



Universidade de São Paulo - São Carlos, SP

Instituto de Ciências Matemáticas e de Computação

USP – ICMC – SSC

SSC0501 - Introdução à Ciência da Computação I (Teórica)

Professor responsável: *Fernando Santos Osório*

Semestre: 2010/1

Bolsista PAE: Gustavo Pessin

Monitor: Matheus Lin (Seg. e Sexta 18h -19h)

Horário: Seg. 21h-22h40 e Terça 19h-20h40

E-mail: fosorio .at. icmc .dot. USP .dot. br

fosorio .at. gmail .dot. com

pessin .at. gmail .dot. com

matheus.lin .at. gmail .dot. com

Web: <http://www.icmc.usp.br/~fosorio/>

LISTA DE EXERCÍCIOS - Nro. 05

[VETORES]

1. Faça um programa que leia um conjunto de 10 valores, armazenando em um vetor. Uma vez lidos os valores, exibir na ordem inversa em que foram lidos os dados, ou seja, o último dado a ser exibido na tela deve ser o primeiro que foi lido. Exemplo:

```
>> Guardando dados em um Vetor <<
Digite 10 valores: 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
Dados Lidos:      10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0
```

2. Altere o programa anterior de modo a validar os dados fornecidos, supondo que estes dados sejam as notas de alunos. Uma nota deve ter um valor entre 0 e 10, sendo que o programa deve repetir a leitura das notas até que o usuário digite uma nota válida. Exemplo:

```
>> Digitar 10 valores entre 0.0 e 10.0 <<
Digite o valor 1: -1.0
Valor Invalido!
Digite o valor 1: 5.45
Digite o valor 2: 10.0
Digite o valor 3: 150.0
Valor Inválido!
Digite o valor 3: 4.56
...
Digite o valor 10: 5.67
Valores Digitados: 5.67... 4.56 10.0 5.45
```

3. Baseado nos programas anteriores, uma vez concluída a leitura das 10 notas para o vetor, exibir na tela o maior, o menor valor, e a média dos valores que foram lidos. Exemplo:

```
>> Exibe a Media, Maior e Menor Nota <<
Digite 10 Notas: 6.0 6.1 6.2 6.3 9.2 6.5 6.7 6.8 6.9 3.4
Maior Nota: 9.2
Menor Nota: 3.4
Média das Notas: 6.41
```

4. Baseado no programa anterior (exercício 3), uma vez concluída a leitura das 10 notas para o vetor, exibir na tela o maior, o menor valor e a média dentre os valores que foram lidos, **juntamente com o seu respectivo índice no vetor**. Exemplo:

```
>> Exibe a Media, Maior e Menor Nota <<
Digite 10 Notas: 6.0 6.1 6.2 6.3 9.2 6.5 6.7 6.8 6.9 3.4
Maior Nota[4]: 9.2
Menor Nota[9]: 3.4
Média das Notas: 6.41
```

5. Faça um que leia um conjunto de **até 20** notas – não é necessário fornecer sempre todas as 20 notas. Os valores das notas devem ser entre 0.0 e 10.0, onde devem ser validas estas notas fornecidas pelo usuário (ler novamente as notas inválidas), e depois armazenadas em uma tabela. Após ler cada nota, pergunte se o usuário deseja fornecer mais uma nova nota (sim/não). Uma vez lidos todos valores, exibir na tela a lista de valores lidos, juntamente com o seu índice, e a média geral. Exemplo:

```
>> Ler Notas de uma Turma de no Maximo 20 Alunos <<
Digite uma nota: 6.0
Deseja entrar com mais dados (S/N)? s
Digite uma nota: 6.1
Deseja entrar com mais dados (S/N)? s
Digite uma nota: 62.0
Nota Invalida! Digite novamente!
Digite uma nota: 6.2
Deseja entrar com mais dados (S/N)? n

Notas fornecidas:
Nota[1] = 6.0
Nota[2] = 6.1
Nota[3] = 6.2
Media das Notas: 6.2
```

Obs.: O programa **não** deve permitir que sejam fornecidas mais que 20 notas!

