


Задания олимпиады
«Будущее Кузбасса. Информатика и ИКТ»
(задания представлены в порядке возрастания сложности
и, соответственно, баллов за их правильное решение)

Задание 1. Максимальная оценка – 5 баллов.

Дан фрагмент электронной таблицы.

	А	В	С	
1		7	12	
2	$=(A1-4)/(B1-2)$	$=(A1-4)/(C1-7)$	$=C1/(A1-3)$	

Какое целое число должно быть записано в ячейке А1, чтобы диаграмма, построенная по значениям ячеек диапазона А2:С2, соответствовала рисунку? Известно, что все значения ячеек из рассматриваемого диапазона неотрицательны.

Необходимо предоставить развёрнутый ответ.

Ответ: 9

Задание 2. Максимальная оценка – 5 баллов.

Запишите число, которое будет напечатано в результате выполнения следующей программы:

```
Pascal:
var n, s: integer;
begin
  n:=0;
  s:=0;
  while s < 25 do
  begin
    s:=s+5;
    n:=n+1;
  end;
  writeln(n);
end.
```

Необходимо предоставить развёрнутый ответ.

Решение:

Каждый повтор цикла от n будет увеличиваться на 1, то есть n — количество повторов цикла.

Для решения мы должны определить, сколько раз цикл будет повторён.

Цикл выполняется, пока $S < 25$, изначально $S = 0$, то есть цикл совершит последний повтор при $S=20$, увеличит S на 5, S станет равна 25 и цикл завершится.

Выходит, что количество повторов равно:

$$25:5 = 5$$

Программа выводит значение n, а n — это количество повторов. То есть $n=5$.

Ответ: 5

Задание 3. Максимальная оценка – 10 баллов.

Задана последовательность трехзначных натуральных чисел, длина последовательности неизвестна, а признаком её конца служит 0. Напечатать четные члены последовательности в убывающем порядке.

Пример решения:

```
Var q: Array [100..999] of integer;
a , i , j : Byte;
BEGIN
    for i:=100 to 999 do q[i]:=0; {обнуление счётчиков}
    WriteLn{"Вводите члены последовательности ' ' }; {ввод}
    Repeat
        Read(a);
        if a in [100..999] then Inc(q[a])
    Until a=0;
END.
for i:=999 downto 100 do
if i mod 2 =0 then
for j:=1 to q[i] do Write(i:4)
```

Необходимо написать программу на любом языке программирования для решения поставленной задачи. Перед программой укажите версию языка программирования.

Задание 4. Максимальная оценка – 10 баллов.

На обработку поступает положительное целое число, не превышающее 10^9 . Необходимо написать программу на любом языке программирования, которая выводит на экран количество разрядов числа, имеющих четные значения. Перед программой укажите версию языка программирования.

Пример решения:

Pascal:

```
var N, digit, sum: longint;
begin
    readln(N);
    sum := 0;
    while N > 0 do
        begin
            digit := N mod 10;
            if digit mod 2 = 0 then
                sum := sum + 1;
            N := N div 10;
        end;
    writeln(sum)
end.
```

Задание 5. Максимальная оценка – 10 баллов.

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 1000 включительно. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести количество элементов массива, значение которых двузначно и заканчивается на 3. Например, для массива из пяти элементов: 5 67 23 48 13 - ответ: 2. Перед программой укажите версию языка программирования.

Пример решения:

Pascal:

```
const
  n=30;
var
  a: array[1..n] of integer;
  i, j, t: integer;
begin
  for i:=1 to n do
    readln(a[i]);
  t := 0;
  for i:=1 to n do
    if (a[i] >= 10) and (a[i] <= 99) and (a[i] mod 10 = 3) then
      t := t+1;
  writeln(t);
end.
```

Задание 6. Максимальная оценка – 20 баллов.

От заданного прямоугольника каждый раз отрезается квадрат максимальной площади (длины сторон фигур выражаются натуральными числами). Найти число таких квадратов. Необходимо написать программу на любом языке программирования для решения поставленной задачи. Перед программой укажите версию языка программирования.

Пример решения:

Pascal:

```
Var a,b,c,q: LongInt;
BEGIN
WriteLn('Введите измерения прямоугольника от 1 до ',MaxLongInt);
ReadLn(a,b) ;
q:=0;
While b <> 0 do
begin { обмен }
  if a>b then begin c:=a; a:=b; b:=c end;
  q:=q + b div a;
  b:=b mod a
end;
Write('Всего квадратов : ',q)
END.
```

Задание 7. Максимальная оценка – 20 баллов.

Разложить натуральное число на простые сомножители. Необходимо написать программу на любом языке программирования для решения поставленной задачи. Перед программой укажите версию языка программирования.

