

Задача А. Хали-хало

Исходя из условия можно вычислить, что:

1 лилипутский шаг равен 3 муравьиным

1 обычный шаг – 2 лилипутских – 6 муравьиных

1 гигантский шаг – 3 обычных – 6 лилипутских – 18 муравьиных

Выделение частей из целого можно организовать с помощью функции взятия остатка и неполного частного.

Пример программы на языке Pascal

```
program game;  
var giant, normal, lilliputian, ant: integer;  
begin  
  readln(ant);  
  giant := ant div 18;  
  normal := (ant mod 18) div 6;  
  lilliputian := ant mod 6 div 3;  
  ant := ant mod 3;  
  write(giant, ' ', normal, ' ', lilliputian, ' ', ant);  
end.
```

Задача В. Треугольники

Учитывая, что хотя бы одна вершина должна быть отмеченной точкой и никакие три точки не лежат на одной прямой, получим, что количество треугольников будет минимальным, если в каждой из трех вершин треугольника находится отмеченная точка (т.е. количество треугольников в три раза меньше точек).

Исключением является ситуация, когда количество точек не кратно трем. В этом случае нужно количество точек разделить на три и округлить полученное число в большую сторону. Это можно сделать, используя стандартный прием или с помощью команды ветвления.

Пример программы на языке Pascal

```
program triangle;  
var points: integer;  
begin  
  read(points);  
  write((points + 2) div 3);  
end.
```

Задача С. Шпаргалка для пароля

Известно, что пароль нечетный делится на 15. Значит, он, во-первых, должен заканчиваться цифрой 5, во-вторых, делиться на 3 (т.е. сумма цифр кратна 3).

Значит, чтобы найти минимальное количество цифр, приписанных в конце, нужно подсчитать все цифры, после последней встреченной цифры 5.

Проверить, была ли приписана еще одна цифра где-то в середине числа, можно после вычисления суммы всех цифр до этой последней пятерки (включая её саму): если сумма цифр не кратна трем, значит, в пароль была дописана еще одна лишняя цифра, о которой мог забыть Миша.

Пример программы на языке Pascal

```
program password;
var n, i, digit, count, sum, s: integer;
    last5 :boolean;
begin
  read(n);
  sum := 0;
  s := 0;
  count := 0;
  last5 := false;
  for i := 1 to n do
    begin
      read(digit);
      s := s + digit;
      if last5 then
        count := count + 1;
      if digit = 5 then
        begin
          last5 := true;
          count := 0;
          sum := sum + s;
          s := 0;
        end;
    end;
  if sum mod 3 <> 0 then
    count := count + 1;
  write(count);
end.
```

Задача D. Очередь в поликлинике

Пациенты могут говорить правду, если количество человек в его ответе (то есть те, кто идут перед ним, и те, кто идут после него), будет на 1 меньше общего количества человек в очереди (т.е. все, не считая его самого).

Значит, нужно подсчитать, для скольких пациентов в очереди выполняется требование, что

$$\text{количество_человек_до_него} + \text{количество_человек_после_него} + 1 = n.$$

Однако следует также учитывать, что если несколько человек назовут одно и то же, значит, один из них говорит неправду (не может быть двоих человек, у которых одинаковое количество тех, кто перед ними и после них).

Для того, чтобы отслеживать повторяющиеся ответы, заведем массив `patient` из N логических значений, где каждое `patient [i]` будет показывать, мог ли больной, стоящий в очереди под номером i говорить правду. Если же встретится кто-то, кто говорит те же значения, что были названы ранее, то повторно этот ответ учитывать не будем.

Пример программы на языке Pascal

```
var patient: array[1..10000] of boolean;
i, x, y, n, count: integer;
begin
  readln(n);
  count := 0;
  for i := 1 to n do
    patient[i] := false;
  for i := 1 to n do
    begin
      read(x);
      read(y);
      if (x + y + 1 = n) then
        patient[x + 1] := true;
    end;
  for i := 1 to n do
    if patient[i] then
      count := count + 1;
  writeln(count);
end.
```