

# kk-página

Uma grande empresa de desenvolvimento de páginas de produtos eletrônicos está com um problema grave! Alguns *bots* e pessoas maliciosas conseguem deixar o servidor não responsivo com uma pesquisa bem curiosa.

Jaime, o rapaz do TI da empresa, percebeu que quando a pessoa clica no botão de consulta avançada do site e marca as opções: mostrar TODOS os produtos; ordenar por *ID*, e; ir para uma página *X* qualquer; o servidor demora a responder (as vezes até minutos). Para piorar, se mais pessoas fazem isso, o servidor fica com várias consultas em execução e eventualmente para de responder totalmente.

O dono da empresa, *Istivi Trabalhos*, precisa de uma ajuda mais especializada e, não por acaso, te encontrou na lista de alunos de Engenharia de Software da UnB e gostou do seu perfil e requer a sua ajuda!

O problema já foi repassado para você e temos a parte que mais interessa.

O seu programa será compilado com os parâmetros: `gcc -O2 -static arquivo.c -o arquivo`

## Entrada

A entrada é composta por um único caso de teste contendo diversas linhas. A primeira linha, do caso de teste, possui três números inteiros:

- $N$  ( $0 <= N <= 2^{25}$ ), sendo a quantidade de produtos;
- $P$  ( $0 <= P <= 2^{16}$ ), sendo a página que deve ser apresentada;
- $X$  ( $1 <= X <= 100$ ), sendo a quantidade de produtos que aparecem por página;

A seguir são apresentadas  $N$  linhas, cada uma contendo um inteiro  $ID_i$  ( $0 <= ID_i <= 2^{31}$ ) representando o  $ID$  de um produto. Não existem ids repetidos.

## Saída

Você deve imprimir os  $X$   $IDs$  da página  $P$ , ordenados de forma não decrescente.

## Exemplo

### Exemplo de entrada

```
10 3 2
1
2
3
4
5
6
7
8
9
10
```

- Atenção: A página é indexada a partir de 0, logo a página ( $P =$ )3 representa a quarta página

### Saída para o exemplo acima

```
7
8
```

## Exemplo de entrada

10 1 3  
248  
125  
378  
268  
343  
45  
78  
71  
297  
150

## Saída para o exemplo acima

125  
150  
248

## Exemplo de entrada

9 4 2  
106  
210  
270  
67  
69  
127  
303  
236  
249

## Saída para o exemplo acima

303

- **ATENÇÃO:** Cuidado quando a impressão acontece na última página, podem sobrar menos elementos que o máximo para se mostrar em cada página

*Author: Bruno Ribas*

# Notas Curiosas

## Preâmbulo

A melhor universidade de Nlognônia, a Universidade Nlogoliana de BioInformática (UNB), possui um provão anual para determinar onde está o aprendizado dos alunos.

O provão é constituído por uma nota entre 0 e 1000, e todos os alunos desta, majestosa, universidade participam.

Os problemas são variados, desde matemática básico à problemas complexos de computação e geografia.

O professor Oak está muito curioso em saber qual é a nota mais frequente entre os alunos durante o provão.

Existem muitos alunos na universidade, e por isso professor Oak, um velho professor de genética animal, não sabe processar a informação em um computador, e para pediu sua ajuda para tirar essa enfadonha curiosidade.

## Entrada

A entrada consiste de  $N + 1$  linhas. A primeira linha contém um número inteiro  $N$ , o número de alunos da universidade. As próximas  $N$  linhas possuem um inteiro,  $N_i$ , cada, representando as notas dos alunos.

## Saída

Seu programa deve imprimir apenas uma linha contendo apenas um número, a nota mais frequente da lista.

Se duas ou mais notas ocorrerem a mesma quantidade de vezes, imprima somente a maior delas.

## Restrições

- $1 \leq N \leq 128000$
- O valor de todas as notas é um inteiro entre 0 e 1000, inclusive

## Exemplo de entrada

```
10
20
25
85
40
25
90
25
40
55
40
```

## Saída para o exemplo acima

```
40
```

## Exemplo de entrada

```
6
0
500
100
1000
```

500  
250

**Saída para o exemplo acima**

500

**Exemplo de entrada**

10  
114  
750  
502  
902  
702  
823  
554  
582  
801  
872  
10  
114  
750  
502  
902  
702  
823  
554  
582  
801  
872

**Saída para o exemplo acima**

902

**Exemplo de entrada**

14  
789  
515  
335  
121  
618  
366  
524  
701  
833  
445  
782  
49  
456  
699  
14  
789  
515  
335  
121  
618  
366  
524  
701  
833  
445  
782

49  
456  
699

Saída para o exemplo acima

833

*Author: Bruno Ribas, baseado no problema da OBI*