли этот процесс для любого натурального  $n$ .

Даны натуральные числа  $k, l, m$  ( $k \leq l$ ). Проверить, верно ли, что для любого натурального числа из диапазона от  $k$  до  $l$  процесс завершается не позднее, чем после  $m$  таких действий.

**Замечание.** Назовём натуральное число *палиндромом*, если оно равно своему обращённому. *Обращённым* к числу  $N$  называется число, которое записано теми же цифрами, что и  $N$ , но в обратном порядке.

15. В качестве основания позиционной системы может быть взято *отрицательное* целое число. Например, можно рассмотреть систему с основанием  $-10$ . Любое целое  $N$  единственным образом представляется в виде суммы

$$a_s \times (-10)^s + a_{s-1} \times (-10)^{s-1} + \dots + a_1 \times (-10) + a_0, \quad 0 \leq a_i \leq 9, \quad i=0, \dots, s,$$

где

Запишите данное целое  $N$  в системе с основанием  $-10$ , т.е. найдите соответствующие числа  $a_s, a_{s-1}, \dots, a_0$ .

Наконец, отметим, что очень много интересных и сложных задач приведено в книге С.А. Абрамова, Г.Г. Гнездилова, Е.Н. Капустина, М.И. Селюна «Задачи по программированию». М.:Наука, 1988.

[http://sysadminmosaic.ru/msx/basic\\_programming\\_guide/12?rev=1584819930](http://sysadminmosaic.ru/msx/basic_programming_guide/12?rev=1584819930)

2020-03-21 22:45

