INF 1620 - P3 - 23/06/07	Questão 1
Nome:	
Matrícula:	Turma

Considere a implementação de listas encadeadas para armazenar inteiros dada pelo tipo abaixo:

```
struct lista {
     int info;
     struct lista* prox;
     };
typedef struct lista Lista;
```

Escreva uma função que receba como entrada duas listas, L e M, e remova da lista L todos os elementos que também pertençam à lista M, devolvendo a lista L modificada. A função deve obedecer ao protótipo:

```
Lista* dif (Lista* L, Lista* M);
```

Assuma que:

1. Em cada uma das listas não ocorrem elementos duplicados.
2. Cada uma das listas, ou ambas, podem ser vazias.

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 1620 - P3 - 23/06/07	Questão 2
Nome:	
Matrícula:	Turma

Considere uma tabela de dispersão T, com tratamento de colisões por listas, que contém o número de matrícula e a nota dos alunos, implementada por:

```
struct elemento {
   char mat[7];
   float nota;
   struct elemento* prox;
};
typedef struct elemento Elemento;

#define N 127
typedef Elemento* Hash[N];
```

a) Escreva uma função de dispersão `hash` que seja adequada para a tabela acima, ou seja, a função:

   1. Deve ter a seguinte assinatura

      ```
      int hash(char* mat);
      ```

   2. Deve devolver um número inteiro entre 0 e 126.
   3. Deve minimizar a chance de colisões.

b) Escreva uma função que insira, numa tabela de dispersão T do tipo acima, as informações (número de matricula e a nota) de um(a) aluno(a). Por simplicidade, assuma que elementos duplicados não serão inseridos. A função deve obedecer ao protótipo:

   ```
   void insere (Hash T, char* mat, float nota);
   ```

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 1620 - P3 - 23/06/07	Questão 3
Nome:	
Matrícula:	Turma

Considere **árvores binárias de busca** em que cada nó segue o tipo definido por

```
struct arv {
     int info;
     int cont;
     struct arv* esq;
     struct arv* dir;
};
typedef struct arv Arv;
```

onde `info` armazena um valor inteiro x
`cont` armazena o número de vezes que x foi inserido em B

a) Escreva uma função que receba como entrada um apontador para a raiz de uma árvore binária de busca B do tipo acima e um inteiro x e:

1. Pesquise se x ocorre em B.
2. Se x ocorrer em um nó N de B, incremente o campo `cont` de N.

A função deve obedecer ao seguinte protótipo:

```
Arv* incr(int x, Arv* B);
```

b) Modifique a função do item (a) de tal forma que:

1. Pesquise se x ocorre em B.
2. Se x ocorrer em um nó N de B, incremente o campo `cont` de N.
3. Se x não ocorrer em B, crie um novo nó M em B, inicializando o campo `info` com x e o campo `cont` com 1. **O novo nó deve ser criado de tal forma que B seja uma arvore binária de busca.**

A função deve obedecer ao seguinte protótipo:

```
Arv* carrega(int x, Arv* B);
```

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 1620 - P3 - 23/06/07	Questão 4
Nome:	
Matrícula:	Turma

Considere uma árvore genérica representando o organograma de uma empresa, onde:

1. Cada nó da árvore representa um cargo e o funcionário que o ocupa.
2. As subárvore de um nó representam os cargo a ele subordinados.
3. Não há dois nós da árvore com o mesmo código de cargo.

Cada nó da árvore segue o tipo definido por:

```
struct arvv {
     int cod;              /* o codigo de um cargo */
     int matr; /* matr. do func. ocupando o cargo */
     struct arvv* prim;          /* primeiro filho */
     struct arvv* prox;           /* proximo irmao */
 };
typedef struct arvv Arvv;
```

Construa uma função que substitua uma pessoa em um cargo cujo protótipo é:

```
int troca (Arvv* O, int C, int M);
```

onde O é o organograma, C o código do cargo e M a matrícula do funcionário que vai ocupar o cargo. Caso o código não esteja no organograma a função não faz nada e retorna 0. Caso contrário, além de fazer a mudança a função deve retornar 1.

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 fgetc (FILE* fp);
int fputc (int c, FILE* fp);
int sscanf(char* str, char* formato, ...);
```

math.h:

```
double sqrt (double x);
double pow (double x, double exp);
```

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*));
void* bsearch (void* info, void *vet, int n, int tam,
               int (*comp) (const void*, const void*));
```

Não separe as folhas deste caderno.