

INF 1007 – P2A - 31/10/09	Questão 1
Nome:	Estação:
Matrícula:	Turma:

Um sistema de administração de pessoal mantém as informações referentes aos funcionários de uma empresa em um vetor de ponteiros para dados estruturados do tipo *Funcionario*, conforme descrito a seguir:

```
struct funcionario {
    int mat;          /* matricula do funcionario      */
    char nome[51];   /* nome do funcionario          */
    int sup;         /* matricula do superior imediato */
};
typedef struct funcionario Funcionario;
```

a) [Valor: 2,0 pontos] Escreva uma função em C que recebe como parâmetros um vetor de ponteiros para *Funcionario*, através do ponteiro *vet*, para seu primeiro elemento, e do inteiro *n*, indicando seu número de elementos, e um inteiro *mat*, representando a matrícula de um funcionário, e retorna o índice no vetor do funcionário correspondente, ou -1, se este não for encontrado. A função deve ter o seguinte protótipo:

```
int Indice_funcionario(Funcionario** vet, int n, int mat);
```

b) [Valor: 3,0 pontos] Escreva uma função em C que recebe como parâmetros um vetor de ponteiros para *Funcionario*, através do ponteiro *vet*, para seu primeiro elemento, e do inteiro *n*, indicando seu número de elementos, e dois inteiros *mat1* e *mat2*, indicando duas matrículas de funcionários. A função deve retornar 1 se o funcionário de matrícula *mat1* estiver subordinado hierarquicamente ao funcionário de matrícula *mat2*, isto é, se o superior imediato (campo *sup*) do primeiro funcionário (*mat1*) for o segundo funcionário (*mat2*) ou se o superior imediato (campo *sup*) do primeiro funcionário (*mat1*) estiver subordinado hierarquicamente ao segundo funcionário (*mat2*). A função deve retornar 0, caso contrário. Um funcionário no topo da hierarquia, isto é, que não tem um superior imediato, tem o valor -1 no campo *sup*. Por exemplo, se o vetor aponta para as funcionárias Luiza (*mat* = 123 e *sup* = 125), Diana (*mat* = 124 e *sup* = 129), Gina (*mat* = 125 e *sup* = 126), Dolores (*mat* = 126 e *sup* = -1), Beatriz (*mat* = 128 e *sup* = 126) e Ana (*mat* = 129 e *sup* = 128), Diana está subordinada hierarquicamente a Beatriz, mas não a Gina. Neste exemplo, todos estão subordinados hierarquicamente a Dolores. Considere que os dados fornecidos como parâmetros são sempre válidos, ou seja, *mat1* 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 Superior_hierarquico(Funcionario** vet, int n,
    int mat1, int mat2);
```

Obs.: A função *Indice_funcionario*, implementada no item a, pode ser utilizada pela função *Superior_hierarquico*.

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 – P2A - 31/10/09	Questão 2
Nome:	Estação:
Matrícula:	Turma:

Considere que no mesmo sistema de administração de pessoal da questão anterior as informações sobre as licenças tiradas pelos funcionários estão em um vetor de ponteiros para dados do tipo *Licenca*, definido conforme o tipo estruturado a seguir:

```
struct licenca {
    char nome[51]; /* nome do funcionario */
    Data inicio; /* data de inicio da licenca */
    Data final; /* data de fim da licenca */
};
typedef struct licenca Licenca;
```

Onde os campos *inicio* e *final* contêm dados do tipo *Data*, que descrevem o dia, mês e ano do início e final de uma licença, respectivamente, 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 *Licenca*. A função deve alocar memória dinamicamente para armazenar a variável, preencher os campos *nome*, *inicio* e *final* e retornar um ponteiro para esta variável. O parâmetro *nome* corresponde à uma cadeia de caracteres com o nome do funcionário licenciado, e *d1*, *m1* e *a1* e *d2*, *m2* e *a2* correspondem ao dia mês e ano de início e final da licença, respectivamente. O protótipo da função é:

```
Licenca* Cria_licenca(char* nome, int d1, int m1, int a1,
                    int d2, int m2, int a2);
```

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 *Licenca*, através do ponteiro *vet*, para seu primeiro elemento, e um inteiro *n*, indicando seu número de elementos, e coloque o vetor em ordem cronológica de acordo com a data de início das licenças. No caso de licenças que começam exatamente no mesmo dia, deve prevalecer a ordem alfabética de acordo com o nome do funcionário. O protótipo da função é:

```
void Ordena_licencas(Licenca** 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.