

C4 (ege27)

55 МИН

Задание.

Имеется набор данных, состоящий из 6 пар положительных целых чисел. **Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 4 и при этом была максимально возможной.** Если получить требуемую сумму невозможно, в качестве ответа нужно выдать 0. В этом варианте задания оценивается только правильность программы, время работы и размер использованной памяти не имеют значения.

Для варианта А на вход программе подаётся 6 строк, каждая из которых содержит два натуральных числа, не превышающих 10000.

Пример входных данных для варианта А:

```
6
1 3
5 12
6 8
5 4
3 3
1 1
```

Пример выходных данных для приведённых выше примеров входных данных:

```
31
```

Решение:

На вход программе в первой строке подаётся количество пар N ($1 \leq N \leq 100000$). Каждая из следующих N строк содержит два натуральных числа, не превышающих 10 000.

```

var N, i, x1, x2, sum: integer;
    d, delta: integer;
begin
  Readln(N);
  sum := 0;
  delta := 10001;
  delta := 10001;
  {числами}
  for i:=1 to N do
  begin
    readln(x1, x2);
    if x1 > x2 then
      sum := sum + x1
    else sum := sum + x2;
    d := abs(x1-x2);
    if (d mod 4 <> 0) and (d < delta) then
      delta := d;
    end;
    if sum mod 4 = 0 then
      if delta = 10001 then
        sum := 0
      else sum := sum - delta;
    writeln(sum);
  end.

```

- {Первоначальная разница между двумя вводимыми числами}
- Чтобы сумма осталась максимально возможной, для замены нужно выбрать такую пару из входных данных, в которой разница между двумя числами минимальная. Ищем пару чисел, для которой выполняются два условия:
 - Если сумма **sum**, полученная в результате работы цикла, не делится на 4, следует просто вывести эту сумму.
 - Если делится, нужно определить, есть ли возможность замены.
 - Если такой возможности нет, то условный оператор ни разу не сработал, и в переменной **delta** осталось начальное значение 100001.
 - Если замена возможна, в результате такой замены сумму нужно уменьшить на разность между двумя числами в паре, в которой эта замена делается,

ЕГЭ (Октябрь)

Дан набор из N целых положительных чисел. Необходимо выбрать из набора произвольное количество чисел так, чтобы их **сумма была как можно больше и при этом не делилась на 6**. В ответе нужно указать количество выбранных чисел и их сумму, сами числа выводить не надо. Если получить нужную сумму невозможно, считается, что выбрано 0 чисел и их сумма равна 0. Напишите эффективную по времени и по памяти программу для решения этой задачи.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 1000$).

В каждой из последующих N строк записано одно натуральное число, не превышающее 10000.

Пример входных данных:

3
1
2
3

В результате работы программа должна вывести два числа: сначала количество выбранных чисел, затем их сумму.

Пример выходных данных для приведённого выше примера входных данных:

2 5

В данном случае из предложенного набора нужно выбрать два числа (2 и 3), их сумма равна 5.

Алгоритм:

Программа должна прочитать все числа, не сохраняя их, подсчитать общую сумму и определить наименьшее число, не кратное 6, а далее действовать по описанным правилам.

- Если сумма всех данных чисел не кратна 6, нужно просто взять все числа.
- Если сумма кратна 6, нужно удалить из неё минимально возможный элемент – наименьшее из заданных чисел, не кратное 6.
- Если таких чисел нет (все числа в наборе кратны 6), то получить требуемую сумму невозможно, в этом случае по условию задачи ответ считается равным нулю.

```

const  d=6;      {делитель}
amax = 10000;  {максимально возможное число}
var    N: integer; {количество чисел}
      a: integer; {очередное число}
      s: integer; {сумма}
      mn: integer; {минимальное число, не кратное d}
      k: integer; {количество выбранных чисел}
      i: integer;

begin
  readln(N);
  s := 0;
  mn := amax+1;
  for i:=1 to N do
    begin
      readln(a);
      s := s+a;
      if (a mod d <> 0) and (a < mn) then mn := a;
    end;
  if s mod d <> 0 then k := N
Else  if mn <= amax then
    begin
      k := N-1;
      s := s - mn;
    end
  else
    begin
      k := 0;
      s := 0;
    end;
  writeln(k, ' ', s);
end.

