

## 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. 06*

#### [VETORES e MATRIZES]

1. Faça um programa para ler uma tabela contendo os nomes dos alunos de uma turma de até 40 alunos. O programa deve solicitar ao usuário os nomes do aluno, sempre perguntando se ele deseja inserir mais um nome na lista. Uma vez lidos todos os alunos, o usuário irá indicar um nome que ele deseja verificar se está presente na lista, onde o programa deve procurar pelo nome (ou parte deste nome) e se encontrar deve exibir na tela o nome completo e o índice do vetor onde está guardado este nome.

```
>> Lista de Alunos da Turma <<

** Cadastrar alunos **
Digite o nome do aluno 1: Fulano da Silva
Deseja entrar com mais um nome (S/N)? s
Digite o nome do aluno 2: Beltrano de Oliveira
Deseja entrar com mais um nome (S/N)? n

** Procurar alunos**
Digite o nome do aluno procurado: Oliveira
Aluno encontrado – Aluno 2: Beltrano de Oliveira
```

- *Desafio:* encontre o nome independente da grafia (maiúsculo/minúsculo, com/sem acentos)

2. Faça um programa baseado no programa anterior, mas onde sejam lidos os seguintes dados para cada um dos alunos, seu nome e também o seu nro. USP e média geral de desempenho no curso. O programa deve solicitar ao usuário os nomes do aluno, seu nro. USP e sua média geral, sempre perguntando se ele deseja inserir mais um dado nesta lista. Uma vez lidos todos os dados dos alunos, o programa deverá exibir os dados do aluno que possui a maior média geral (exibir o nome completo, o nro. USP e a média geral deste aluno).

```
>> Desempenho dos Alunos <<

** Cadastrar alunos **
Digite o nome do aluno 1: Fulano da Silva
Digite o nro. USP do aluno 1: 1234567
Digite a média geral do aluno 1: 8.76
Deseja entrar com mais um nome (S/N)? n
** Melhor Aluno **
Dados do aluno - Nro. USP: 1234567 - Nome: Fulano da Silva - Média Geral: 8.76
```

3. Faça um programa para armazenar na memória uma tabela de 12x3 valores, onde estes dados correspondem aos valores da temperatura mínima, média e máxima de cada um dos meses do ano que passou. O programa deve solicitar ao usuário para fornecer as temperaturas mínima e máxima, calculando de modo automático o valor da temperatura média. Uma vez lidos todos os dados, exibir na tela a tabela contendo os dados mensais de temperatura: mínima, média e máxima.
4. Baseado no programa anterior, indicar qual foi a menor temperatura do ano, bem como qual foi a maior temperatura alcançada neste ano. Exibir na tela a mínima e máxima anual que se encontram armazenadas nesta tabela de temperaturas de um ano.
5. Faça um programa para ler 10 números DIFERENTES a serem armazenados em um vetor. Os dados deverão ser armazenados no vetor na ordem que forem sendo lidos, sendo que caso o usuário digite um número que já foi digitado anteriormente, o programa deverá pedir para ele digitar um outro número. Note que cada valor digitado pelo usuário deve ser pesquisado no vetor, verificando se ele existe entre os números que já foram fornecidos. Exibir na tela o vetor final que foi digitado.

```
>> Vetor de Núemros Diferentes <<
Digite o número 1: 15
Digite o número 2: 7
Digite o número 3: 12
Digite o número 4: 15
Número inválido! Digite um novo número diferente dos anteriores!
Digite o número 4: 12
Número inválido! Digite um novo número diferente dos anteriores!
Digite o número 4: 13
Digite o número 5: 19
...
Digite o número 10: 100
Números digitados:
15, 7, 12, 13, 19, ... 100
```

6. Baseado no programa anterior faça um programa para ler 25 números DIFERENTES a serem armazenados em uma tabela 5x5. O programa deve evitar que o usuário digite 2 números iguais para serem armazenados na tabela. Exibir na tela o conteúdo final da tabela que foi digitada.

```
>> Tabela digitada <<
Linha 1: 10, 15, 19, 12, 11
Linha 2: 21, 25, 28, 23, 20
Linha 3: 31, 32, 35, 37, 39
Linha 4: 47, 43, 42, 44, 46
Linha 5: 55, 56, 57, 58, 59
```

7. Faça um programa para gerar automaticamente números, entre 0 e 99, de uma cartela de bingo. Sabendo que cada cartela deverá conter 5 linhas de 5 números, gere estes dados de modo a **não ter números repetidos** dentro das cartelas! Lembre-se que é importante armazenar os números sorteados em uma tabela com 5 linhas e 5 colunas, pois no sorteio do bingo vale pontos quem completar primeiro uma linha ou uma coluna. O programa deve gerar a cartela e depois exibir na tela a cartela gerada, linha a linha, coluna por coluna.

