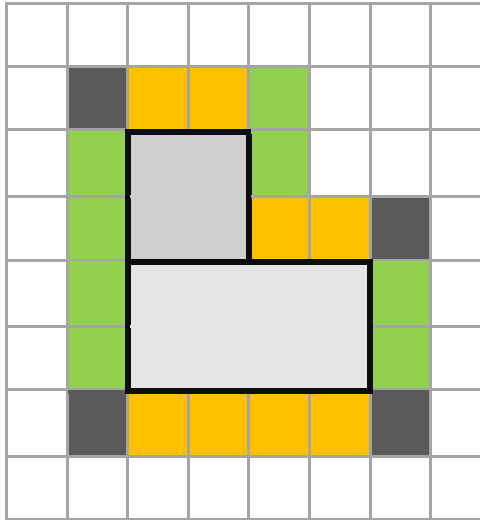


**Разбор заданий муниципального этапа
Всероссийской олимпиады школьников по информатике
2019-2020 учебный год
7-8 классы**

Задача А. Грядки

Рассмотрим на рисунке, как образуется обрамление грядки



Длина обрамления складывается из:

- горизонтальных границ (на рисунке отмечены желтым), общая длина которых равна удвоенной большей ширине;
- вертикальных границ (на рисунке – зеленые), общая длина которых равна удвоенной сумме высот;
- четырех угловых клеток (серые).

Тогда общее количество клеток находится как

$$2 \cdot \max(w1, w2) + 2 \cdot (h1 + h2) + 4$$

Пример программы на языке Pascal:

```
program task_A;  
  var w1, w2, h1, h2, m: longint;  
begin  
  read(w1, h1, w2, h2);  
  if w1 > w2 then  
    m := w1  
  else  
    m := w2;  
  writeln(2 * m + 2 * (h1 + h2) + 4) ;  
end.
```

Задача В. Авиаперелеты

Для решения задачи требуется вычислить величину скидки в каждом аэропорт и вывести наименование аэропорта, предоставляющего наибольшую скидку. Если несколько аэропортов дают одинаково большую скидку, то выбрать тот, который по условию ближе. Это можно реализовать, проверяя на минимальность в первую очередь наиболее близкий аэропорт, а если он не подходит, то проверять на минимальность наиболее предпочтительный из двух оставшихся.

Пример программы на языке Pascal:

```
program task_B;
var S, svo, dme: integer; vko : real;
begin
  read(S);
  svo := (S div 100) * 5;
  dme := (S div 50) * 2;
  vko := 0.03 * S;
  if (vko >= svo) and (vko >= dme) then
    write('VKO')
  else
    if (dme >= svo) then
      write('DME')
    else
      write('SVO');
  end.
```

Задача С. Кладоискатели

Смоделируем перемещение Димы.

Оно аналогично движению на координатной плоскости. Направления 1 и 3 соответствуют движению вдоль оси Oy соответственно вверх и вниз, а направления 2 и 4 – по оси Ox (влево и вправо). Вычислим перемещение по каждой оси. Тогда наименьшее количество шагов Димы – сумма этих значений.

Пример программы на языке Pascal:

```
program task_C;  
  var x, y, n, l: integer;  
  begin  
    x := 0;  
    y := 0;  
    repeat  
      read(n);  
      if n <> 0 then read(l);  
      if n = 1 then y := y + l;  
      if n = 2 then x := x - l;  
      if n = 3 then y := y - l;  
      if n = 4 then x := x + l;  
    until n = 0;  
    write(abs(x) + abs(y));  
  end.
```

Задача D. Всё наоборот

Рассмотрим последовательность. Очевидно, что если третий элемент получается как сумма первого и второго, то, зная третий и второй элементы, первый вычисляется как их разность. В общем случае $a_{i-1} = a_{i+1} - a_i$.

Зная два последних элемента, будем вычислять подобным образом предшествующие им элементы. По условию нам даны элементы с номерами n и $n + 1$, значит нужно восстановить предыдущие $n - 1$ элемент и вывести на экран первый и второй.

Пример программы на языке Pascal:

```
program task_D;
var a1, a2, n, d, i: longint;
begin
  read (n);
  read (a1, a2);
  for i := 1 to n - 1 do
    begin
      d := a2 - a1;
      a2 := a1;
      a1 := d;
    end;
  write (a1, ' ', a2);
end.
```