

10-11 класс. Вариант 1

Задача 1

Сколько различных решений имеет уравнение

$$((J \rightarrow K) \rightarrow (M \wedge N \wedge L)) \wedge ((M \wedge N \wedge L) \rightarrow (\neg J \vee K)) \wedge (M \rightarrow J) = 1$$

где J, K, L, M, N – логические переменные? В ответе не нужно перечислять все различные наборы значений J, K, L, M и N , при которых выполнено данное равенство. В качестве ответа Вам нужно указать количество таких наборов.

Решение (использование свойств импликации):

1) Перепишем уравнение, используя более простые обозначения операций:

$$[(J \rightarrow K) \rightarrow M \cdot N \cdot L] \cdot [M \cdot N \cdot L \rightarrow (\bar{J} + K)] \cdot (M \rightarrow J) = 1$$

2) Логическое произведение трех сомножителей равно единице, поэтому каждый из них должен быть тоже равен единице

3) Учитывая, что $J \rightarrow K = \bar{J} + K$, и выполняя замены $A = \bar{J} + K$ и $B = M \cdot N \cdot L$, получаем

$$(A \rightarrow B) \cdot (B \rightarrow A) \cdot (M \rightarrow J) = 1.$$

4) Рассмотрим последнюю импликацию, которая должна быть равна 1: $M \rightarrow J = 1$; по таблице истинности импликации сразу находим, что возможны три варианта:

а) $M = J = 0$

б) $M = 0, J = 1$

в) $M = J = 1$

5) Поскольку все (в том числе и первые две) импликации должны быть равны 1, по таблице истинности импликации сразу определяем, что $A = B$, то есть $\bar{J} + K = M \cdot N \cdot L$

6) В случае «а» последнее уравнение превращается в $1 + K = 0$ и не имеет решений

7) В случае «б» имеем $K = 0$, тогда как N и L – произвольные; поэтому есть 4 решения, соответствующие четырем комбинациям N и L

8) В случае «в» получаем $K = N \cdot L$, то есть для $K = 1$ есть единственное решение ($N = L = 1$), а для $K = 0$ – три решения (при $N = L = 0$; $N = 1$ и $L = 0$; $N = 0$ и $L = 1$)

9) Проверяем, что среди полученных решений, нет одинаковых

10) Таким образом, всего есть $4 + 1 + 3 = 8$ решений

Ответ – 8

Задача 2

Каким свойством(-ами) обладают все элементы приведенного массива, содержащего числа позиционной системы счисления? Ответ обоснуйте.

441 1134 1400 2124 2421 3231

Решение: Исходя из условия задачи, представлены числа позиционной системы счисления и так как в числах нет цифр, превышающих 4, предположим, что числа представлены в пятеричной системе счисления.

Переведем числа из пятеричной системы счисления в десятичную: 121 169 225 289 361 441, в десятичной последовательности можно заметить, что все цифры нечетные и являются квадратами 11^2 13^2 15^2 17^2 19^2 21^2 , следовательно, представленный ряд: Возрастающая последовательность, состоящая из квадратов нечетных чисел, записанных в пятеричной системе счисления.

Ответ: Возрастающая последовательность, состоящая из квадратов нечетных чисел (11^2 13^2 15^2 17^2 19^2 21^2) записанных в пятеричной системе счисления.

Задача 3

Существует некоторое число, XYZ, в десятичной записи которого любая пара цифр является простым числом. Например, 5979 такое четырехзначное число, так как числа 59, 97 и 79 являются простыми.

- Напишите эффективную программу, позволяющую получить количество способов составления N-значного числа XYZ, при $N=10$ и $N=15$;
- Ответьте на вопрос сколько чисел XYZ при $N=2$.

Входные данные. Входной файл INPUT.TXT содержит одно число N.

Выходные данные. В выходной файл OUTPUT.TXT вывести одно число – количество десятизначных и пятнадцатизначных чисел XYZ.

Пример.