Пример решения:

Pascal:

```
uses crt;
procedure PrsDel(a:longint);
var n:Longint;
    b:boolean;
begin
  n:=2;
  while a>0 do
    begin
```

```

b:=true;
while n <= round(sqrt(a)) do
begin
  if a mod n=0 then
  begin
    write(n, ' ');
    a:=a div n;
    b:=false;
    break;
  end;
  inc(n);
end;
if b then
begin
  write(a);
  break;
end;
end;
end;
var a:longint;
begin
clrscr;
write('a=');readln(a);
PrsDel(a);
readln
end.

```

Задание 8. Максимальная оценка – 25 баллов.

На спутнике «Восход» установлен прибор, предназначенный для измерения энергии космических лучей. Каждую минуту прибор передаёт по каналу связи неотрицательное целое число – количество энергии солнечного излучения, полученной за последнюю минуту, измеренное в условных единицах. Временем, в течение которого происходит передача, можно пренебречь.

Необходимо найти в заданной серии показаний прибора минимальное чётное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удаётся, ответ считается равным –1. Количество энергии, получаемое прибором за минуту, не превышает 1000 условных единиц. Общее количество показаний прибора в серии не превышает 10 000.

Входные данные представлены следующим образом. В первой строке задаётся число N – общее количество показаний прибора. Гарантируется, что $N > 6$. В каждой из следующих N строк задаётся одно неотрицательное целое число – очередное показание прибора.

Пример входных данных: 11; 12; 45; 5; 3; 17; 23; 21; 20; 19; 18; 17

Программа должна вывести одно число – описанное в условии произведение, либо – 1, если получить такое произведение не удаётся.

Пример выходных данных для приведённого выше примера входных данных: 54

Напишите на любом языке программирования программу для решения поставленной задачи. Для получения максимального результата (25 баллов) программа должна быть эффективна по времени и по используемой памяти.

Программа считается эффективной по времени, если время работы программы пропорционально количеству полученных показаний прибора N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

Решение:

Чтобы произведение было чётным, хотя бы один сомножитель должен быть чётным, поэтому при поиске подходящих произведений чётные показания прибора можно рассматривать в паре с любыми другими, а нечётные – только с чётными.

Для каждого показания с номером k , начиная с $k = 7$, рассмотрим все допустимые по условиям задачи пары, в которых данное показание получено вторым. Минимальное произведение из всех этих пар будет получено, если первым в паре будет взято минимальное подходящее показание среди всех, полученных от начала приёма и до показания с номером $k - 6$. Если очередное показание чётное, минимальное среди предыдущих может быть любым, если нечётное – только чётным.

Для получения эффективного по времени решения нужно по мере ввода данных помнить абсолютное минимальное и минимальное чётное показание на каждый момент времени, каждое вновь полученное показание умножать на соответствующий ему минимум, имевшийся на 6 элементов ранее, и выбрать минимальное из всех таких произведений.

Поскольку каждое текущее минимальное показание используется после ввода ещё 6 элементов и после этого становится ненужным, достаточно хранить только 6 последних минимумов. Для этого можно использовать массив из 6 элементов и циклически заполнять его по мере ввода данных.

Размер этого массива не зависит от общего количества введённых показаний, поэтому такое решение будет эффективным не только по времени, но и по памяти. Чтобы хранить абсолютный и чётный минимумы, нужно использовать два таких массива. Ниже приводится пример такой программы, написанной на алгоритмическом языке.

Программа 1. Пример правильной программы на алгоритмическом языке.
Программа эффективна по времени и по памяти

алг

нач

```
цел s = 6 | требуемое расстояние между показаниями
цел аmax = 1001 | больше максимально возможного показания
цел N
ввод N
цел a | очередное показание прибора
целтаб мини[0:s-1] | текущие минимумы последних s элементов
целтаб миничет[0:s-1] | чётные минимумы последних s
```

элементов

```
цел i
```

```
| вводим первые s показаний, фиксируем минимумы
```

```
цел ма; ма := аmax | минимальное показание
```

```
цел мчет; мчет := аmax | минимальное чётное показание
```

```
нц для i от 1 до s
```

```
    ввод a
```

```
    ма := imin(ма, a)
```

```
    если mod(a, 2) = 0 то мчет := imin(мчет, a) все
```

```
    мини[mod(i, s)] := ма
```

```
    миничет[mod(i, s)] := мчет
```

```
кц
```

```
цел mp = аmax*аmax | минимальное значение произведения
```

```
цел p
```

```
нц для i от s+1 до N
```

```
    ввод a
```

```

если mod(a,2)=0
    то п := а * мини[mod(i, s)]
    иначе если мчет < аmax
        то п := а * миничет[mod(i, s)]
        иначе п := аmax*аmax;
    все
все
мп := imin(мп, п)
ма := imin(ма, а)
если mod(a,2) = 0 то мчет := imin(мчет,а) все
мини[mod(i, s)] := ма
миничет[mod(i, s)] := мчет
кц
если мп = аmax*аmax то мп:=-1 все
вывод мп
конт