```

```
var i,s6,s,min,n,k:longint;  
begin  
  s6:=0; s:=0;  
  readln(n);  
  min:=10001;  
  for i:=1 to n do  
  begin  
    readln(k);  
    if k mod 6 = 0 then s6:=s6+k  
    else  
      begin  
        if k<min then min:=k;  
        s:=s+k;  
      end;  
    end;  
    if s mod 6 =0 then writeln((n-1),'',(s+s6-min))  
    else writeln(n,'',(s+s6));  
end.
```

Дан набор из N целых положительных чисел. Необходимо выбрать из набора произвольное количество чисел так, чтобы их сумма была как можно больше и при этом не делилась на 8. В ответе нужно указать количество выбранных чисел и их сумму, сами числа выводить не надо. Если получить нужную сумму невозможно, считается, что выбрано 0 чисел и их сумма равна 0.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leq N \leq 1000$). В каждой из последующих N строк записано одно натуральное число, не превышающее 10 000.

Пример входных данных:

3
1
2
5

В результате работы программа должна вывести два числа: сначала – количество выбранных чисел, затем их сумму.

Пример выходных данных для приведённого выше примера входных данных:

2 7

В данном случае из предложенного набора нужно выбрать два числа (2 и 5), их сумма равна 7.

Содержание верного ответа

- Если сумма всех данных чисел не кратна 8, нужно просто взять все числа.
- Если сумма кратна 8, нужно удалить из неё минимально возможный элемент – наименьшее из заданных чисел, не кратное 8.
- Если таких чисел нет (все числа в наборе кратны 8), то получить требуемую сумму невозможно, в этом случае по условию задачи ответ считается равным нулю.

Программа должна прочитать все числа, не сохраняя их, подсчитать общую сумму и определить наименьшее число, не кратное 8.

```

const  d=8;      {делитель}
amax = 10000;  {максимально возможное число}
var    N: integer; {количество чисел}
      a: integer; {очередное число}
      s: integer; {сумма}
      mn: integer; {минимальное число, не кратное d}
      k: integer; {количество выбранных чисел}
      i: integer;

begin
  readln(N);
  s := 0;
  mn := amax+1;
  for i:=1 to N do
    begin
      readln(a);    s := s+a;
      if (a mod d <> 0) and (a < mn) then mn := a;
    end;
  if s mod d <> 0 then k := N
  else
    if mn <= amax then
      begin
        k := N-1;    s := s - mn;
      end
    else
      begin
        k := 0;    s := 0;
      end;
  writeln(k, ' ', s);
end.

```

Программа

Последовательность натуральных чисел характеризуется числом Y – наибольшим числом, кратным 26 и являющимся произведением двух элементов последовательности с различными номерами.

Напишите эффективную, в том числе по используемой памяти, программу, находящую число Y для последовательности натуральных чисел, значение каждого элемента которой не превосходит 1000.

Программа должна напечатать найденное число, если оно существует для заданной последовательности, или ноль в противном случае.

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

На вход программе в первой строке подаётся количество чисел N . В каждой из последующих N строк записано одно натуральное число, не превышающее 1000.

Пример входных данных:

5
40
100
130
28
51

Пример выходных данных для приведённого выше примера входных данных:

13000

Содержание верного ответа:

Произведение двух чисел делится на 26, если:

- один из сомножителей делится на 26 (второй может быть любым) либо
- ни один из сомножителей не делится на 26, но один из сомножителей делится на 13, а другой – на 2.

Программа, вычисляющая число Y , может работать так.

Программа читает все входные данные один раз, не запоминая все данные в массиве.

