

ограничение по времени на тест	1 second
ограничение по памяти на тест	64 megabytes
ввод	стандартный ввод
вывод	стандартный вывод

## 1. Количество максимумов

Дано 3 натуральных числа. Посчитать, сколько раз среди них повторилось максимальное из этих чисел.

### Входные данные

В единственной строке входных данных записаны натуральные числа  $a, b, c$ , не превышающие 100.

### Выходные данные

В единственной строке вывести количество максимальных элементов.

### Примеры

#### входные данные

```
5 3 5
```

#### выходные данные

```
2
```

### Идея решения

Задачу можно решать через нахождение максимального числа из трёх (сложные условия, вложенные ветвления или введение переменной для хранения максимума). Далее можно было просто сравнить максимум со всеми введёнными числами, либо на этапе нахождения максимума запоминать новое значение количества максимальных элементов.

### Пример программы решения без сложных условий и циклов:

```
Program z1;  
var a, b, c, max, k : longint;  
begin  
  read(a, b, c);  
  max := a; k := 1;  
  if b > max  
  then begin max := b; k := 1; end  
  else if b = max then k := k + 1;  
  if c > max  
  then begin max := c; k := 1; end  
  else if c = max then k := k + 1;  
  write(k);  
end.
```

## 2. Волчья доля

Три поросёнка собирали яблоки. Первый поросёнок собрал  $a$  яблок, второй –  $b$ , третий –  $c$ . Они хотят урожай разделить поровну, не разрезая яблок. Кроме того, часть яблок (не меньше половины) придётся отдать волку за охрану сада. Какое минимальное количество яблок придётся отдать? Яблоки будем считать одинаковыми.

### Входные данные

В первой и единственной строке входных данных записаны натуральные числа  $a, b, c$ , не превышающие один миллиард.

### Выходные данные

В единственной строке вывести количество яблок, которое достанется волку.

### Примеры

#### входные данные

```
3 5 7
```

#### выходные данные

```
9
```

### Примечание

Всего собрали 15 яблок. Волк не может получить меньше половины, т.е. 8. Из оставшихся 7 яблок 6 поделит поросята (каждому по 2), и одно достанется волку. Итого  $8 + 1 = 9$  яблок.

## Идея решения

Сразу отбрасываем половину яблок волку, округлённую вверх. Оставшиеся яблоки нужно поделить на три равные части. Останется остаток от деления на 3, который также достанется волку. При решении задачи необходимо учесть очень большое возможное количество яблок, поэтому либо аккуратно работаем по отдельности с числом яблок, набранным каждым из поросят, либо используем тип данных, позволяющий хранить числа около трёх миллиардов (int64).

**Пример программы решения без сложных условий и циклов:**

```
Program z2;  
var a, b, c, sum : int64;  
begin  
  read(a, b, c);  
  sum := a + b + c;  
  write(sum div 2 + sum mod 2 + sum div 2 mod 3);  
end.
```

## 3. Пасьянс

Кот Базилио раскладывает карты из колоды по следующим правилам:

Сначала он кладёт верхнюю карту на стол, получив 1 стопку. На каждом из следующих шагов он сначала кладёт по карте на выложенные ранее стопки, потом кладёт 1 карту на пустое место, создавая ещё одну стопку, и так до тех пор, пока не кончатся карты в колоде. Сколько стопок получится у кота Базилио, если в его колоде  $n$  карт?

### Входные данные

В первой и единственной строке входных данных записано натуральное число  $n$  ( $1 \leq n \leq 10^{18}$ ) – количество карт в колоде.

### Выходные данные

В единственной строке вывести количество стопок после выкладывания всех карт из колоды.

### Примеры

#### входные данные

11

#### выходные данные

4

### Примечание

После первого шага в колоде останется  $11 - 1 = 10$  карт. После второго  $10 - 2 = 8$  карт. После третьего  $8 - 3 = 5$  карт. После четвёртого  $5 - 4 = 1$  карта. На пятом шаге не хватит карт для создания новой стопки, поэтому на столе так и останется 4 стопки (из 5, 3, 2 и 1 карты).