INPUT.TXT	OUTPUT.TXT
10	48912
15	6404362

Ответ: 21 и приведённый листинг.

Решение:

Для решения данной задачи воспользуемся приемом динамического программирования.

1. Рассмотрим все двузначные простые числа: 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97. Их всего 21 число. Если продолжить цепочку длины 2 дальше, то следующие двузначные простые числа могут начинаться только со цифр 1, 3, 7 и 9. Введем массив a , где элемент $a[i]$ – количество вариантов формирования цепочки простых двузначных чисел, последняя цифра которого равна i . Для цепочки длиной $N = 2$ зададим начальные данные для a : $a[1] = 5$, $a[3] = 6$, $a[7] = 5$, $a[9] = 5$.

2. Воспользуемся вспомогательным массивом b , где элемент $b[i]$ – количество вариантов формирования цепочки простых двузначных чисел на одну цифру длиннее, последняя цифра которого также равна i .

3. Тогда $b[1] = a[1] + a[3] + a[7]$, $b[3] = a[1] + a[5] + a[7]$, $b[7] = a[1] + a[3] + a[9]$, $b[9] = a[1] + a[7]$.

4. В цикле по k , $k = 3, \dots, N$ вычислим значения $b[1]$, $b[3]$, $b[7]$, $b[9]$ и массиву a присвоим массив b . Ответом задачи будет сумма $a[1] + a[3] + a[7] + a[9]$.

```
program prostoprosto;
var a,b:array[1..9] of longint;
    i,j,n,k:integer;
    s:longint;
begin
    assign(input,'input.txt'); reset(input);
    assign(output,'output.txt'); rewrite(output);
    readln(n);
    FillChar(a,Sizeof(a),0); FillChar(b,Sizeof(b),0);
    s:=0;
{11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97}
    a[1]:=5;
    a[3]:=6;
```

```

a[7]:=5;
a[9]:=5;
for i:=3 to n do
  begin b[1]:=a[1]+a[3]+a[7];
        b[3]:=a[1]+a[7];
        b[7]:=a[1]+a[3]+a[9];
        b[9]:=a[1]+a[5]+a[7];
        a:=b
  end;
for i:=1 to 9 do s:=s+a[i];
writeln(s);
close(input); close(output);
end.

```

Задача 4

Для проведения геолого-технических мероприятий на фонде скважин предприятия необходимо произвести некоторые измерения. Значения этих измерений передаются в зашифрованном виде. В результате технического сбоя при передачи полученных измерений, некоторое количество данных не было передано.

Известно, что незашифрованные данные содержат четное количество десятичных цифр и для некоторых данных известны зашифрованные сообщения. Они приведены в таблице.

Незашифрованные данные	Зашифрованное сообщение
881665	2116 3306 2314 3116 2116
294653	2106 6346 2354 2154 2344
366511	7104 2144 2346 6104 6346
703109	2346 6114 2106 7156 6104
394775	3154 2316 6144 2114 2346

Зашифруйте оставшиеся данные: 566518 594083

Ответ обоснуйте.

Ответ: 566518 3304 2344 2346 6104 7306
594083 2144 6346 6344 2114 3306

Решение:

1) Из условия видно, что шифр 6-значного числа представляет 20-ти значное, такое возможно для связки десятичная и двоичная системы счисления, и восьмеричная и двоичная системы счисления, так как в зашифрованном сообщении нет цифр более 7-ми, очевидно восьмеричное представление числа.

2) Представим каждую цифру зашифрованного сообщения в двоичном виде, по 3 бита на цифру (зашифрованного сообщения) - 60 битов на 6 цифр (незашифрованного сообщения).

3) Запишем шифры в двоичном виде:

Код:

```
010001001110 011011000110 010011001100 011001001110 010001001110
010001000110 110011100110 010011101100 010001101100 010011100100
111001000100 010001100100 010011100110 110001000100 110011100110
010011100110 110001001100 010001000110 111001101110 110001000100
011001101100 010011001110 110001100100 010001001100 010011100110
```

4) Видно, что биты на нечетных позициях, начиная с младшего разряда - лишние, так как они все одинаковые и чередуются.