Программа для прочитанного фрагмента входной последовательности хранит значения четырёх величин:

M13 – самое большое число, кратное 13, но не кратное 2;

M2 – самое большое число, кратное 2, но не кратное 13;

M26 – самое большое число, кратное 26;

MAX – самое большое число среди всех элементов последовательности, отличное от M26 (если число M26 встретилось более одного раза и оно же является максимальным, то $MAX = M26$).

После того как все данные прочитаны, искомое число Y вычисляется как максимум из произведений $M26 * MAX$ и $M13 * M2$.

```

var M13,M2,M26,MAX,dat,res,i,N: longint;
begin
  M13 := 0; {самое большое число, кратное 13, но не кратное 2}
  M2 := 0;  {самое большое число, кратное 2, но не кратное 13}
  M26 := 0; {самое большое число, кратное 26; }
  MAX := 0; {самое большое число среди всех элементов
             последовательности,           отличное от M26 }
  readln(N);
  for i := 1 to N do
    begin
      readln(dat);
      if ((dat mod 13) = 0) and ((dat mod 2) > 0) and (dat > M13) then M13 := dat;
      if ((dat mod 2) = 0) and ((dat mod 13) > 0) and (dat > M2) then M2 := dat;
      if (dat mod 26 = 0) and (dat > M26) then
        begin
          if M26 > MAX then MAX := M26;
          M26 := dat
        end
      else
        if dat > MAX then    MAX := dat;
      end;
      if (M13*M2 < M26*MAX) then  res := M26*MAX
      else  res := M13*M2;
    writeln(res);
  end.

```

Последовательность натуральных чисел характеризуется числом X – наибольшим числом, кратным 14 и являющимся произведением двух элементов последовательности с различными номерами.

Напишите эффективную, в том числе по используемой памяти, программу, находящую число X для последовательности натуральных чисел, значение каждого элемента которой не превосходит 1000.

Программа должна напечатать найденное число, если оно существует для заданной последовательности, или ноль в противном случае.

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

На вход программе в первой строке подаётся количество чисел N .

В каждой из последующих N строк записано одно натуральное число, не превышающее 1000.

Пример входных данных:

5
40
1000
7
28
55

Пример выходных данных для приведённого выше примера входных данных:

28000

Содержание верного ответа

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, но один из сомножителей делится на 7, а другой – на 2.

Программа читает все входные данные один раз, не запоминая все данные в массиве.

Программа для прочитанного фрагмента входной последовательности хранит значения четырёх величин:

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

M2 – самое большое число, кратное 2, но не кратное 7;

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 число X вычисляется как максимум из произведений $M14 * MAX$ и $M7 * M2$.

```

var M7,M2,M14,MAX,dat,res,i,N: longint;
begin
  M7 := 0; M2 := 0; M14 := 0; MAX := 0;
  readln(N);
  for i := 1 to N do
    begin
      readln(dat);
      if ((dat mod 7) = 0) and ((dat mod 2) > 0) and (dat > M7) then M7 := dat;
      if ((dat mod 2) = 0) and ((dat mod 7) > 0) and (dat > M2) then M2 := 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;
    if (M7*M2 < M14*MAX) then res := M14*MAX
    else res := M7*M2;
  writeln(res);
end.

```

На плоскости задано множество точек с целочисленными координатами.

Необходимо найти максимально возможную площадь невырожденного (то есть, имеющего ненулевую площадь) треугольника, одна вершина которого расположена в начале координат, а две другие лежат на осях координат и при этом принадлежат заданному множеству.

Если такого треугольника не существует, необходимо вывести соответствующее сообщение.

Входные данные

В первой строке задаётся N – количество точек в заданном множестве.

Каждая из следующих строк содержит два целых числа – координаты очередной точки.

Пример входных данных:

```
3
6 0
0 8
9 7
```

Выходные данные

Если искомый треугольник существует, программа должна напечатать одно число: максимально возможную площадь треугольника, удовлетворяющего условиям.

Если искомый треугольник не существует, программа должна напечатать сообщение: «Треугольник не существует».

