

# Primeira Atividade de Avaliação EDA-2 2020/1

## [25 pontos] - T1: Estabilidade de um algoritmo de ordenação

A que se refere a estabilidade de um algoritmo de ordenação? Mostre um exemplo que exercite a estabilidade (ou a falta dela) no algoritmo de ordenação InsertionSort.

## [25 pontos] - T2: Separa

Um programador inexperiente afirma que a seguinte implementação da função de separação rearranja o vetor  $v[p..r]$ , com  $p < r$ , e devolve um índice  $j$  em  $p..r-1$  tal que  $v[p..j] \leq v[j+1..r]$ .

```
int sep( int v[], int p, int r) {
    int q, i, j, t;
    i = p; q = (p + r) / 2; j = r;
    do {
        while (v[i] < v[q]) ++i;
        while (v[j] > v[q]) --j;
        if (i <= j) {
            t = v[i], v[i] = v[j], v[j] = t;
            ++i, --j;
        }
    } while (i <= j);
    return i;
}
```

Mostre um exemplo onde essa função não dá o resultado esperado. E se trocarmos `return i` por `return i-1`? É possível fazer algumas poucas correções de modo que a função dê o resultado esperado?

# 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 de mostrar TODOS os produtos, ordenar por  $ID$  e ir para uma página  $X$  qualquer o servidor demora a reponder (as vezes até minutos), e para piorar se mais pessoas fazem isso o servidor fica com várias consulta 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:

- $N$  ( $0 \leq N \leq 2^{25}$ ), sendo a quantidade de produtos;
- $P$  ( $0 \leq P \leq 2^{16}$ ), sendo a página que deve ser apresentada;
- $X$  ( $10 \leq X \leq 100$ ), sendo a quantidade de produtos que aparecem por página;

A seguir são apresentada  $N$  linhas, cada uma contendo um inteiro  $ID_i$  ( $0 \leq ID_i \leq 2^{31}$ ) 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*