```

Возможны и другие реализации. Например, вместо циклического заполнения массива можно каждый раз сдвигать его элементы. В приведённом ниже примере хранятся и сдвигаются не минимумы, а исходные значения. Это требует чуть меньше памяти (достаточно одного массива вместо двух), но по времени решение со сдвигами менее эффективно, чем с циклическим заполнением. Однако время работы остаётся пропорциональным N .

Программа 2. Пример правильной программы на языке Паскаль. Программа использует сдвиги, но эффективна по времени и по памяти

```

const s = 6; {требуемое расстояние между показаниями}
      аmax = 1001; {больше максимально возможного показания}
var
  N: integer;
  а: array[1..s] of integer; {хранение s показаний прибора}
  а_: integer; {ввод очередного показания}
  ма: integer; {минимальное число без s последних}
  ме: integer; {минимальное чётное число без s последних}
  мп: integer; {минимальное значение произведения}
  р: integer;
  i, j: integer;
begin
  readln(N);
  {Ввод первых s чисел}
  for i:=1 to s do readln(a[i]);
  {Ввод остальных значений, поиск минимального произведения}
  ма := аmax; ме := аmax;
  мп :=аmax*аmax;
  for i := s + 1 to N do begin
    readln(а_);
    if а[1] < ма then ма := а[1];
    if (а[1] mod 2 = 0) and (а[1] < ме) then ме := а[1];
    if а_ mod 2 = 0 then п := а_ * ма
    else if ме < аmax then п := а_ * ме
    else п := аmax* аmax;
    if (п < мп) then мп := п;
    {сдвигаем элементы вспомогательного массива влево}
    for j := 1 to s - 1 do

```

```

        a[j] := a[j + 1];
    a[s] := a_
end;
if mp = amax*amax then mp:=-1;
writeln(mp)
end.

```

Если вместо небольшого массива фиксированного размера (циклического или со сдвигами) хранятся все исходные данные (или все текущие минимумы), программа сохраняет эффективность по времени, но становится неэффективной по памяти, так как требуемая память растёт пропорционально N . Ниже приводится пример такой программы на языке Паскаль. Подобные (и аналогичные по сути) программы оцениваются не выше 20 баллов.

Программа 3. Пример правильной программы на языке Паскаль. Программа эффективна по времени, но неэффективна по памяти

```

const s = 6; {требуемое расстояние между показаниями}
      amax = 1001; {больше максимально возможного показания}
var
  N, p, i: integer;
  a: array[1..10000] of integer; {все показания прибора}
  ma: integer; {минимальное число без s последних}
  me: integer; {минимальное чётное число без s последних}
  mp: integer; {минимальное значение произведения}
begin
  readln(N);
  {Ввод всех показаний прибора}
  for i:=1 to N do readln(a[i]);
  ma := amax;
  me := amax;
  mp := amax*amax;
  for i := s + 1 to N do
  begin
    if a[i-s] < ma then ma := a[i-s];
    if (a[i-s] mod 2 = 0) and (a[i-s] < me) then
      me := a[i-s];
    if a[i] mod 2 = 0 then p := a[i] * ma
    else if me < amax then p := a[i] * me
    else p := amax * amax;
    if (p < mp) then mp := p
  end;
  if mp = amax*amax then mp := -1;
  writeln(mp)
end.

```

Возможно также переборное решение, в котором находятся произведения всех возможных пар и из них выбирается минимальное. Это (и аналогичные ему) решение неэффективно ни по времени, ни по памяти. Оценка за такое решение – 15 баллов.

Программа 4. Пример правильной программы на языке Паскаль. Программа неэффективна ни по времени, ни по памяти

```

const s = 6; {требуемое расстояние между показаниями}
var
  N: integer;
  a: array[1..10000] of integer; {все показания прибора}

```

```
mp: integer; {минимальное значение произведения}
i, j: integer;
begin
  readln(N);
  {Ввод значений прибора}
  for i:=1 to N do
    readln(a[i]);
  mp := 1000 * 1000 + 1;
  for i := 1 to N-s do begin
    for j := i+s to N do begin
      if (a[i]*a[j] mod 2 = 0) and (a[i]*a[j] < mp)
        then mp := a[i]*a[j]
    end;
  end;
  if mp = 1000 * 1000 + 1 then mp := -1;
  writeln(mp)
end.
```