Пример выходных данных для приведённого выше примера входных данных:

```
24
```

Содержание верного ответа

Вершины невырожденного треугольника должны лежать на разных осях, их координаты должны иметь вид $(x, 0)$ и $(0, y)$.

Площадь такого треугольника равна $|x| \cdot |y| / 2$.

Площадь будет максимальной при максимальных значениях $|x|$ и $|y|$.

```
var    N: integer; {количество точек}
x,y: integer; {координаты очередной точки}
xmax, ymax: integer;
s: real; {площадь}
i: integer;
begin
    readln(N);
    xmax:=0;
    ymax:=0;
    for i:=1 to N do
        begin
            readln(x,y);
            if (x=0) and (abs(y)>ymax) then ymax:=abs(y);
            if (y=0) and (abs(x)>xmax) then xmax:=abs(x);
        end;
    s:=xmax*ymax/2;
    if (s=0) then      writeln('Треугольник не существует')
    else writeln(s)
end.
```

На плоскости задано множество точек с целочисленными координатами. Необходимо найти количество отрезков, обладающих следующими свойствами:

- 1) оба конца отрезка принадлежат заданному множеству;
- 2) ни один конец отрезка не лежит на осях координат;
- 3) отрезок пересекается ровно с одной осью координат.

Входные данные

В первой строке задаётся N – количество точек в заданном множестве.

Каждая из следующих строк содержит два целых числа x и y – координаты очередной точки. Гарантируется, что $1 \leq N \leq 10\,000$; $-1000 \leq x, y \leq 1000$.

Пример входных данных:

4

6 6

-8 8

-9 -9

7 -5

Выходные данные

Необходимо вывести единственное число: количество удовлетворяющих требованиям отрезков. Пример выходных данных для приведённого выше примера входных данных:

4

Содержание верного ответа

Отрезок, концы которого не лежат на осях координат, пересекается ровно с одной осью в том случае, если его концы лежат в соседних четвертях.

Если известны величины n_1, n_2, n_3, n_4 , показывающие количество точек в каждой четверти, то количество отрезков равно

$$n_1n_2 + n_2n_3 + n_3n_4 + n_4n_1.$$

Упростив это выражение, получим

$$(n_1 + n_3)(n_2 + n_4).$$

Это выражение можно получить и непосредственно из условий задачи

- у каждого подходящего отрезка один конец лежит в нечётной четверти, а другой – в чётной и
- для определения количества отрезков точки считаются не в каждой четверти отдельно, а их общее количество в чётных и нечётных четвертях.

Пример правильной и эффективной программы на языке Паскаль (подсчёт по чётным и нечётным четвертям)

```
program C427;
var   N: integer;           {количество точек}
      x,y: integer;        {координаты очередной точки}
      n1, n2: integer;     {количество точек по чётным и нечётным четвертям}
      s: integer;         {количество отрезков}
      i: integer;
begin
  readln(N);
  n1:=0; n2:=0;
  for i:=1 to N do
  begin
    readln(x,y);
    if x*y > 0 then n1:=n1+1;
    if x*y < 0 then n2:=n2+1;   {нельзя ставить else – возможны нули}
  end;
  s := n1*n2;
  writeln(s)
end.
```

На плоскости задано множество точек с целочисленными координатами, никакие две из которых не совпадают и никакие три не лежат на одной прямой. Необходимо найти количество треугольников, обладающих следующими свойствами:

- 1) все вершины треугольника принадлежат заданному множеству;
- 2) ни одна вершина не лежит на осях координат;
- 3) треугольник не пересекается с осью Ox , но пересекается с осью Oy .

Входные данные

В первой строке задаётся N – количество точек в заданном множестве. Каждая из следующих строк содержит два целых числа x и y – координаты очередной точки.

Гарантируется, что $1 \leq N \leq 10000$, $-1000 \leq x, y \leq 1000$, никакие две точки не совпадают, никакие три не лежат на одной прямой.

Пример входных данных:

```
4
6 6
-8 8
-9 -9
7 5
```

Выходные данные Необходимо вывести единственное число: количество удовлетворяющих требованиям треугольников.