## Идея решения

Первое, что приходит в голову – промоделировать процесс. При этом нужно учесть большое возможное число карт в колоде, и использовать тип данных с достаточно большими ограничениями (int64). При такой реализации не пройдёт по времени последний тест.

Для полного решения нужно понять, что полностью уложенные стопки представляют собой арифметическую прогрессию. Сумма прогрессии должна стать больше либо равной числу карт, тогда количество членов прогрессии и будет ответом на задачу (если большей, то -1). Сумма элементов прогрессии считается по формуле

$\frac{a_1 + a_k}{2} k$ , где  $k$  – количество элементов прогрессии. Имеем  $\frac{1+k}{2} k = n$ ;  $\frac{k}{2} + \frac{k^2}{2} = n$ ;  $k + k^2 = 2n$ . Имеем

квадратное уравнение  $k^2 + k - 2n = 0$ . Два решения:  $x_1 = \frac{-1 + \sqrt{1^2 - 4 \cdot 1 \cdot (-2n)}}{2 \cdot 1}$ ;  $x_2 = \frac{-1 - \sqrt{1^2 - 4 \cdot 1 \cdot (-2n)}}{2 \cdot 1}$ .

Второе решение отрицательное и не годится для нас. Используем первое для решения:  $k = \left\lfloor \frac{\sqrt{1+8n}-1}{2} \right\rfloor$

#### Пример программы решения с моделированием:

```
Program z3;  
var n, m, k : int64;  
begin  
  read(n);  
  k := 0;  
  while n > 0 do begin k := k + 1; n := n - k; end;  
  if n < 0 then k := k - 1;  
  write(k);  
end.
```

#### Пример программы решения без моделирования:

```
Program z3;  
var n, m, k : int64;  
begin  
  read(n);  
  m := n; m := m * 8 + 1;  
  write(trunc((sqrt(m) - 1) / 2));  
end.
```

## 4. Ожерелье

Медведь подарил Маше красивое ожерелье – бусы из бусин трёх цветов: R (Red, красный), G (Green, зелёный) и B (Blue, синий). Будем называть ожерелье красивым, если в нём нет двух бусинок одинакового цвета подряд. Волк случайно порвал ожерелье, собрал **все** бусинки и заново нанизал на нить, после чего завязал нить. При этом ожерелье могло стать некрасивым. Сколько бусинок придётся перекрасить, чтобы сделать ожерелье красивым, не разрывая его? Считаем, что в подаренном ожерелье было как минимум по одной бусине каждого цвета. Перекрашивать можно только в цвета R, G, B, количество бусин каждого цвета не обязательно должно совпадать с исходным.

### Входные данные

В единственной строке ввода задана непустая строка из прописных символов латинского алфавита R, G, B. Длина строки не превышает 256 символов.

### Выходные данные

Единственная строка выходного файла должна содержать минимальное количество перекрашиваемых бусин.

### Примеры

#### входные данные

```
RGB
```

#### выходные данные

```
0
```

#### входные данные

```
GRGBBRBRG
```

#### выходные данные

```
2
```

### Примечание

В первом примере ожерелье уже является красивым, ничего перекрашивать не надо.

Во втором примере совпадают первая и последняя бусины – перекрашиваем первую (G) в B, также совпадают 4 и 5 – перекрашиваем 4(B) в R. Есть и другие варианты раскраски.

### Идея решения

Первое, что приходит в голову – найти все подряд идущие бусины одинаковых цветов, и заменить их на другой. Но на какой? Давайте введём 4-й цвет. Можно доказать, что при данных условиях мы сможем заменить этот цвет на один из трёх основных. Остаётся промоделировать процесс. Нужно обратить внимание, что бусы замыкаются в кольцо, поэтому нужно проверить пару первого и последнего символов.

**Пример программы решения:**

```
Program z3;  
var s : string; i, k : integer;  
begin  
  read(s); k := 0;  
  for i := 2 to Length(s) do  
    if s[i] = s[i - 1]  
      then begin  
        s[i] := '*';  
        k := k + 1;  
      end;  
  if s[1] = s[Length(s)]  
  then begin  
    s[1] := '*';  
    k := k + 1;  
  end;  
  write(k);  
end.
```