8. Crie uma matriz para armazenar o tabuleiro do jogo da velha (3x3). Faça um programa que permita interagir com 2 jogadores, lendo suas jogadas e apresentando na tela a situação atual do tabuleiro. A partida será realizada entre os 2 jogadores, e o programa terá apenas as funções de:
- Ler a jogada de cada jogador. Exemplo: Jogar na posição 1 1 (Linha 1, Coluna 1);
  - A cada nova jogada deve ser exibido o estado do tabuleiro na tela, para que os jogadores possam ver a situação do jogo e fazer suas jogadas.
- Opcionalmente você pode:
- Verificar se a jogada é válida: não podemos jogar em uma posição onde já foi colocada uma peça, ou tentar jogar em uma casa fora do tabuleiro;
  - Verificar se o jogo acabou, ou seja, se tem algum vencedor (alguém completou linha, coluna ou diagonal) ou se o tabuleiro inteiro foi preenchido e não teve vencedores (empate);
- Para exibir o tabuleiro, use os caracteres de '0', 'X' (casas ocupadas) e um '?' para as casas livres.
9. Crie uma agenda de compromissos gerenciada pelo computador, ou seja, uma matriz que armazene a data e hora de um compromisso, assim como um texto que descreve o compromisso agendado. Cada compromisso deve possuir um mês, dia e hora (data) e mais um texto associado a este de até 30 caracteres (string) que o descreve. Inicialmente o usuário deve poder indicar a data e hora de seus compromissos, entrando um a um, e depois indicando se deseja informar mais um compromisso, até que ele responda que não deseja mais fornecer compromissos para sua agenda. O programa deve então permitir que o usuário consulte os compromissos agendados, lendo o dia, mês, hora, e exibindo na tela as informações sobre o compromisso agendado (repetir isto até que o usuário informe que não deseja mais consultar a agenda). **Dica:** Use uma matriz do tipo Agenda [mes][dia][hora] que armazene textos. Atenção este programa requer uma quantidade de memória grande:  $12 \times 31 \times 24 \times 30 = 267.840$  bytes (~ 260Kbytes).
10. 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: nome do produto e preço. 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:

```
>> Cadastro de Produtos e Precos <<
Produto [0]: Mouse Ótico USB Marca ACME
Preço [0]: 30.00
Produto [1]: DVD- R 100 unid Marca ACME
Preço [1]: 180.00
...
Produto [9]: Pen Drive USB 2Gb Marca ACME
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]: Mouse Ótico USB Marca ACME
Preço Original: R$ 30.00 - Preço Reajustado: R$ 27.00
Produto [1]: DVD-R 100 unid Marca ACME
Preço Original: R$ 180.00 - Preço Reajustado: R$ 162.00
...
Produto [9]: Pen Drive USB 2Gb Marca ACME
Preço Original: R$ 50.00 - Preço Reajustado: R$ 45.00
```

11. Faça um programa que realize a leitura dos seguintes dados relativos a um conjunto de alunos: Nro. USP, Nome, 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 - Nome [0] = Fulado da Silva
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 - Nome [1] = Ciclano de Oliveira
Aluno 2 - Cod. Disciplina [1] = 301
Aluno 2 - Nota1[2] = 6.0
...
```

12. Faça um programa que leia uma matriz 3x3 de inteiros e gere a sua transposta, exibindo na tela.
13. Faça um programa que leia uma matriz 3x3 de inteiros e calcule a soma de todos os seus elementos.
14. Faça um programa que leia duas matrizes 3x3 de inteiros e calcule a soma entre elas.
15. Faça um programa que leia uma matriz 3x3 de inteiros, calcule a soma dos elementos de cada coluna, armazenando o resultado da soma, em um vetor de 3 elementos. Faça o mesmo para as linhas da matriz. Exibir na tela os vetores que armazenam as somas das colunas e das linhas.
16. Faça um programa que leia uma matriz 3x3x3 de inteiros e calcule a soma de todos os seus elementos.

#### **Exercícios – QUESTÃO DESAFIO:**

17. Implemente uma Planilha inspirada no *Visicalc* (primeira planilha que depois deu origem ao Excel). Dicas: você pode ter uma planilha (tabela) para armazenar inteiros, outra para reais, outra para textos e outra para fórmulas, onde seria interessante ter uma tabela que indica o tipo de dado armazenado em cada célula (I/R/T/F). Assim você consegue “juntar” estes 4 tipos de dados em uma só planilha. Para implementar fórmulas, você pode ler os seguintes dados: operador (+,-,\*,/), a linha e coluna inicial da operação, assim como a linha e coluna final da operação. Depois basta aplicar esta operação nas células desta região da planilha.
- Referências: Wikipedia - <http://en.wikipedia.org/wiki/Visicalc>  
Dan Bricklin Visicalc - <http://www.danbricklin.com/history/vcexecutable.htm>