

Universidade Federal do ABC
BCM0505–15 — Processamento da Informação — Prática
Simulado 2

Primeiro Quadrimestre de 2018

- Escreva um programa que recebe como entrada 3 inteiros, S , N e M , $N, M > 0$ seguidos de $N \times M$ inteiros. Esses inteiros são os elementos de uma matriz com N linhas e M colunas fornecidos linha a linha (veja exemplo abaixo). O seu código deve então imprimir na saída padrão todos os pares de coordenadas de elementos vizinhos da matriz cuja soma seja igual a S . Um elemento é vizinho de outro se ele está à sua esquerda, direita, acima ou abaixo. Note que os pares “(0,0)-(1,0)” e “(1,0)-(0,0)” são equivalentes e apenas um deles deve estar presente na saída do programa. Atente-se para o fato de que os elementos da primeira linha, da primeira coluna, da última linha e da última coluna não tem vizinhos acima, à esquerda, abaixo ou à direita respectivamente.

Exemplo: $S = 2$, $M = 3$, $N = 4$

$$\begin{pmatrix} \boxed{2} & 5 & 27 & 8 \\ \boxed{0} & 13 & 50 & 78 \\ \boxed{2} & 6 & \boxed{7} & \boxed{-5} \end{pmatrix}$$

ENTRADA	SAÍDA
2 3 4 2 5 27 8 0 13 50 78 2 6 7 -5	(0,0)-(1,0) (1,0)-(2,0) (2,2)-(2,3)

- Entremeando vetores

- Implemente uma função com a assinatura abaixo:

```
public static int[] entremeia(int[] v1, int[] v2)
```

Sua função deverá receber como parâmetro dois vetores de inteiros $v1$ e $v2$. Ambos conterão elementos em ordem crescente, ou seja, dados quaisquer dois elementos a e b de um desses vetores, se o índice de a for menor do que o índice de b , então $a \leq b$. Sua função deverá devolver como resultado um vetor cujo tamanho seja igual a $v1.length + v2.length$, que contenha todos os elementos de $v1$ e $v2$ e que também esteja em ordem crescente.

- Escreva um programa que leia da entrada padrão um inteiro $N > 0$, seguido de N inteiros dados em ordem crescente. Em seguida o seu programa deverá ler um inteiro $M > 0$, seguido de M inteiros também em ordem crescente. O seu programa deverá, então, imprimir os $N + M$ números lidos em ordem crescente.

ENTRADA	SAÍDA
4 1 7 8 9 5 0 2 3 9 10	0 1 2 3 7 8 9 9 10

3. Algoritmos de ordenação são muito comuns em computação. Eles são utilizados, por exemplo, para mostrar a sua lista de músicas em ordem de preferência ou ainda para listar os vídeos do YouTube por número de exibições. Um dos mais simples algoritmos de ordenação utiliza vetores para armazenar os elementos a serem ordenados e trabalha através da troca de posições entre elementos adjacentes que estão fora de ordem. Por exemplo, imagine que o seu programa receba o seguinte vetor de inteiros (9, 5, 2, 3, 1). A saída desejada é então: (1, 2, 3, 5, 9). Para chegar a este ponto, o algoritmo varre o vetor $N - 1$ vezes comparando os elementos adjacentes, $v[i]$ e $v[i + 1]$ e efetuando a troca entre eles caso não estejam em ordem crescente, ou seja, se $v[i + 1] < v[i]$.

Exemplo:

- Entrada: (9, 5, 2, 3, 1) → Vetor \mathbf{V} , com $N = 5$ elementos.
- 1ª varredura:
 - (9, 5, 2, 3, 1) → Os elementos nas posições 0 e 1 são comparados. Como $V[1] < V[0]$, estes elementos são trocados de posição.
 - (5, 9, 2, 3, 1) → Elementos nas posições 1 e 2 são comparados, efetua-se a troca.
 - (5, 2, 9, 3, 1) → Elementos nas posições 2 e 3 são comparados, efetua-se a troca.
 - (5, 2, 3, 9, 1) → Elementos nas posições 3 e 4 são comparados, efetua-se a troca.
- 2ª varredura:
 - (5, 2, 3, 1, 9) → Repetimos o processo a partir do início do vetor. Comparamos os elementos nas posições 0 e 1. Efetuamos a troca.
 - (2, 5, 3, 1, 9) → Elementos nas posições 1 e 2 são comparados, efetua-se a troca.
 - (2, 3, 5, 1, 9) → Elementos nas posições 2 e 3 são comparados, efetua-se a troca.
 - Sabemos que o último elemento já está na sua posição correta, pois a 1ª varredura já se encarregou disto. Logo podemos parar as comparações no penúltimo elemento (índice $N - 2$).
- 3ª Varredura
 - (2, 3, 1, 5, 9) → Repetimos o processo a partir do início do vetor. Comparamos os elementos nas posições 0 e 1. Já estão em ordem crescente, **não** efetuamos a troca.
 - (2, 3, 1, 5, 9) → Elementos nas posições 1 e 2 são comparados, efetua-se a troca.
 - A 2ª e 1ª varreduras já se encarregaram de deixar os 2 últimos elementos em ordem. Assim, paramos a comparação no elemento de índice $N - 3$.
- 4ª Varredura
 - (2, 1, 3, 5, 9) → Repetimos o processo a partir do início do vetor. Comparamos os elementos nas posições 0 e 1. Efetuamos a troca.

– Paramos a comparação no elemento com índice $N - 4$.

- Vetor após a última varredura (1, 2, 3, 5, 9)

Escreva um programa que leia da entrada padrão N inteiro e, em seguida, leia N números inteiros distintos. Seu programa deverá então imprimir os N números lidos em ordem crescente.

ENTRADA	SAÍDA
5 9 5 2 3 1	1 2 3 5 9

4. A Lotofácil é uma loteria onde o jogador pode escolher, na aposta mais simples, 15 números **distintos** dentre os 25 disponíveis (numerados sequencialmente a partir de 1) no volante de jogo. Sua tarefa é escrever um programa que escreva na saída padrão uma lista com 15 números aleatórios que formem um jogo válido.

SAÍDA (não determinística)
1 2 3 5 7 8 9 14 15 16 17 20 23 24 25