



Instituto de Informática – INF
Cursos de Bacharelado do INF/UFV (NBC)

Avaliação

Disciplina:	Algoritmos & Estruturas de Dados – 1	
Professor:	Wanderley de Souza Alencar	Nota:
Aluno(a):		Matrícula:
1ª Avaliação Formal – 2020/1		Data: 29/10/2020

INFORMAÇÕES IMPORTANTES

1. Esta avaliação é individual, sem consulta a qualquer colega de turma, estudante de curso das áreas de Informática e/ou Computação (seja da UFV ou de quaisquer outras IESs brasileiras ou estrangeiras) ou profissionais das mencionadas áreas. Apesar disso, você pode utilizar consultas à Internet, bem como fontes bibliográficas quaisquer;
2. Você pode resolver qualquer quantidade de questões com o objetivo de atingir a nota máxima desta avaliação, que é 10,0 (dez pontos);
3. As resoluções das questões devem ser submetidas ao *Sharif Judge System* <<https://sharif.inf.ufv.br/wanderley>> na atividade denominada “Av1 - PRIMEIRA avaliação formal”. A entrega deverá ser realizada até às 23h59min do dia 30 de outubro de 2020, sexta-feira. **Não haverá prorrogação no prazo de entrega.**
Observação: Apesar de improvável, se, e somente se, o *Sharif* ficar inoperante durante todo o período de entrega, você deverá nomear cada uma de suas resoluções da seguinte maneira: `questao01.c`, `questao02.c`, `questao03.c` e `questao04.c`.
Em seguida, você deve criar um único arquivo “.zip”(contendo todos os arquivos .c/.cpp anteriores) com o seu nome completo. Por exemplo, `JoaquimMariaMachadoDeAssis.zip` e enviá-lo para o e-mail `wanderleyalencar@ufv.br`.
4. Todas as questões devem ser resolvidas empregando o conceito de TAD (Tipo Abstrato de Dados), ou seja, os programas elaborados (em C ou C++) devem utilizar TAD para declarar as principais estruturas que utilizarem.
Observação: Devido às restrições de submissão de programas-fonte impostas pelo *Sharif*, a resolução de cada questão corresponde a um único programa-fonte que conterá todas as declarações, funções e corpo principal do programa (função *main*).

Questão 01 (3,0 pontos) Um(a) estudante de programação está aprendendo o conceito de *recursividade* e, por isso, resolveu desafiar-se: apenas implementar código-fonte recursivo para todos os problemas que tiver que resolver computacionalmente a partir de hoje. Abrindo um livro cujo tema é programação, viu o seguinte problema proposto: “Escrever, utilizando a linguagem C, um programa que seja capaz de receber como entrada três números: n , x e y , nesta ordem, e imprimir uma *matriz moldura* de ordem $n \times n$, sendo x o valor da moldura mais externa.

Sabe-se que os valores obedecem às seguintes regras: $1 \leq n \leq 100$; $1 \leq x, y < 10$; $x \neq y$ e n é sempre um número ímpar.”

Observação:

Uma *matriz moldura* de ordem $n \times n$, com valores x e y é uma matriz com a seguinte forma geral:

	1	2	3	...	i	...	(n-2)	(n-1)	n
1	x	x	x	x	x	x	x	x	x
2	x	y	y	y	y	y	y	y	x
3	x	y	x	x	x	x	x	y	x
...	x	y	x	x	y	x
i	x	y	x	...	y	...	x	y	x
...	x	y	x	x	y	x
(n-2)	x	y	x	x	x	x	x	y	x
(n-1)	x	y	y	y	y	y	y	y	x
n	x	x	x	x	x	x	x	x	x

Por exemplo, se tivermos que $n = 7$, $x = 1$ e $y = 0$, então a *matriz moldura* será:

1	1	1	1	1	1	1
1	0	0	0	0	0	1
1	0	1	1	1	0	1
1	0	1	0	1	0	1
1	0	1	1	1	0	1
1	0	0	0	0	0	1
1	1	1	1	1	1	1

Outro exemplo é $n = 5$, $x = 2$ e $y = 0$, tem-se:

2	2	2	2	2
2	0	0	0	2
2	0	2	0	2
2	0	0	0	2
2	2	2	2	2

Entrada: Numa única linha, os valores de n , x e y , nesta ordem, e separados entre si por um único espaço em branco.