5) Уберем их, останутся 30 битов для 6 цифр (незашифрованного сообщения).

```
000011 011001 001010 010011 000011
000001 101101 001110 000110 001100
110000 001100 001101 100000 101101
001101 100010 000001 110111 100000
010110 001011 100100 000010 001101
```

6) Так как цифры по условию идут парами, в условии сказано, что их всегда четное количество, разобьем полученный двоичный код по 10 битов на пару цифр

Код:

```
88 16 65    0000110110 0100101001 0011000011
29 46 53    0000011011 0100111000 0110001100
```

36 65 11	1100000011 0000110110 0000101101
70 31 09	0011011000 1000000111 0111100000
39 47 75	0101100010 1110010000 0010001101

7) В нешифрованных сообщениях число 65 присутствует 2 раза (1-е число на третьей позиции и 3-е число на второй позиции) и шифруется 0000110110. Что означает, что числа шифруются в обратном порядке, т.е. первое число надо записать как 566188 и т.д

56 61 88	0000110110 0100101001 0011000011
35 64 92	0000011011 0100111000 0110001100
11 56 63	1100000011 0000110110 0000101101
90 13 07	0011011000 1000000111 0111100000
57 74 93	0101100010 1110010000 0010001101

8) В любой декаде ровно по 4 единиц и 6 нулей. Если посмотреть на них через разряд (1,3,5,7,9 и 2,4,6,8,10), увидим, что в этих пятерках ровно по 2 единицы и 3 нуля. Соответственно закодировано 10 комбинаций (10 цифр).

Разделим их:

56 61 88	00101 00110 00110 10001 01001 01001
35 64 92	00011 00101 00110 10100 01010 10010
11 56 63	10001 10001 00101 00110 00110 00011
90 13 07	01010 01100 10011 00001 01100 11000
57 74 93	00101 11000 11000 10100 01010 00011

9) Получим шифр каждой цифры:

0	01100
1	10001
2	10010
3	00011
4	10100
5	00101
6	00110
7	11000

8 01001

9 01010

10) Зашифруем необходимые 566518 594083, запишем цифры с помощью шифров в обратном порядке:

566518: 01001 10001 00101 00110 00110 00101

594083: 00011 01001 01100 10100 01010 00101

11) Объединим в декады, согласно разрядам. Добавим в нечетные разряды 1 и 0, начиная с 0 и младшего разряда:

566518:

011011000100010011100100010011100110110001000100111011000110

594083:

010001100100110011100110110011100100010001001100011011000110

12) Запишем числа в прямом порядке и шифр к ним:

566518 3304 2344 2346 6104 7306

594083 2144 6346 6344 2114 3306

Задача 5

На поле размером $N \times N$ клеток, клетки закрашиваются прямоугольниками $m \times n$, расположенными строго по горизонтали или вертикали. Закрашенные прямоугольники не соприкасаются между собой по горизонтали и вертикали. (Проверять правильность закрашенных прямоугольников не нужно.)

Напишите программу, которая получает на вход закрашенные на поле клетки, а на выходе получает средний размер прямоугольников (в клетках).

Входные данные:

N – длина стороны поля в клетках ($0 < N < 101$); далее построчно – обозначения клеток поля (0 – клетка не закрашена, 1 – клетка закрашена).

В ответе должно быть целое число: средний размер прямоугольника.

(Дробную часть числа отбрасывать.)

