Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Sistemas de Computação
SSC502 – Laboratório ICC – Turma 2 – 1º semestre 2010
3a. Lista de Exercícios

1. Conta Bancária.
Considere a seguinte estrutura:

```
typedef struct
{
        int NumConta;
        char Cliente[100];
        float Saldo;
        int Senha;
        char Chave;
} contabancaria;
```

faça um programa com os requisitos:

a)Cria uma conta
b)Consulta o saldo do cliente (entra com o numero da conta conferindo apenas a senha)
c)Deposita um valor (entra com o numero da conta e confere o nome do cliente)
d)Saca um valor (entra com o numero da conta e confere senha e chave, o cliente tem apenas autorização de sacar o seu dinheiro, conta sem limite)
e)Encerre a conta (entra com o numero da conta e confere senha e chave, faça uma pergunta para que o cliente confira a operação e apague seus dados)

Obs.: Para cada item faça uma função.

2)Faça um programa que entre com um numero romano (de tamanho máximo 3999) e de como saída seu correspondente em decimal, use uma struct.

3)Seja o seguinte trecho de programa:

```
int i=3,j=5;
int *p, *q;
p = &i;
q = &j;
```

Qual é o valor das seguintes expressões ?

a) p == &i; b) *p - *q c) **&p d) 3* - *p/(*q)+ 7

Obs.: Faça na mão e depois execute para ver o resposta.

4)Qual a saída do programa?

```
#include <conio.h>
#include <stdio.h>
void main(){
        float vet[5] = {1.1,2.2,3.3,4.4,5.5};
        float *f;
        int i;
        f = vet;
        printf("contador/valor/valor/endereco/endereco");
        for(i = 0 ; i <= 4 ; i++)
        {
                printf("\ni = %d",i);
                printf(" vet[%d] = %.1f",i, vet[i]);
                printf(" *(f + %d) = %.1f",i, *(f+i));
                printf("&vet[%d] = %X",i, &vet[i]);
                printf(" (f + %d) = %X",i, f+i);
        }
}
```

Obs.: Faça na mão e depois execute para ver o resposta.

5)Faça um programa que abra 2 arquivo.c e conte os números de for, while, if, else if, else, case, do e o número de bibliotecas declaradas. Compare os valores, caso todos sejam iguais imprima a mensagem: possível plágio, caso contrário imprima os valores de ambos os arquivos.

6)Refaça o programa 1 buscando e salvando os dados do cliente em um arquivo.

7) Defina uma *struct* **compromisso**, que contenha informações como data (dia, mês e ano), horário e nome do compromisso. Logo após utilize a descrição de vetor agenda dada abaixo.

```
typedef struct compromisso{
.
.
.
};

compromisso[100] agenda;
```

a) Faca funções que permitam inserir e remover compromissos da **agenda**.

b) Faca uma função que permita ao usuário digitar uma data e o programa imprimir os compromissos da pessoa para aquele dia.

8) Utilizando o exercício anterior, crie um arquivo e faca as seguintes funções:

a) Uma função que salve a agenda toda no arquivo.
b) Uma função que recupere a agenda toda do arquivo.
c) Uma função que pede a data atual, exclui todos os compromissos anteriores a esta data do arquivo e salva as alterações.

10) Compare as duas funções abaixo. Qual o valor de *p* após a chamada de cada uma das funções, se *p* for inicializado com o valor 2?

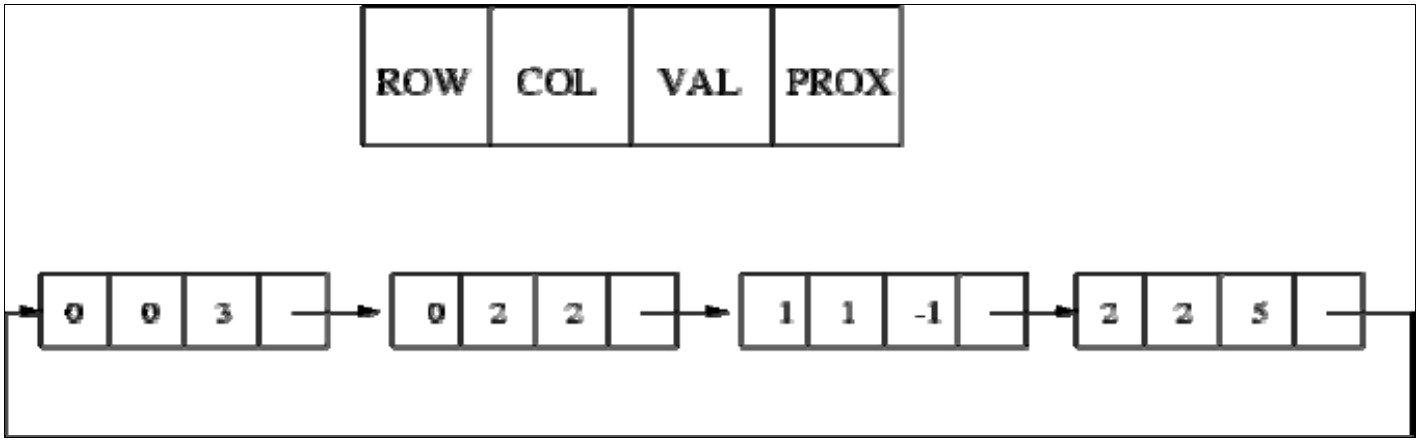
```
void funcao1(int p){
        p++;
}

void funcao1(int *p){
        (*p)++;
}
```

11) Uma matriz é dita **esparsa** quando a maioria de seus elementos é igual a zero. Podemos ver no exemplo abaixo que muito espaço em memória seria economizado se apenas os elementos não nulos fossem armazenados.

$$\begin{bmatrix} 0 & 0 & 7 & 0 & 0 & 11 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & -1 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Uma forma de representação das matrizes esparsas é por meio de listas encadeadas.



A *struct* responsável por armazenar a matriz esparsa na forma de lista encadeada deve conter para cada elemento, o valor da linha e da coluna de cada elemento, o valor do elemento e um ponteiro para o próximo elemento.

Referência: http://www.lcad.icmc.usp.br/~nonato/ED/Matrizes/node30.html

Sendo assim, faca:

a) Uma função que transforme uma matriz esparsa em uma lista encadeada. Teste com a matriz abaixo:

b) Uma função que receba linha e coluna e retorne o valor da matriz. (Usar a lista encadeada para achar o valor.)

c) Uma função que receba uma coluna e retorne a soma dos elementos da coluna.

12) Continuando o exercício anterior, faca:

a) Uma função que salve a lista encadeada em um arquivo.

b) Uma função que receba dois arquivos com listas encadeadas e gere um terceiro arquivo que contenha a lista encadeada referente à soma das duas matrizes esparsas.