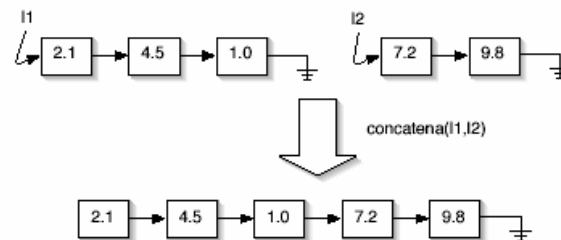


INF 1620 – P2 - 22/05/04	Questão 1
Nome:	
Matrícula:	Turma

Considere estruturas de listas encadeadas que armazenam valores reais. O tipo que representa um nó da lista é dado por:

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

Implemente uma função que, dadas duas listas encadeadas l1 e l2, concatene a lista l2 no final da lista l1, conforme ilustra a figura abaixo.



A função deve retornar a lista resultante da concatenação, obedecendo ao protótipo:

```
Lista* concatena (Lista* l1, Lista* l2);
```

Observe que l1 e/ou l2 podem ser listas vazias.

INF 1620 – P2 - 22/05/04	Questão 2
Nome:	
Matrícula:	Turma

Considere estruturas de listas encadeadas que armazenam valores inteiros. O tipo que representa um nó da lista é dado por:

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

Implemente uma função que receba um vetor de valores inteiros com n elementos e construa uma lista encadeada armazenando os elementos do vetor nos nós da lista. Assim, se for recebido o vetor $v[5] = \{3, 8, 1, 7, 2\}$, a função deve retornar uma nova lista cujo primeiro nó tem a informação 3, o segundo a informação 8, e assim por diante. Se o vetor tiver zero elementos, a função deve ter como valor de retorno uma lista vazia. O protótipo da função é dado por:

```
Lista* constroi (int n, int* v);
```

INF 1620 – P2 - 22/05/04	Questão 3
Nome:	
Matrícula:	Turma

Considere a existência de um tipo abstrato `Pilha` de números reais, cuja interface é definida pelo arquivo `pilha.h` com o seguinte conteúdo:

```
typedef struct pilha Pilha;

Pilha* cria (void);           /* cria uma pilha vazia */
void push (Pilha* p, float v); /* insere um elemento no topo */
float pop (Pilha* p);        /* retira o elemento do topo */
int vazia (Pilha* p);        /* retorna se está vazia */
void libera (Pilha* p);      /* libera a pilha */
```

Sem conhecer a representação interna deste tipo abstrato e usando apenas as funções declaradas no arquivo `pilha.h`, implemente uma função que retorne o número de elementos armazenados numa pilha dada. Ao final da função, a pilha deve ter o mesmo conteúdo original. Esta função deve obedecer ao seguinte protótipo:

```
int num_elem (Pilha* p);
```

Dica: use uma pilha auxiliar para transferir os valores da pilha original ou faça uma implementação recursiva.

INF 1620 – P2 - 22/05/04	Questão 4
Nome:	
Matrícula:	Turma

Considerando uma estrutura de árvore binária que armazena valores inteiros definida pelo tipo abaixo:

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

Escreva uma função que retorne a soma dos números inteiros armazenados em nós que são folhas da árvore (isto é, nós que não têm filhos). O protótipo da função deve ser:

```
int soma_info_folhas (Arv* a);
```

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 – P2 - 22/05/04	Questão 5
Nome:	
Matrícula:	Turma

Considerando a estrutura de árvore genérica que armazena valores inteiros definida pelo tipo abaixo:

```
struct arvgen {
    int info;
    struct arvgen *prim; /* ponteiro p/ primeiro filho */
    struct arvgen *prox; /* ponteiro p/ irmão */
};
typedef struct arvgen ArvGen;
```

Escreva uma função que retorne o número de nós existentes na árvore que tem a informação associada maior que um dado valor x . A função deve seguir o protótipo:

```
int num_nos_x (ArvGen* a, int x);
```

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, ...);
```

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);
```

Não separe as folhas deste caderno.