Saída: A *matriz moldura* conforme apresentada anteriormente, ou seja, com n linhas e n colunas por linha, com elementos da linha sempre separados por um único espaço em branco entre si.

Nota: O programa elaborado deve, necessariamente, utilizar *recursividade* para ser considerado válido. Assim, mesmo que todos os casos de teste estejam corretos de acordo

com a avaliação do *Sharif Judge System*, se o programa não empregar *recursividade* o professor poderá atribuir-lhe nota 0,0 (zero).

Questão 02 (7,0 pontos). Seja dado um vetor \mathcal{V}_A contendo n números naturais, com $n \in \mathbb{N}^*$ e $n \leq 10^4$. Construa um programa que seja capaz de determinar, e imprimir, qual é o número de pares (i, j) , com $i \leq j$, cujo valor $(\mathcal{V}_A[i] \times \mathcal{V}_A[j])$ é máximo.

Entrada: Na primeira linha, tem-se o número de casos de teste, $t \in \mathbb{N}^*$ e $t \leq 10$.

Cada par de linhas seguinte representa um caso de teste, sendo que a primeira linha do par contém o número n , ou seja, o número de elementos do vetor \mathcal{V}_A e a segunda linha do par contém os valores dos elementos separados, entre si, por um único espaço em branco e fornecidos do primeiro elemento ao último elemento (1° , 2° , 3° , ..., n -ésimo).

Saída: Contém, em cada linha, o número de pares distintos que atendem à regra estabelecida anteriormente.

Observação: Sabe-se que os elementos do vetor \mathcal{V}_A são menores ou iguais a 10^6 .

Exemplo 01:

ENTRADA	SAÍDA
3	1
4	2
1 2 3 4	15
3	
1 2 1	
5	
1 1 1 1 1	

Explicação para as saídas geradas:

1º teste : Os pares possíveis são os seguintes: (1,1); (1,2); (1,3); (1,4); (2,2); (2,3); (2,4); (3,3); (3,4) e (4,4). Multiplicando-se os respectivos valores de $\mathcal{V}_A[i]$ e $\mathcal{V}_A[j]$ tem-se, respectivamente: 1, 2, 3, 4, 4, 6, 8, 9, 12 e 16. O produto máximo tem valor igual a 16, e corresponde ao produto de $\mathcal{V}_A[4]$ com $\mathcal{V}_A[4]$. Portanto, tem-se uma única ocorrência do valor máximo e, por isso, a saída é igual a 1 para este caso de teste;

2º teste : Os pares possíveis são os seguintes: (1,1); (1,2); (1,3); (2,2); (2,3) e (3,3). Multiplicando-se os respectivos valores de $\mathcal{V}_A[i]$ e $\mathcal{V}_A[j]$ tem-se, respectivamente: 1, 2, 1, 4, 2 e 1. O produto máximo tem valor igual a 4, e corresponde ao produto de $\mathcal{V}_A[2]$ com $\mathcal{V}_A[2]$. Portanto, tem-se uma única ocorrência do valor máximo e, por isso, a saída é igual 1 para este caso de teste.

3º teste : Os pares possíveis são os seguintes: (1,1); (1,2); (1,3); (1,4); (1,5); (2,2); (2,3); (2,4); (2,5); (3,3); (3,4); (3,5); (4,4); (4,5) e (5,5). Multiplicando-se os respectivos valores de $\mathcal{V}_A[i]$ e $\mathcal{V}_A[j]$ tem-se que todos resultam em 1. Logo o produto máximo tem valor igual a 1 e ocorre quinze vezes, pois este é o número total de pares ordenados gerados. Por isso, a saída é igual a 15.