Пример выходных данных для приведённого выше примера входных данных:

```
1
```

Содержание верного ответа

Чтобы треугольник не пересекался с осью Ox и пересекался с осью Oy , его вершины должны лежать в одной полуплоскости относительно Ox и в разных относительно Oy .

Вершины треугольника должны лежать в первой и второй либо в третьей и четвёртой четвертях, причём в одной из этих четвертей должны лежать две вершины, в другой – одна.

Зная количество точек в каждой четверти, можно подсчитать количество искомых треугольников.

Например

Если в первой четверти лежит n_1 точек, а во второй – n_2 точек, то количество треугольников, у которых две вершины лежат в первой четверти, а одна – во второй, равно

$$(n_1(n_1-1)/2) * n_2 = n_1(n_1-1)n_2/2.$$

Если известны величины n_1 , n_2 , n_3 , n_4 , показывающие количество точек в каждой четверти, то общее количество треугольников равно

$$(n_1(n_1-1)n_2 + n_2(n_2-1)n_1 + n_3(n_3-1)n_4 + n_4(n_4-1)n_3) / 2.$$

```

program P27;
var  N: integer;           {количество точек}
     x,y: integer;       {координаты очередной точки}
     n1, n2, n3, n4: integer; {количество точек по четвертям}
     s: integer;         {количество треугольников}
     i: integer;
begin
  readln(N);
  n1:=0; n2:=0; n3:=0; n4:=0;
  for i:=1 to N do
  begin
    readln(x,y);
    if (x>0) and (y>0) then n1 := n1+1;
    if (x<0) and (y>0) then n2 := n2+1;
    if (x<0) and (y<0) then n3 := n3+1;
    if (x>0) and (y<0) then n4 := n4+1;
  end;
  s := (n1*n2*(n1+n2-2) + n3*n4*(n3+n4-2)) div 2;
writeln(s)
end.

```

По каналу связи передается последовательность положительных целых чисел X_1, X_2, \dots все числа не превышают 1000, их количество заранее неизвестно.

Каждое число передается в виде отдельной текстовой строки, содержащей десятичную запись числа. Признаком конца передаваемой последовательности является число 0.

Участок последовательности от элемента X_T , до элемента X_{T+N} называется подъемом, если на этом участке каждое следующее число больше предыдущего. Высотой подъема называется разность $X_{T+N} - X_T$.

Напишите эффективную программу, которая вычисляет наибольшую высоту среди всех подъемов последовательности. Если в последовательности нет ни одного подъема, программа выдает 0.

Программа должна напечатать отчет по следующей форме:

Пример входных данных:

144

17

27

3

7

9

11

10

0

Пример выходных данных для приведенного выше примера входных данных:

Получено 8 чисел

Наибольшая высота подъема: 10

Программа читает все входные данные один раз, не запоминая все входные данные в массиве.

Во время чтения программа помнит:

- число **LMax** – высоту самого высокого из уже закончившихся подъемов
- необходимые сведения о текущем подъеме, например, число **L** – высоту текущего подъема (то есть разность между последним и первым числом участка)
- и последнее прочитанное число **T** (число – наибольшее из чисел текущего подъема).

Прочитав очередное число **R**, программа сравнивает его с числом **T**.

Если $R > T$, то значение **L** увеличивается на $R - T$.

В противном случае фиксируется конец подъема и начало нового участка.

То есть, значение **L** сравнивается с **LMax** и, при необходимости, **LMax** полагается равным **L**. В противном случае, полагаем $L = 0$

```
var R, T, N, L, LMax : Integer;
begin
  N:=0;
  L:=0;
  LMax:=0;
  T:=1001;
  repeat
    ReadLn(R);
    if R<>0 then N:=N+1;
    if R>T then L:=L+R-T
    else
      begin
        if L>LMax then LMax:=L;
        L:=0;
      end;
      T:=R;
  until R=0;
  WriteLn('Получено ', N, ' чисел');
  WriteLn('Наибольшая высота подъема ', LMax);
end.
```