

## PROVA1 — Profa Nádia Félix

### Conteúdo

<b>1</b>	<b>Investindo no mercado de ações 1 - RECURSÃO – VALOR 2,0</b>	<b>2</b>
<b>2</b>	<b>Quem está no meio - ORDENAÇÃO – VALOR 2,0</b>	<b>3</b>
<b>3</b>	<b>Insertion Sort – VALOR 2.0</b>	<b>5</b>



(++)

## 1 Investindo no mercado de ações 1 - RECURSÃO – VALOR 2,0

João é um dos muitos investidores que vem aumentando sua fortuna nos últimos anos com negociações no mercado de ações. Curiosamente, seu patrimônio cresceu consideravelmente desde que ele resolveu adotar uma estratégia bem particular de investimento.

Considere que João possui  $N$  reais para investir e que ele nunca investe mais do que  $K$  reais em ações de uma mesma empresa, com o objetivo de diversificar sua carteira e teoricamente reduzir o seu risco. Para tanto, João divide seu capital em partes de no máximo  $K$  reais, de acordo com a estratégia descrita a seguir. Inicialmente, se  $N > K$ , João divide seu capital em duas partes de  $\lfloor N/2 \rfloor$  e  $\lceil N/2 \rceil$  reais e continua dividindo cada uma dessas partes de maneira similar, até resultar em partes de no máximo  $K$  reais cada. Ao final desse processo, João terá seu capital inicial dividido em  $E$  partes e investirá integralmente cada uma delas em ações de uma única empresa, não podendo investir mais de uma parte em uma mesma empresa. Sua tarefa consiste em ajudar João a descobrir em quantas empresas ele irá investir utilizando essa estratégia.

Por exemplo, considere que  $N = 18$  e  $K = 4$ . Após a primeira divisão João terá duas partes de 9 reais. Cada uma dessas partes será dividida, resultando em duas partes de 5 reais e duas partes de 4 reais. As partes de 5 reais são então divididas novamente, resultando em duas partes de 2 reais e duas partes de 3 reais. As partes de 4 reais não precisam mais ser divididas. Logo, todas as 6 partes resultantes (duas de 2 reais, duas de 3 reais e duas de 4 reais) possuem no máximo 4 reais e são utilizadas por João para investir em ações de 6 empresas distintas.

### Entrada

Há vários casos de teste.

Cada caso de teste é descrito em uma única linha contendo dois inteiros  $N$  e  $K$ , respectivamente o capital inicial de João ( $1 \leq N \leq 1.000.000$ ) em reais e a quantidade máxima de reais ( $1 \leq K \leq 1.000.000$ ) que João pode investir para comprar ações de uma mesma empresa.

A entrada termina com  $N = K = 0$ , que não deve ser processado.

### Saída

Para cada caso de teste, imprima uma única linha contendo um único número, a quantidade de empresas  $E$  em que João irá investir seu capital.

### Exemplo

Entrada	Saída
18 4	6
5 10	1
100 1	100
64 6	16
0 0	



(++)

## 2 Quem está no meio - ORDENAÇÃO – VALOR 2,0

FJ está examinando seu rebanho para encontrar a vaca “que está no meio”. Ele quer saber quanto leite esta vaca “mediana” dá: metade das vacas dão tanto leite ou mais do que a mediana; metade dá tanto leite ou menos.

Dado um número ímpar de vacas  $N$  ( $1 \leq N < 10.000$ ) e sua produção de leite ( $1 \dots 1.000.000$ ), encontre a quantidade mediana de leite, de tal forma que pelo menos metade das vacas dê a mesma quantidade de leite ou mais e pelo menos metade das vacas dê a mesma quantidade de leite ou menos.

### Entrada

Linha 1: Um inteiro  $N$

Linha 2  $\dots N + 1$ : Cada linha contém um único número inteiro que é a saída de leite de uma vaca.

### Saída

Linha 1: Um único número inteiro que é a saída mediana de leite.

### Exemplo

Entrada	Saída
5 2 4 1 3 5	3

### Dicas

5 vacas com produção de leite de  $1 \dots 5$ .

1 e 2 são menores que 3; 4 e 5 são maiores que 3.

Entrada	Saída
5 3 4 1 6 3	3

<b>Entrada</b>		<b>Saída</b>	
7		5	
8			
9			
10			
5			
3			
1			
2			



(++)

### 3 Insertion Sort – VALOR 2.0

*Insertion Sort* é uma técnica de ordenação clássica. Uma variante *Insertion Sort* funciona da seguinte maneira ao ordenar o vetor  $a[1 \dots N]$  em ordem não decrescente:

```
for i <- 2 to N
  j <- i
  while j > 1 and a[j] < a[j - 1]
    swap a[j] and a[j - 1]
  j <- j - 1
```

O pseudocódigo é simples de seguir. Na iteração  $i$ , o elemento  $a[i]$  é inserido na sequência ordenada  $a[1 \dots i - 1]$ . Isso é feito movendo um  $a[i]$  para trás, trocando-o com o elemento anterior até que ele acabe na posição correta.

Como você provavelmente já sabe, o algoritmo pode ser muito lento. Para estudar isso mais, você deseja descobrir o número de vezes que a operação de troca é realizada ao ordenar um vetor.

#### Entrada

A primeira linha contém o número de casos de teste  $T$ . A primeira linha para cada caso contém  $N$ , o número de elementos a serem ordenados. A próxima linha contém  $N$  inteiros  $a[1], a[2] \dots, a[N]$ .

#### Saída

Linhas de saída  $T$ , contendo a resposta necessária para cada caso de teste.

#### Restrições

$$1 \leq T \leq 5$$

$$1 \leq N \leq 100.000$$

$$1 \leq a[i] \leq 1.000.000$$

#### Exemplo

Entrada	Saída
2	0
5	4
1 1 1 2 2	
5	
2 1 3 1 2	