6. Fazer um programa de “criptografia” (codificação de dados visando a privacidade de acesso as informações), onde dada uma **string** (vetor de caracteres) este programa codifique os dados através de um processo de substituição de letras. Você pode definir o seu próprio método de criptografia, desde que depois seja possível reverter este processo, ou seja, um código criptografado deve poder ser convertido novamente ao valor inicial. Exemplo:

```
>> Criptografador – Codifica uma String <<
Entre como texto (string) a ser criptografado: LinguagemC
Texto criptografado: MjohvbhfnD
```

Dicas:

Strings são vetores de caracteres, por exemplo, `char Texto[80]; /* texto de até 80 letras */`
Strings são vetores com seqüências de caracteres terminados por um `'\0'` (NULL)
Caracteres também permitem operações numéricas como por exemplo: `Letra = Letra + 1;`
Somar 1 ao código de uma letra, implica em transformar esta no caracter seguinte (Tabela ASCII¹).

¹ Tabela ASCII: Tabela de códigos, onde cada código corresponde a um caracter usado no computador. Para maiores informações sobre a Tabela ASCII consulte a Wikipedia - <http://pt.wikipedia.org/wiki/ASCII>

7. Fazer um segundo programa, complementar a este anterior, que deve ser capaz de descriptografar a string, ou seja, deve pegar uma string codificada e retornar ao texto original. Exemplo:

```
>> Descriptografador – Decodifica uma String <<
Entre como texto a ser descriptografado: MjohvbhfnD
Texto descriptografado: LinguagemC
```

Dicas:

Subtrair um do código de uma caracter implica em transformar no caracter anterior. Exemplo:
Letra = 'A'; Letra = Letra + 1; /* Agora a variável Letra irá conter a Letra 'B' */

8. Fazer três programas, um programa com FOR, um programa com WHILE, e um programa com DO-WHILE, que leiam uma string e convertam todos os caracteres desta string para maiúsculo. Depois de convertida a string, exibir o resultado na tela. Dicas: (1) Uma string de caracteres é um vetor do tipo char, terminado pelo caractere '\0'; (2) O código ASCII da letra 'a' (minúsculo) é 97 e o código da letra 'A' (maiúsculo) é 65, o código da letra 'z' é 122 e o código da letra 'Z' é 90; (3) Cuidado para não acabar convertendo aquilo que não é nem texto e nem está em minúsculas! Exemplo:

```
>> Converte Texto: Minusculas para Maiusculas <<
Entre um texto : Prezados Alunos da USP de Sao Carlos (2009)
Texto convertido: PREZADOS ALUNOS DA USP DE SAO CARLOS (2009)
```

9. Faça um programa que compare duas strings (vetores de caracteres), indicando se são iguais ou diferentes. Exemplo:

```
>> Compara Textos <<
Entre com o primeiro texto: vamos comparar 2 textos?
Entre com o segundo texto: vamos comparar 8 textos?

Os dois textos sao diferentes!
```

10. Faça um programa que realiza a leitura dos seguintes dados relativos a um conjunto de alunos: Nro. USP, Cod.Disc., Nota1 e Nota2. Considere uma turma de até 10 alunos. Após ler todos os dados digitados, e depois de armazená-los em vetores, exibir na tela a listagem final dos alunos com as suas respectivas médias finais (use uma média ponderada: Nota1 com peso=1.0 e nota 2 com peso=2.0). **Dica:** use múltiplos vetores, um para cada dado, com o mesmo índice para os dados de um mesmo aluno. Exemplo:

```
Aluno 1 - Numero USP [0] = 1234567
Aluno 1 - Cod. Disciplina[0] = 301
Aluno 1 - Nota1[0] = 8.0
Aluno 1 - Nota2[0] = 9.0
Aluno 1 - Média = 8.66

Aluno 2 - Numero USP [1] = 9876543
Aluno 2 - Cod. Disciplina [1] = 301
Aluno 2 - Nota1[2] = 6.0
...
```

11. Uma empresa está preparando o reajuste de preços de venda de seus produtos (10 produtos), para isto ela solicita que você implemente um programa que primeiramente crie uma tabela com a lista dos produtos vendidos pela empresa: código do produto e preço (você pode armazenar em 2 vetores, como no exercício anterior). Uma vez lida esta tabela de preços o gerente da empresa irá fornecer o índice de reajuste (desconto percentual) a ser aplicado nos produtos e deseja ver na tela como irá ficar a nova tabela de preços. Exemplo de uso do programa:

