

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. 08

[Programas com Estrutura de Dados Compostas – typedef e struct]

1. Escreva um trecho de código em “C” para fazer a criação dos novos tipos de dados conforme solicitado abaixo:
 - A) Horário: composto de hora, minutos e segundos
 - B) Data: composto de dia, mês e ano
 - C) Compromisso: composto de uma data, horário e texto que descreve o compromisso
 - D) Byte: usado para armazenar 8 bits (definido em função dos tipos básicos do “C”)
2. Escreva um trecho de código em “C” para fazer a leitura dos dados (digitados pelo usuário) para os novos tipos de dados criados no exercício anterior: A) Horário; B) Data; C) Compromisso; D) Byte
3. Faça um programa que leia e armazene em um registro de dados as seguintes informações: Nome do Aluno, Nro. USP, Nota1, Nota2 e Média. Os dados devem ser digitados pelo usuário, armazenados na estrutura e exibidos na tela (leitura dos dados de apenas 1 aluno).
4. Faça um programa que armazene em um registro de dados (estrutura composta) os dados de um funcionário de uma empresa, compostos de: Nome, Idade, Sexo (M/F), CPF, Data de Nascimento, Código do Setor onde trabalha (0-99), Cargo que ocupa (string de até 30 caracteres) e Salário. Os dados devem ser digitados pelo usuário, armazenados na estrutura e exibidos na tela.
5. Baseado no programa 3 de leitura dos dados de um aluno, faça um programa que leia e armazene em um registro os dados de um aluno com as seguintes informações: Nome do Aluno, Nro. USP, Nota1, Nota2 e Média. Os dados devem ser digitados pelo usuário, armazenados na estrutura de dados em memória e exibidos na tela. A seguir deve ser perguntado ao usuário se ele deseja adicionar estes dados em um arquivo em disco (“cadastro.txt”), onde caso o usuário confirme, todos os dados devem ser adicionados ao fim deste arquivo texto; caso contrário os dados são simplesmente descartados. O programa deve ter um laço de entrada de dados, seguido da pergunta “Adicionar dados no Arquivo (S/N)?”, e seguido da pergunta “Deseja fornecer mais dados (S/N)?”, onde caso o usuário responda que sim, o programa repete a entrada de novos dados, caso responda que não, o programa é encerrado.
6. Baseado no programa 3 de leitura dos dados de um aluno, faça um programa que armazene em memória os dados de toda uma turma, composta de até 5 alunos. Os dados devem ser armazenados em um vetor de estruturas compostas (nome, nro, n1, n2 e média). O programa deve ler os dados dos alunos (sem a média), armazenar no vetor, calcular a média simples das notas, e depois exibir na tela todos os dados de cada um dos alunos.

7. Baseado no programa anterior, faça um programa que leia, armazene e exiba os dados de toda uma turma, composta de até 50 alunos. A cada novo aluno cujos dados são informados pelo usuário, pergunte se o usuário deseja continuar a entrar novos dados, caso ele diga que sim, continue lendo os dados dos alunos até ter lido no máximo 50 dados de alunos.
8. Faça um programa que leia os dados de 50 alunos (ou até que o usuário diga que não deseja mais entrar dados), onde cada aluno possui as seguintes informações: Nome, Nro. USP, Nota1, Nota2. Calcule a média dos alunos usando uma média ponderada, onde a Nota1 tem peso 1 e a Nota2 tem peso 2. Armazene estes dados em um vetor em memória. Salve os dados lidos em um arquivo texto.
9. Faça um programa que leia os dados de 10 alunos (Nome, Nro. USP, Média Final), armazenando em um vetor. Uma vez lidos os dados, divida estes dados em 2 novos vetores, o vetor dos aprovados e o vetor dos reprovados, considerando a média mínima para a aprovação como sendo 5.0. Exibir na tela os dados do vetor de aprovados, seguido dos dados do vetor de reprovados.
10. Escolha um jogo de cartas, baseado em um “baralho tradicional” (cada carta tem seu naipe e seu valor) ou tipo “super trunfo” (cada carta possui um conjunto de atributos). Implemente a parte de distribuição (sorteio) de cartas para 2 jogadores, considerando que cada jogador irá receber 5 cartas. Exiba na tela as cartas que cada um dos jogadores recebeu.

Exercícios – QUESTÃO DESAFIO:

1. Questão Desafio: Faça a implementação de um jogo completo de cartas, do tipo “batalha”. Inicialmente cada jogador recebe um conjunto de cartas (escolha algo entre 5 e 10 cartas), e depois são realizadas sucessivas batalhas onde cada jogador coloca em jogo uma das suas cartas seguindo a ordem em que foram recebidas (carta contra carta, uma de cada jogador). A cada nova batalha o jogador que possuir a carta mais alta fica com ambas as cartas jogadas. O jogo termina quando um dos jogadores não tiver mais cartas na mão. Você pode adaptar o jogo e as regras de acordo com as suas preferências.
2. Questão Desafio: Faça um editor gráfico, inspirado no PowerPoint ou Paint, que permita ao usuário editar um gráfico composto pelas seguintes primitivas gráficas: ponto, reta, círculo, retângulo ou texto. Cada primitiva possui seus próprios parâmetros: ponto (x, y, cor), reta (xini, yini, xfim, yfim, cor), círculo (x, y, raio, cor), retângulo ((xini, yini, xfim, yfim, cor), texto (x, y, mensagem_de_texto). A partir da definição desta primitivas, faça um programa que permita que seja fornecido pelo usuário uma coleção de primitivas a serem armazenadas em uma estrutura de dados em memória: o programa deve consistir de um laço onde é perguntado ao usuário o tipo da primitiva (p=ponto, r=reta, c=círculo, b=box, t=texto) e seus parâmetros, e depois perguntando se deseja inserir mais primitivas ou sair do programa. Uma vez terminada a inserção das primitivas, você deve salvar em um arquivo texto todos os dados digitados.
3. Questão Desafio: adapte o programa da questão desafio 2 para que se torne uma ferramenta interativa, de modo que as primitivas possam ser criadas e visualizadas a medida que vão sendo fornecidas pelo usuário, usando o teclado e o mouse para entrar as informações das coordenadas de cada primitiva. Para implementar este programa use a biblioteca de rotinas gráficas WinBGI. Informações sobre a WinBGI podem ser obtidas na seguinte página web: <http://coteia.icmc.usp.br/mostra.php?ident=574.6>