INF 1007 – P2B - 31/10/09	Questão 1
Nome:	Estação:
Matrícula:	Turma:

Uma grande empresa do setor financeiro possui agências espalhadas por diversas localidades. Esta empresa é estruturada de forma que, em uma dada localidade, existe uma única agência. Além disso, há uma forte hierarquia administrativa, de tal modo que agências menores respondem sempre a uma agência maior, com exceção da agência matriz, que não responde a nenhuma outra. O sistema administrativo desta empresa mantém as informações referentes às agências em um vetor de ponteiros para dados estruturados do tipo *Agencia*, conforme descrito a seguir:

```
struct agencia {
   int codigo;            /*  codigo da agencia                  */
   char localidade[31];   /*  localizacao da agencia             */
   char responsavel[31];  /*  localizacao da agencia responsavel  */
};
typedef struct agencia Agencia;
```

a) [Valor: 2,0 pontos] Escreva uma função em C que recebe como parâmetros o ponteiro *vet*, para o primeiro elemento um vetor de ponteiros para *Agencia*, o inteiro *n*, indicando número de elementos desse vetor, e a cadeia de caracteres *local*, representando o nome de uma localidade, e retorna o índice no vetor da agência que fica nesta mesma localidade, se existir, ou -1, caso contrário. A função deve ter o seguinte protótipo:

```
int Indice_agencia(Agencia** vet, int n, char* local);
```

b) [Valor: 3,0 pontos] Escreva uma função em C que recebe como parâmetros o ponteiro *vet*, para um vetor de ponteiros para *Agencia*, o inteiro *n*, indicando o número de elementos desse vetor, e duas cadeias de caracteres *local1* e *local2*, indicando duas localidades. A função deve retornar 1 se a agência de *local1* estiver subordinada administrativamente à agência de *local2*, isto é, se o campo *responsavel* da agência de *local1* for igual a *local2* ou se a agência responsável pela agência de *local1* estiver subordinada administrativamente à agência de *local2*. A função deve retornar 0, caso contrário. A agência matriz, que não tem agência responsável, tem uma string vazia (“”) no campo *responsavel*. Por exemplo, se o vetor aponta para as agências “Tijuca” (r*esponsavel* = “Centro”), “Olaria” (r*esponsavel* = “Tijuca”), “Centro” (r*esponsavel* = “Brasilia”), “Meier” (r*esponsavel* = “Tijuca”) e “Brasilia” (r*esponsavel* = “”), a agência “Olaria” está subordinada administrativamente à agência “Centro”, mas não à agência “Meier”. Neste exemplo, todos estão subordinados administrativamente à agência “Brasilia”. Considere que os dados fornecidos como parâmetros são sempre válidos, ou seja, *local1* sempre existe no vetor, o vetor tem pelo menos um elemento e *n* apresenta o valor correto. A função deve ter o seguinte protótipo:

```
int Subordinada(Agencia** vet, int n, char* local1,
                char* local2);
```

Obs.: A função *Indice_agencia*, implementada no item *a*, pode ser utilizada pela função *Subordinada*.

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

INF 1007 – P2B - 31/10/09	Questão 2
Nome:	Estação:
Matrícula:	Turma:

Considere que no mesmo sistema da questão anterior, as informações sobre os funcionários das agências estão em um vetor de ponteiros para dados do tipo *Funcionario*, definido conforme o tipo estruturado a seguir:

```
struct funcionario {
   int mat;          /*  matricula do funcionario           */
   char nome[51];    /*  nome do funcionario                */
   Data admissao;    /*  data de admissão do funcionario    */
};
typedef struct funcionario Funcionario;
```

Onde o campo *mat* é um inteiro que armazena a matrícula do funcionário, o campo *nome* é uma string que armazena o nome do funcionário e o campo *admissao* é do tipo *Data*, descrevendo o dia, mês e ano da admissão de um funcionário, conforme o tipo estruturado descrito a seguir:

```
struct data {
   int dia, mes, ano;
};
typedef struct data Data;
```

a) [Valor: 2,0 pontos] Escreva uma função que cria uma nova variável do tipo *Funcionario*. A função deve alocar memória dinamicamente para armazenar a variável, preencher os campos *mat*, *nome* e *admissao* e retornar um ponteiro para esta variável. O parâmetro *mat* corresponde à matrícula do funcionário, o parâmetro *nome* corresponde ao nome do funcionário e *d*, *m* e *a* , correspondem ao dia mês e ano de sua admissão, respectivamente. O protótipo da função é:

```
Funcionario* Cria_funcionario(int mat, char* nome, int d,
                              int m, int a);
```

b) [Valor: 3,0 pontos] Usando uma das técnicas de ordenação apresentadas no curso, escreva uma função que recebe como parâmetros um vetor de ponteiros para *Funcionario*, através do ponteiro *vet*, para seu primeiro elemento, e um inteiro *n*, indicando seu número de elementos, e coloque o vetor em ordem cronológica inversa de acordo com as datas de admissão. No caso de funcionários admitidos exatamente no mesmo dia, o desempate deve ser pela ordem crescente de matrícula. O protótipo da função é:

```
void Ordena_funcionarios(Funcionario** vet, int n);
```

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

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.