Exemplo 02:

ENTRADA	SAÍDA
2	1
4	3
4 3 2 1	
6	
1 2 3 3 2 1	

Explicação para as saídas geradas:

1º teste : Os pares possíveis são os seguintes: (1,1); (1,2); (1,3); (1,4); (2,2); (2,3); (2,4); (3,3); (3,4) e (4,4). Multiplicando-se os respectivos valores de $\mathcal{V}_A[i]$ e $\mathcal{V}_A[j]$ tem-se, respectivamente: 16, 12, 8, 4, 9, 6, 3, 4, 2 e 1. O produto máximo tem valor igual a 16, e corresponde ao produto de $\mathcal{V}_A[1]$ com $\mathcal{V}_A[1]$. Portanto, tem-se uma única ocorrência do valor máximo e, por isso, a saída é igual a 1 para este caso de teste;

2º teste : Os pares possíveis são os seguintes: (1,1); (1,2); (1,3); (1,4); (1,5); (1,6); (2,2); (2,3); (2,4); (2,5); (2,6); (3,3); (3,4); (3,5); (3,6); (4,4); (4,5); (4,6); (5,5); (5,6) e (6,6). Multiplicando-se os respectivos valores de $\mathcal{V}_A[i]$ e $\mathcal{V}_A[j]$ tem-se, respectivamente: 1, 2, 3, 3, 2, 1, 4, 6, 6, 4, 2, 9, 9, 6, 3, 9, 6, 3, 4, 2 e 1. O produto máximo tem valor igual a 9 e ocorre três vezes e, por isso, a saída é igual 3 para este caso de teste.

Questão 03 (5,0 pontos) Uma estudante universitária precisa desenvolver um programa \mathbb{C} para resolver o seguinte problema:

“A partir de uma lista de entrada \mathcal{L}_A , contendo uma sequência de n números inteiros, é necessário gerar uma lista de saída \mathcal{L}_B , com os x maiores números e os y menores números da lista original, apresentados em ordem decrescente.”

Entrada:

A primeira linha da entrada contém três números naturais n , x e y , nesta ordem e separados por um único espaço em branco entre eles. Sabe-se que $1 \leq x, y \leq n \leq 100$ e $(x + y) \leq n$.

A segunda linha contém os n números inteiros, cada um deles entre -1000 e $+1000$, inclusive extremos, separados por um único espaço em branco entre eles.

Saída:

A saída contém uma única linha, com os $(x + y)$ maiores e menores números da lista de entrada, apresentados em ordem decrescente de valor.

Exemplo 01:

ENTRADA	SAÍDA
10 3 2	91 74 65 13 10
10 13 21 36 65 18 47 74 61 91	

Explicação para a saída gerada:

A primeira linha da entrada indica que a lista de entrada tem 10 números e deseja-se obter os 3 maiores e os 2 menores. A segunda linha contém os 10 números. Portanto, os três maiores são 91, 74 e 65, mostrados em ordem decrescente. Os dois menores são o 13 e o 10, também já mostrados em ordem decrescente de valor.

Assim a única linha da saída deve conter, em ordem decrescente, os seguintes números: 91, 74, 65, 13 e 10.

Exemplo 02:

ENTRADA	SAÍDA
5 2 2	1000 500 50 5
50 5 500 1000 500	

Explicação para a saída gerada:

A primeira linha da entrada indica que a lista de entrada tem 5 números e deseja-se obter os 2 maiores e os 2 menores. A segunda linha contém os 5 números. Portanto, os dois maiores são 1000 e 500, mostrados em ordem decrescente. Os dois menores são o 50 e o 5, também já mostrados em ordem decrescente de valor. Assim a única linha da saída deve conter, em ordem decrescente, os seguintes números: 1000, 500, 50 e 5.

Exemplo 03:

ENTRADA	SAÍDA
9 4 4	58 23 23 20 19 16 14 14
23 58 23 20 19 14 19 14 16	

Observação: Perceba que ao tomar os quatro maiores números o número 23 aparece duas vezes e ao considerar os quatro menores, um dos 19 (dezenove) foi excluído da saída, já que a lista final deve conter apenas oito números.

Nota: O programa elaborado deve, necessariamente, utilizar *Lista Linear Simplemente Encadeada* (LLSE) para ser considerado válido. Assim, mesmo que todos os casos de teste estejam corretos de acordo com a correção do *Sharif Judge System*, se o programa não empregar LLSE o professor poderá atribuir-lhe nota 0,0 (zero).

Questão 04 (5,0 pontos) Você, que participa da equipe de desenvolvimento de *software* de uma empresa e está incumbido(a) de resolver o seguinte problema:

“Um usuário fornece uma lista \mathcal{L}_1 de até n números naturais, em ordem, sendo que cada um deles está intervalo de 1 a N , incluindo extremos. Há a possibilidade de múltiplas ocorrências de um número na lista.