Пример решения:

Решение:

```
program p3;
var x:array[1..100,1..100]of integer;
i,j,n,k,s :integer; left,top:boolean;
begin
  {ввод поля}
  readln(n);
  for i:=1 to n do
    begin
      for j:=1 to n do
        read(x[i,j]);
      end;

s:=0; {количество 1}
k:=0; {количество клеток}
  for i:=1 to n do
    for j:=1 to n do
      if x[i,j]=1 then {если в клеточке часть}
        begin
          s:=s+1; {увеличиваем площадь}
          if (i=1) then {сверху - граница матрицы}
            left:=true
          else
            if x[i-1,j]=0 then {сверху нет}
              left :=true
            else
              left:=false;
          if (j=1) then {слева- граница матрицы}
            top:=true
          else
            if x[i,j-1]=0 then {слева нет}
              top:=true
            else
              top:=false;
          if left and top then {слева и сверху нет - это верхний
левый угол}
            k:=k+1;
          end;
          if k<>0 then s:=s div k;
          writeln(s); {подсчет среднего}
        end.
      end;
    end;
  end.
```

Задача 6

Во время проведения бухгалтерского отчета по предприятию, произошел сбой при передаче данных по затратам подразделений. Предприятие имеет 10 подразделений. Известно, что затраты каждого подразделения не превышают 500 млн. рублей и контрольная сумма SUM. Контрольная сумма удовлетворяет следующим условиям:

- 1) SUM — произведение затрат от двух подразделений.
- 2) SUM делится на 14.

Если контрольной суммы, удовлетворяющей условиям, нет, то значения не корректны и бухгалтерский отчет не сформирован.

Напишите эффективную программу, проверяющую корректность контрольной суммы.

Программа должна напечатать отчет по следующей форме:

Контрольная сумма: ...

Отчет сформирован. (или — Отчет не сформирован)

Кратко опишите используемый алгоритм решения.

На вход программе в каждой строке подаются затраты подразделений в млн. рублей (натуральные числа, не превышающее 500). В последней строке записана контрольная сумма.

Пример входных данных:

12
32
18
18
64
14
7
9
49
10
168

Пример выходных данных для приведенного выше примера входных данных:

Контрольная сумма: 168

Отчет сформирован.

Пояснение.

Произведение двух чисел делится на 14, если:

- один из сомножителей делится на 14 (второй может быть любым) либо
- ни один из сомножителей не делится на 14, причём один из сомножителей делится на 2, а другой - на 7.

Поэтому программа, вычисляющая кодовое число, может работать так.

Программа читает все входные данные один раз, не запоминая все данные в массиве. Программа для прочитанного фрагмента входной последовательности хранит значения четырех величин:

M2 — самое большое чётное число, не кратное 7;

M7 — самое большое число, кратное 7, но не кратное 2;

M14 — самое большое число, кратное 14;

MAX — самое большое число среди всех элементов последовательности, отличное от M14 (если число M14 встретилось более одного раза, и оно же является максимальным, то MAX = M14).

После того как все данные прочитаны, искомое кодовое слово вычисляется как максимум из произведений M14*MAX и M2*M7.

Ниже приведён пример программы на языке Паскаль, которая реализует описанный алгоритм.

Возможны и другие правильные алгоритмы. Допускаются решения, записанные на других языках программирования.

Пример правильной и эффективной программы на языке Паскаль

```
var M2,M7,M14,R,MAX,dat,res,i,N: longint;  
begin  
  N :=10;  
  M2 := 0;  
  M7 := 0;  
  M14 := 0;  
  MAX := 0;  
  for i := 1 to N do begin  
    readln(dat);
```

```
    if ((dat mod 2) = 0) and ((dat mod 7) > 0) and (dat > M2)
then M2 := dat;
    if ((dat mod 7) = 0) and ((dat mod 2) > 0) and (dat > M7)
then M7 := dat;
    if (dat mod 14 = 0) and (dat > M14) then begin
    if M14 > MAX then MAX := M14; M14 := dat
    end
    else
    if dat > MAX then MAX := dat;
    end;
    readln(R);
    if (M2*M7 < M14 *MAX) then
    res := M14 * MAX else
    res := M2*M7; writeln(' Контрольная сумма: ',res);
    if R = res
    then
    writeln('Отчет сформирован.')
    else
    writeln('Отчет не сформирован.');
```

end.