
Maior quantidade de vendas

Autor: João Vítor Rocon Maia

nome do arquivo: vendas.c, vendas.cpp, vendas.java

Você faz parte da equipe de desenvolvimento do Sucatas.com.br que é um e-commerce focado em vender produtos usados. Seu chefe resolveu criar uma nova área do sistema administrativo e conta com o conhecimento de sua equipe para criar essa nova funcionalidade.

É uma verdade que aplicações com o passar dos tempos crescem em termos de volume de dados e quanto maior for o tempo e maior a quantidade de usuários estes numeros podem chegar a valores assustadores.

Na última reunião foi definido que na aplicação da sua empresa será criado um widget novo onde poderá ser definido uma data de inicio e fim e o resultado será o valor de produtos vendidos no dia que teve mais vendas.

Sabendo do problema do volume de dados sua equipe estudou a implementação de um algoritmo eficiente para resolver esse problema. Como forma de prova de conceito sua equipe decidiu implementar uma abstração do problema usando uma lista de inteiros simples.

Sua missão é escrever esse algoritmo de prova de conceito.

Entrada

A primeira linha de cada caso de teste é composta por um valor N ($2 \leq N \leq 50000$) onde N é a quantidade de dias, no caso de $N = 0$ seu programa deve encerrar a execução. Na linha seguinte serão dados N valores de vendas separados por espaço.

Na terceira linha será dado um valor M ($1 \leq M \leq 100$) onde M é a quantidade de consultas a serem realizadas na sua coleção de dados sobre a quantidade de vendas. As M linhas seguintes são compostas por dois inteiros separados por espaço, o primeiro valor será sempre menor ou igual ao segundo.

Observação: considere que a lista contendo os N dias possui o indice começando por 0 (Zero).

Saída

Cada entrada começa com a uma linha indicando o numero do teste. Após isso, cada linha seguinte deve retornar o valor máximo das consultas realizadas.

Os testes são separados por uma linha em branco, no último caso de teste não tem a linha em branco.

Exemplo

Entrada

```
10
1 10 2 9 3 8 4 7 5 6
5
0 9
1 3
3 8
7 8
2 5
2
1 3
1
0 1
0
```

Saída

```
Entrada #1:
10
10
9
7
9

Entrada #2:
3
```