Deseja-se obter uma *nova* lista, \mathcal{L}_2 , a partir da lista \mathcal{L}_1 aplicando-se, nesta ordem, as seguintes regras:

- (1) Eliminam-se as múltiplas ocorrências de números, preservando somente a primeira ocorrência na lista de entrada;
- (2) Faz-se com que o primeiro número da lista se torne o último, que o segundo se torne o penúltimo, que o terceiro se torne o antepenúltimo e, assim, sucessivamente.”

Entrada:

A primeira linha da entrada conterà o número de casos de teste, t , com $1 \leq t \leq 10$. Cada uma das t linhas seguintes conterà uma sequência i números naturais, com $1 \leq i \leq n \leq 100$ e $0 \leq n_i \leq N \leq (2^{32} - 1)$, onde n_i denota o i -ésimo elemento de um certo caso de teste.

Observação: Em cada caso de teste, os números estão separados por um único espaço em branco entre eles e o último número é sempre igual a -1 , para indicar a finalização da lista de números daquele caso de teste (um *flag* de término de lista).

Saída:

A saída conterà t linhas, cada uma delas correspondendo à saída do respectivo caso de teste e obedecendo à regra imposta para o problema.

Observação: Os números de saída, de cada caso de teste, devem estar separados por um único espaço em branco entre eles.

Exemplo 01:

ENTRADA	SAÍDA
4	6 16 20 29 13 1 9 2 5 7 3 8
8 3 7 7 5 2 9 5 2 1 13 29 20 16 6 -1	1 5 7 10 9 2 8
8 2 8 9 9 10 7 5 2 1 -1	92 78 36 44 45 89 25 14
14 25 89 45 44 36 78 92 -1	10
10 10 10 10 10 10 10 -1	

Explicação para a saída gerada:

- 1º teste** : A lista contém quinze números, sendo que há repetição do 7, 5 e do 2. Aplicando-se a primeira regra, as repetições são eliminadas da lista: 8, 3, 7, 5, 2, 9, 1, 13, 29, 20, 16, 6. Em seguida a aplicação da segunda regra gera a saída: 6 16 20 29 13 1 9 2 5 7 3 8;
- 2º teste** : A lista contém dez números, sendo que há repetição do 8, 2 e do 9. Aplicando-se a primeira regra, as repetições são eliminadas da lista: 8, 2, 9, 10, 7, 5, 1. Em seguida a aplicação da segunda regra gera a saída: 1 5 7 10 9 2 8;
- 3º teste** : A lista contém oito números, sendo que não há repetição de nenhum deles. Assim a primeira não alterará a lista. A aplicação da segunda regra gera a saída: 92 78 36 44 45 89 25 14;
- 4º teste** : A lista contém sete números iguais. Assim, após a aplicação da primeira regra, restará apenas o número 10, que integrará a lista de saída:10.

Exemplo 02:

ENTRADA	SAÍDA
1	11 12 9 10 7 8 5 6 3 4 1 2
2 1 4 3 6 5 8 7 10 9 12 11 -1	

Nota: O programa elaborado deve, necessariamente, utilizar o conceito de *Lista Linear Simplesmente Encadeada* (LLSE) para ser considerado válido. Assim, mesmo que todos os casos de teste estejam corretos de acordo com a avaliação do *Sharif Judge System*, se o programa não empregar LLSE o professor poderá atribuir-lhe nota 0,0 (zero).

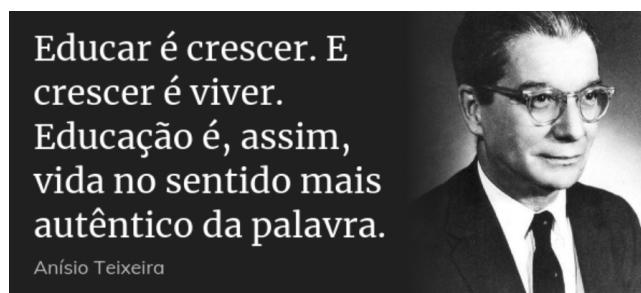


Figura 1: Anísio Spínola Teixeira, o inventor da escola pública no Brasil.