

EXERCÍCIOS LISTAS

- 1) O tipo Lista definido a seguir pode armazenar uma lista de caracteres de tamanho no máximo igual a 80. Os caracteres são armazenados em um array a partir da posição 1. O campo tamanho guarda o comprimento da lista.

```
#define fantasma '$'
#define Maximo 81
typedef struct {
    int tamanho;
    char vetor[Maximo];
}Lista;
```

Escreva a implementação de cada uma das funções a seguir:

- a) Lista criarLista();
- b) Lista esvaziarLista(Lista);
- c) int obterTamanho(Lista); //devolve o comprimento da lista
- d) bool verificarCheia(Lista);
- e) bool verificarListaVazia(Lista);
- f) char obterElemento(Lista, int);
- g) Lista inserir(Lista, char);
- h) int buscarElemento(Lista, char);

- 2) O tipo ListaR definido na biblioteca TipoReal.h permite o processamento de uma lista de números reais positivos. A interface oferecida contém as seguintes funções:

```
void criarListaVazia(ListaR *); // construtor
void construirLista(ListaR *, int); // construtor
int obterTamanho(ListaR *); // acesso
bool verificarListaVazia(ListaR *); // acesso
double obterElemento(ListaR *, int); // acesso
void inserir(ListaR *,double); // insere no final da lista
```

Complete diretamente na função main dada a seguir, a implementação de um algoritmo para calcular a média aritmética simples dos valores guardados em uma lista.

```
#include "TipoReal.h"
int main(){
    ListaR A;
    int n;
    double media;
    n = obterNum();
    construirLista(&A, n); // pré-condição:  $n \geq 0$ 
```

completar com o cálculo da média

```
printf("media = %f \n",media);
return 0;
}
```

EXERCÍCIOS LISTAS

- 1) O tipo Lista definido a seguir pode armazenar uma lista de caracteres de tamanho no máximo igual a 80. Os caracteres são armazenados em um array a partir da posição 1. O campo tamanho guarda o comprimento da lista.

```
#define fantasma '$'
#define Maximo 81
typedef struct {
    int tamanho;
    char vetor[Maximo];
}Lista;
```

Escreva a implementação de cada uma das funções a seguir:

- a) void criarLista(Lista *);
- b) void esvaziarLista(Lista *);
- c) int obterTamanho(Lista *);
- d) bool verificarCheia(Lista *);
- e) bool verificarListaVazia(Lista *);
- f) char obterElemento(Lista *, int);
- g) void inserir(Lista *, unsigned char);
- h) int buscarElemento(Lista *, char);

- 2) O tipo ListaF definido na biblioteca TipoReal.h permite o processamento de uma lista de números reais positivos. A interface oferecida contém as seguintes funções:

```
ListaF criarLista( );           // construtor
ListaF construirLista(int);     // construtor
int obterTamanho(ListaF);       // acesso
bool verificarVazia(ListaF);     // acesso
double obterElemento(ListaF, int); // acesso
ListaF inserirNoFim(ListaF, double); // insere no final da lista
```

Complete na função main dada a seguir, a implementação de um algoritmo para calcular a média aritmética simples dos valores guardados em uma lista.

```
#include "TipoReal.h"
int main(){
    ListaF B;
    int n;
    double media;
    n = obterNum();
    B = construirLista(n); // pré-condição: n ≥ 0

    completar com o cálculo da média

    printf("media = %f \n",media);
    return 0;
}
```