

Nome: BRUNO FREIRE PEREIRA EVANDRO
Matricula: 2022100471
Turma: 931 PROG ESTRUT 2022/2
Professor: SERGIO ASSUNÇÃO MONTEIRO
Curso: CIÊNCIA DA COMPUTAÇÃO

```
/*-----*/  
#include<stdio.h>  
#include <stdlib.h>  
/*-----*/  
  
struct ELista  
{  
    int elem;  
    struct ELista *prox;  
};  
  
struct EMonitor  
{  
    struct ELista*prim;  
    struct ELista*fim;  
    int qtde;  
};  
/*-----*/  
  
void create();  
void inserir(int elem);  
int menu();  
void imprimir();  
struct ELista *pesquisar(int chave);  
struct ELista *pesquisarAnterior(int chave,  
                                struct ELista **ant);  
  
int excluir(int elem);  
/*-----*/  
  
struct EMonitor monitor;  
/*-----*/
```

```

main()
{
    create();
    while(menu());
}

/*-----*/
void create()
{
    monitor.prim=NULL;
    monitor.fim=NULL;
    monitor.qtde=0;
}

/*-----*/
int menu()
{
    int opcao;
    system("cls");
    printf("\n Digite:");
    printf("\n  0. Para Sair ");
    printf("\n  1. Para Inserir ");
    printf("\n  2. Para Imprimir ");
    printf("\n  3. Para Pesquisar ");
    printf("\n  4. Para gerar dados ");
    printf("\n  5. Para Pesquisar Anterior ");
    printf("\n  6. Para Excluir ");
    printf("\n");
    scanf("%d",&opcao);
    switch(opcao)
    {
        case 0: return 0;
        case 1:

```

```

{ int elem;

printf("\nDigite o elemento para inserir na lista: ");

scanf("%d",&elem);

inserir(elem);

return 1;

}

case 2:

{ int elem;

imprimir();

printf("\nPressione qualquer tecla para continuar");

system("pause");

return 1;

}

case 3:

{ int elem;

struct ELista *p;

printf("\nDigite o elemento para pesquisar na lista: ");

scanf("%d",&elem);

p=pesquisar(elem);

if(p){

printf("Elemento Encontrado!\n");

}else{

printf("Elemento nao consta na Lista!\n");

}

printf("\nPressione qualquer tecla para continuar");

system("pause");

return 1;

}

case 4:

{

inserir(10);

```

```

    inserir(7);
    inserir(40);
    inserir(50);
    inserir(30);
    return 1;
}

case 5:
{ int elem;

  struct ELista *p;
  struct ELista *ant;
  printf("\nDigite o elemento para pesquisar na lista: ");
  scanf("%d",&elem);
  p=pesquisarAnterior(elem,&ant);
  if(p){
      printf("Elemento Encontrado!\n");
      if(ant){
          printf("O anterior eh: %i\n",ant->elem);
      }else{
          printf("O Elemento eh o primeiro da Lista!\n");
      }
  }

  }else{
      printf("Elemento nao consta na Lista!\n");
  }
  printf("\nPressione qualquer tecla para continuar");
  system("pause");
  return 1;
}

case 6:
{ int elem;

  printf("\nDigite o elemento que serah excluido da lista: ");

```

```

        scanf("%d",&elem);

        int f = excluir(elem);

        if(f==1){

            printf("Elemento excluido com sucesso!");

        }else{

            printf("Nao foi possivel excluir o elemento!");

        }

        return 1;

    }

    default:

        printf("\nDigite uma opcao VALIDA! ");

        return 1;

    }

}

/*-----*/

void inserir(int elem)

{

    struct ELista *p;

    p=(struct ELista *)malloc(sizeof(struct ELista));

    p->elem=elem;

    p->prox=NULL;

    monitor.qtde++;

    /*verifica se a lista eh vazia*/

    if(monitor.prim==NULL)

    {

        monitor.prim=monitor.fim=p;

        return;

    }

    /*inserir no Inicio da Lista*/

    if(p->elem<=monitor.prim->elem)

    {

```

```

        p->prox=monitor.prim;

        monitor.prim=p;

        return;

    }

    /*inserir no Final da Lista*/

    if(p->elem>=monitor.fim->elem)

    {

        monitor.fim->prox=p;

        monitor.fim=p;

        return;

    }

    /*Insercao Intermediaria. Existem, pelo menos, dois elementos*/

    struct ELista *aux;

    aux=monitor.prim;

    while((aux->prox->elem)<(p->elem))

        aux=aux->prox;

    p->prox=aux->prox;

    aux->prox=p;

}

/*-----*/

void imprimir()

{

    struct ELista *p;

    p=monitor.prim;

    while(p)

    {

        printf("\np->elem=[%d]",p->elem);

        p=p->prox;

    }

}

/*-----*/

```

```

struct ELista *pesquisar(int chave){
    struct ELista *p;
    p=monitor.prim;
    while(p)
    {
        if(p->elem==chave){
            return p;
        }
        p=p->prox;
    }
    return NULL;
}
/*-----*/

```

```

struct ELista *pesquisarAnterior(int chave,
                                   struct ELista **ant){

    struct ELista *p;
    p=monitor.prim;
    *ant=NULL;
    while(p)
    {
        if(p->elem==chave){
            return p;
        }
        *ant=p;
        p=p->prox;
    }
    return NULL;
}
/*-----*/

```

```

int excluir(int elem){
    struct ELista *p;

```

```

struct ELista *ant;

p=pesquisarAnterior(elem,&ant);
if(p==NULL){
    return 0;
}
monitor.qtde--;
//Quando existe apenas 1 elem na lista
if((p==monitor.prim)&&(p==monitor.fim)){
    free(p);
    monitor.prim=monitor.fim=NULL;
    return 1;
}
//Continua ...
}
/*-----*/

```