

INF 1620 – P2 - 14/10/08	Questão 1
Nome:	
Matrícula:	Turma

Suponha que uma empresa aérea mantém um cadastro de aeroportos como um vetor de ponteiros para estruturas que contêm as seguintes informações:

1. Sigla: cadeia com até 3 caracteres;
2. Cidade: cadeia com até 50 caracteres;
3. País: cadeia com até 30 caracteres;
4. Taxa: um valor real;
5. Capacidade: um valor inteiro;

a) [Valor: 1,0 ponto] Defina uma estrutura em C com o nome de *aeroporto*, que tenha os campos apropriados para guardar todas as informações descritas anteriormente. Defina também um novo tipo de dados com o nome de *Aeroporto*, correspondendo a essa estrutura.

b) [Valor: 2,0 pontos] Escreva uma função que cria dinamicamente e preenche uma variável do tipo *Aeroporto*. Esta função deve receber como parâmetros as cadeias de caracteres *s*, contendo a sigla do aeroporto, *c* e *p*, contendo, respectivamente, a cidade e o país onde esse se localiza, o valor real *tx*, contendo a taxa cobrada por ele, e o inteiro *cap*, indicando sua capacidade; e retornar um ponteiro para a variável alocada dinamicamente. Se não for possível alocar uma variável, a função deve retornar NULL. A função deve ter o seguinte protótipo:

```
Aeroporto* Cadastra(char* s, char* c, char* p, float tx, int cap);
```

INF 1620 – P2 - 14/10/08	Questão 2
Nome:	
Matrícula:	Turma

Um programa de contabilidade usa uma matriz com  $m$  linhas e  $n$  colunas para representar a planilha de despesas de uma empresa. Nessa matriz, cada elemento contém um valor real referente a uma despesa, sendo que cada linha corresponde a um tipo de despesa e cada coluna corresponde à semana em que uma despesa foi registrada. A matriz é armazenada como um vetor unidimensional. Como exemplo, a tabela a seguir apresenta o registro de 4 tipos de despesas ao longo de 5 semanas.

	semana 1	semana 2	semana 3	semana 4	semana 5
<b>despesa A</b>	5.65	7.54	5.41	4.86	6.24
<b>despesa B</b>	145.45	134.52	124.52	135.63	159.86
<b>despesa C</b>	49.85	54.88	63.87	65.78	78.35
<b>despesa D</b>	81.29	89.62	75.64	87.89	77.55

a) [Valor: 2,0 pontos] Escreva uma função em C que calcule a média de um certo tipo de despesa durante um intervalo de algumas semanas. A função recebe como parâmetros a matriz de despesas (através de um ponteiro para o primeiro elemento do vetor que a armazena, seu número de linhas  $m$  e seu número de colunas  $n$ ), o índice de uma linha  $i$ , definindo o tipo de despesa de interesse, e os índices de duas colunas  $j$  e  $k$ , delimitando o intervalo fechado a ser considerado. A função tem o seguinte protótipo:

```
float media(float* despesas, int m, int n, int i, int j, int k);
```

b) [Valor: 2,0 pontos] Escreva um programa completo em C que crie (estaticamente) uma variável para armazenar uma matriz de despesas de dimensões 4 x 5 e:

1. inicialize essa matriz com os valores do exemplo acima;
2. chame a função *media* para calcular a média de despesa B considerando o intervalo fechado entre as semanas 3 e 5;
3. escreva esse resultado na tela.

INF 1620 – P2 - 14/10/08	Questão 3
Nome:	
Matrícula:	Turma

[Valor: 3,0 pontos] Um programa que administra uma agenda de compromissos pessoais mantém as informações referentes aos eventos em um vetor de ponteiros para dados estruturados do tipo *Evento*, definido conforme descrito a seguir:

```
struct evento
{
    char nome[81];
    Datahora inicio;
    Datahora final;
};
typedef struct evento Evento;
```

Onde o *nome* é uma *string* que contém o nome que descreve o evento e os campos *inicio* e *final* contêm dados do tipo *Datahora* que descrevem, respectivamente, a data e hora do início e final do evento. Esse tipo de dados é definido contendo os campos *dia*, *mês*, *hora* e *minuto* definidos como inteiros, conforme descrito a seguir:

```
struct datahora
{
    int dia;
    int mes;
    int hora;
    int minuto;
};
typedef struct datahora Datahora;
```

Usando uma das técnicas de ordenação apresentadas no curso, escreva uma função que recebe como parâmetros um vetor de ponteiros para eventos agendados (através de um ponteiro para seu primeiro elemento e um inteiro *n* indicando seu número de elementos) e coloque o vetor em ordem cronológica de acordo com o início dos eventos. Considere que a agenda não contém dois eventos começando exatamente no mesmo instante e que todos os eventos ocorrem num mesmo ano. A função tem o seguinte protótipo:

```
void Cronologica(Evento** agenda, int n);
```

# RASCUNHO

*Respostas nesta folha não serão consideradas.*

## Protótipos de funções que podem ser úteis:

### **stdio.h:**

```
int scanf (char* formato, ...);
int printf (char* formato, ...);
FILE* fopen (char* nome, char* modo);
int fclose (FILE* fp);
int fscanf (FILE* fp, char* formato, ...);
int fprintf (FILE* fp, char* formato, ...);
char* fgets(char* str, int size, FILE* fp);
int sscanf(char* str, char* formato, ...);
```

### **math.h:**

```
double sqrt (double x);
double pow (double x, double exp);
double cos (double radianos);
double sin (double radianos);
```

### **string.h:**

```
int strlen (char* s);
int strcmp (char* s, char *t);
char* strcpy (char* destino, char* fonte);
char* strcat (char* destino, char* fonte);
```

### **stdlib.h:**

```
void* malloc (int nbytes);
void free (void* p);
void qsort (void *vet, int n, int tam, int (*comp) (const void*, const void*));
```

*Não separe as folhas deste caderno.*