Produto [0]: **321**
Preço [0]: **30.00**
Produto [1]: **333**
Preço [1]: **180.00**
...
Produto [9]: **345**
Preço [9]: **50.00**

Qual o índice de desconto percentual a ser aplicado? **10.00** [10.00% de desconto]

Nova tabela de preços ao consumidor:

Produto [0]: **321**
Preço Original: R\$ **30.00** - Preço Reajustado: R\$ **27.00**
Produto [1]: **333**
Preço Original: R\$ **180.00** - Preço Reajustado: R\$ **162.00**
...
Produto [9]: **345**
Preço Original: R\$ **50.00** - Preço Reajustado: R\$ **45.00**

12. Considerando o programa anterior, de uma empresa está preparando o reajuste de preços de venda de seus produtos (10 produtos). Altere o programa de modo a criar uma tabela com a lista dos produtos vendidos pela empresa contendo: NOME do produto (string) e preço do produto. Exemplo:

Produto [0]: **Mouse com conector USB**
Preço [0]: **30.00**
Produto [1]: **WebCam Alta-Resolucao**
Preço [1]: **180.00**
...
Produto [9]: **PenDrive 4Gb**
Preço [9]: **50.00**

Exercícios – *JUST FOR FUN*:

13. Faça um simulador para avaliar a movimentação de passageiros do trem-bala de Campinas ao Rio de Janeiro. O trem realiza o trajeto Campinas-Rio com estações em: Campinas [Estação 0], São Paulo [Estação 1] São José dos Campos [Estação 2], Resende [Estação 3], Rio de Janeiro [Estação 4], com um total de 5 paradas. Em cada parada irão embarcar um certo número de passageiros e desembarcar outra quantidade de passageiros (controlados pelas catracas eletrônicas), podendo restar um certo número de passageiros a bordo. O programa deve coletar as informações de quantas pessoas entram e saem em cada uma das estações, realizando uma estatística final da movimentação de passageiros: total de passageiros transportados, movimento de passageiros por estação, e total de passageiros transportados por trecho. Os dados de movimentação de passageiros (embarque e desembarque) devem ser armazenados em vetores. Exemplo de uso do programa:

Trajetos Campinas-Rio 8h

Estação 0: Campinas

- Quantos passageiros embarcaram: **1234**

- Quantos passageiros desembarcaram: **0**

Estação 1: São Paulo

- Quantos passageiros embarcaram: **1234**

- Quantos passageiros desembarcaram: **0**

Estação 2: São José dos Campos

- Quantos passageiros embarcaram: **1234**

- Quantos passageiros desembarcaram: **0**

Estação 3: Resende

- Quantos passageiros embarcaram: **1234**

- Quantos passageiros desembarcaram: **0**

Estação 4: Rio de Janeiro

- Quantos passageiros embarcaram: **0**

- Quantos passageiros desembarcaram: **4936**

Total de passageiros transportados: **4936**

Circulação de passageiros por estação: (entra+sai)

Estação 0: **1234** Estação 1: **1234** Estação 2: **1234** Estação 3: **1234** Estação 4: **4936**

Passageiros transportados por estação: (dentro do trem ao sair da estação)

Estação 0: **1234** Estação 1: **2468** Estação 2: **3702** Estação 3: **4096** Estação 4: **0**

Diferença entre Embarque-Desembarque (problema no controle de passageiros): **0**



Fonte: <http://www.destakjournal.com.br/images/news/20090406/2482798.jpg>

14. Faça um programa que permita jogar um jogo do tipo “9-numbers puzzle” (baseado em um tabuleiro tamanho 3x3), inspirado no tabuleiro representado pela imagem mostrada logo abaixo.

O jogo deve apresentar ao jogador um tabuleiro na tela e pedir para que ele selecione uma peça para mover. A peça que é movida sempre troca de posição com o lugar vago. O jogo é iterado “n” rodadas até que o tabuleiro se encontre ordenado. Sugestão: use uma matriz 3x3.



Site exemplo: Versão mais complexa de 16 casas (4x4).

http://www.learn4good.com/games/puzzle/numeric_puzzle.htm

Lista complementares de exercícios disponível junto a Wiki ICMC da Disciplina SSC0501

<http://wiki.icmc.usp.br/index.php/SSC-501> - Listas de Exercícios
