

Exercício Programa 0: Pista de Esqui

Rodrigo de Souza

Prazo: 28 de Agosto de 2016 (23h55)

Uma estação de esqui é frequentemente afetada por avalanches que causam danos em suas pistas. Uma pista é representada por uma sequência de números inteiros em ordem crescente, indicando as altitudes em diversos setores da pista. Os desastres naturais tem a particularidade de só afetar um trecho final da pista, a partir de uma determinada posição, e seu efeito é trocar os setores de lugar a partir dessa posição, embaralhando-os, mas sem alterar as altitudes.

A fim de realizar os reparos, a administração da pista precisa saber a sequência original da pista (em ordem crescente). Uma restrição importante é que pede-se que os reparos sejam feitos apenas no trecho afetado, sem importar-se com o trecho inicial da pista que não foi afetado. Para isso, ela contratou os seus serviços.

1 Entrada

A entrada deve ser lida de um arquivo e consiste de diversos casos de teste, cada um em uma linha. Um caso de teste é formado por uma sequência de inteiros positivos: o primeiro inteiro é o número n de setores da pista; em seguida, tem-se n inteiros representando uma pista (possivelmente avariada por uma avalanche). Uma linha cujo primeiro número é 0 representa o final da entrada.

Seu programa deve, para cada linha, escrever na tela a posição da pista onde começa o trecho atingido pela avalanche (assuma que as posições da pista são numeradas de 1 a n) e a sequência original da pista começando nesse trecho. Caso a pista não tenha sido afetada, seu programa escreve simplesmente -1 .

Recorde da restrição de que operações na pista só devem ser feitas no trecho afetado. Ou seja: após descobrir onde começa esse trecho, as posições anteriores não podem ser acessadas, manipuladas, etc.

2 Exemplo

Eis um exemplo de arquivo de entrada:

```
5 10 15 8 17 21
8 5 12 17 25 14 18 13 17
3 20 23 31
0
```

Para esse arquivo, a saída do seu programa deve ser a seguinte:

```
1 8 10 15 17 21
3 13 14 17 18 25
-1
```

3 Instruções gerais

- Seu programa deve ser feito em C. Você não pode usar algoritmos de busca e ordenação já implementados na linguagem, mas deve implementá-los explicitamente. Para ordenação, nós aprendemos um algoritmo na última aula, a *ordenação por inserção*; você pode ver uma implementação dele em C aqui: <http://www.ime.usp.br/~pf/algoritmos/aulas/ordena.html>. Você pode usar outro algoritmo, se quiser.
- Não tente embelezar a saída do seu programa com mensagens, formatações, etc. Seu programa será julgado segundo a adequação de sua saída à descrição do exercício.
- Seu programa deve consistir de um único arquivo, e deve ser razoavelmente modular, ou seja: deve consistir de diversas funções que, juntas, realizam a tarefa descrita.
- Documente cada função dizendo o quê ela faz.
- Escreva no início do código um cabeçalho com comentários, indicando nome, número do EP, data, nome da disciplina.
- A descoberta da posição inicial da avalanche deve ser tão eficiente quanto possível.
- A entrega deverá ser feita no ambiente Moodle, no espaço reservado para essa finalidade. Consulte o ambiente para instruções. A entrega será eletrônica (não receberei exercícios impressos).