

SME0230 - Introdução à Programação de Computadores

Primeiro semestre de 2010

Professora: Marina Andretta (andretta@icmc.usp.br)

Estagiário PAE: Mayron César de Oliveira Moreira (mayron@icmc.usp.br)

Monitor: Luiz Carlos Lucca (lluca@icmc.usp.br)

Forma de entrega: O exercício desta aula prática deverá ser entregue por e-mail para andretta@icmc.usp.br, **com cópia para** mayron@icmc.usp.br e lluca@icmc.usp.br.

Como se trata de **três arquivos** fonte da linguagem C, os **nomes dos arquivos** deverão ser: **ipc_ap8_númeroUsp_1.c**, **ipc_ap8_númeroUsp_2.c** e **ipc_ap8_númeroUsp_3.c** (referentes ao primeiro, segundo e terceiro exercícios, respectivamente) . Exemplo: ipc_ap8_6711776_1.c; ipc_ap8_6711776_2.c; ipc_ap8_6711776_3.c.

O título da mensagem será: ipc_ap8_númeroUsp. Exemplo: ipc_ap8_6711776.

Atenção: Mandar **apenas 1 email** com os **três arquivos** de código em **anexo**.

Observações importantes:

1. Trabalhos entregues após a 23h59 do dia 20/05 terão 10% a menos na nota;
2. O aluno que enviar o trabalho fora do padrão ou que não enviar para os e-mails especificados no enunciado terá 20% a menos da nota referente ao exercício desta aula. Caso na próxima aula o aluno entregue o trabalho novamente fora do padrão ou não envie para os e-mails especificados no enunciado, o mesmo receberá zero como nota do referente exercício da aula prática.

Aula Prática 8

1-) Escreva um programa utilizando estruturas (*structs*) para representar um aluno da disciplina IPC. Esse aluno contém: um nome limitado em 30 caracteres, um número USP, três notas e um número de faltas. Seu programa deve ter duas funções: a primeira delas deve imprimir os dados do aluno passado como parâmetro; a segunda deve retornar 1 se o aluno foi aprovado, 0 se o aluno ficou de recuperação e -1 se ele foi reprovado. A seguir, seguem os cabeçalhos sugeridos para essas funções:

```
void imprimeAluno(Aluno a);
```

```
int aprovado(Aluno a);
```

Tome como **entrada** para seu programa os seguintes dados (**use a função scanf**):

Aluno1

Nome: Mayron Moreira

Número USP: 6711776

Nota1: 6.7

Nota2: 7.1
Nota3: 10
Número de faltas: 2
Aluno2
Nome: Bill Gates
Número USP: 3455467
Nota1: 10
Nota2: 10
Nota3: 10
Número de faltas: 0
Aluno3
Nome: George Dantzig
Número USP: 1116789
Nota1: 3.9
Nota2: 3.9
Nota3: 1.6
Número de faltas: 2
Aluno4
Nome: John Nash
Número USP: 5690087
Nota1: 9.7
Nota2: 9.3
Nota3: 8.4
Número de faltas: 25

Saída do programa:

Alunos aprovados em IPC:

Nome: Mayron Moreira
Número USP: 6711776
Nota1: 6.7
Nota2: 7.1
Nota3: 10.0
Número de faltas: 2
Média: 7.9

Nome: Bill Gates
Número USP: 3455467
Nota1: 10.0
Nota2: 10.0
Nota3: 10.0
Número de faltas: 0
Média: 10.0

Alunos de recuperação em IPC:

Nome: George Dantzig
Número USP: 1116789
Nota1: 3.9
Nota2: 3.9
Nota3: 1.6
Número de faltas: 2
Média: 3.1

Alunos reprovados em IPC:

Nome: John Nash
Número USP: 5690087
Nota1: 9.7
Nota2: 9.3
Nota3: 8.4
Número de faltas: 25
Média: 9.1

Obs.: 1) Para representar as notas na estrutura, **utilize um vetor de três posições.**

2) Se um aluno tiver mais que 25% de faltas na disciplina, o mesmo é considerado reprovado. Suponha que IPC tenha 32 aulas no semestre.

3) As notas dos alunos devem ser valores reais estando no intervalo de 0 a 10.

2-) Sua tarefa consiste em escrever um programa que leia um número inteiro n tal que $n \in [0,32767]$ e imprimir sua representação binária de 16 bits. Para tanto, faça uma função que receba esse valor e imprima o número binário correspondente. Sugestão para o cabeçalho da função:

void imprimeBinario16 (**int** n);

Tome como **entrada** para seu programa os seguintes valores de n (**NÃO use a função scanf, insira esses dados de entrada diretamente no programa**):

3, 5, 40, 127, 321, 10205, 32767

Saída do programa:

```
0000000000000011
0000000000000101
0000000000101000
0000000001111111
0000000101000001
```

0010011111011101
0111111111111111

3-) Os elementos a_{ij} de uma matriz inteira $A_{n \times n}$ representam os custos de transporte da cidade i para a cidade j . Dados 4 itinerários, cada um com **5 cidades**, faça um programa que calcule o custo para cada caminho e imprima. Para isso, utilize a função `calculaCaminho`, que retorna o custo do caminho passado como parâmetro. O cabeçalho da função é sugerido abaixo:

int calculaCaminho (Caminho c);

Note que será necessária a criação de uma estrutura `Caminho`, que possui um vetor representando o número de cidades do itinerário.

Obs.: As cidades que **NÃO** possuem ligação tem custo igual a **-1**.

A =

-1	5	10	32	-1
65	-1	-1	4	19
10	15	-1	40	43
30	-1	-1	-1	34
5	21	69	56	-1

Tome como **entrada** para seu programa os seguintes valores de rota (**NÃO** use a função `scanf`, insira esses dados de entrada diretamente no programa):

Rota1: 1-2-4-5-4

Rota2: 2-3-5-4-1

Rota3: 1-3-4-5-2

Rota4: 3-1-4-2-5

Saída do programa:

Custo1: 99

Custo2: itinerário inviável

Custo3: 105

Custo4: itinerário inviável