

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define TRUE 1
#define FALSE 0

typedef struct {
    char nome[31];
    int idade;
}Elem;

typedef struct {
    Elem *elementos;
    int quantidade;
    int tamanho;
}ListaEstatica;

int Criar(ListaEstatica *lista, int tam){
    lista->elementos = calloc(tam, sizeof(Elem));
    if (lista->elementos == NULL)
        return FALSE;
    lista->tamanho = tam;
    lista->quantidade = 0;
    return TRUE;
}

int Vazia(ListaEstatica lista){
    return (lista.quantidade == 0);
}

int Cheia(ListaEstatica lista){
    return (lista.quantidade == lista.tamanho);
}

int Quantidade(ListaEstatica lista){
    return (lista.quantidade);
}

int InserirInicio(ListaEstatica *lista, Elem novo){
    int i;

    if (Cheia(*lista)) return FALSE;
    for (i = lista->quantidade; i > 0; --i)
        lista->elementos[i] = lista->elementos[i - 1];
    lista->elementos[0] = novo;
    lista->quantidade++;
    return TRUE;
}

int InserirFim(ListaEstatica *lista, Elem novo){

```

```

        if (Cheia(*lista))
            return FALSE;
        lista->elementos[lista->quantidade++] = novo;
        return TRUE;
    }

void Imprimir(ListaEstatica lista){
    int i;
    for (i = 0; i < lista.quantidade; ++i){
        printf("Nome: %s - ", lista.elementos[i].nome);
        printf("Idade: %d\n", lista.elementos[i].idade);
    }
}

int Pesquisar(ListaEstatica lista, Elem elem){
    int i;
    for (i = 0; i < lista.quantidade; ++i)
        if ((strcmp(lista.elementos[i].nome, elem.nome) == 0) &&
            (lista.elementos[i].idade == elem.idade))
            return i;
    return i;
}

int RemoverInicio(ListaEstatica *lista, Elem *elem){
    int i;
    if (Vazia(*lista))
        return FALSE;
    strcpy(elem->nome, lista->elementos[0].nome);
    elem->idade = lista->elementos[0].idade;
    --lista->quantidade;
    for (i = 0; i < lista->quantidade; ++i)
        lista->elementos[i] = lista->elementos[i + 1];
    return TRUE;
}

int RemoverFim(ListaEstatica *lista, Elem *elem){
    if (Vazia(*lista))
        return FALSE;
    strcpy(elem->nome, lista->elementos[lista->quantidade].nome);
    elem->idade = lista->elementos[lista->quantidade].idade;
    --lista->quantidade;
    return TRUE;
}

void Remover(ListaEstatica *lista, int pos){
    int i;
    for (i = pos; i < lista->quantidade; ++i)
        lista->elementos[i] = lista->elementos[i + 1];
    lista->quantidade--;
}

void main()

```

```

{
    int escolha=1;
    int retorno;
    int listaCriada = FALSE;
    ListaEstatica *lista;
    Elem e;

    while (escolha!=12){
        printf("\n\n ----- ");
        printf("\n 1 - Cria lista ");
        printf("\n 2 - A lista está vazia ");
        printf("\n 3 - A lista está cheia ");
        printf("\n 4 - Quantos elementos tem na lista ");
        printf("\n 5 - Iserir no início da lista");
        printf("\n 6 - Iserir no fim da lista ");
        printf("\n 7 - Imprimir lista ");
        printf("\n 8 - Verificar se elemeto peretence a lista ");
        printf("\n 9 - Remover do início da lista ");
        printf("\n 10 - Remover do fim da lista ");
        printf("\n 11 - Remover da lista ");

        printf("\n\n Escolha uma opcao: ");
        scanf("%d",&escolha);
        switch (escolha) {
            case 1:{
                system("@cls||clear"); // clrscr();
                if (listaCriada == TRUE){
                    printf("\n\n A lista já foi criada! ");
                }
                else{
                    retorno = Criar(lista,5);
                    if (retorno == TRUE){
                        printf("\n\n A lista foi criada! ");
                        listaCriada = TRUE;
                    }
                    else{
                        printf("\n\n A lista não foi criada! ");
                    }
                }
                break;
            }
            case 2:{
                system("@cls||clear"); // clrscr();
                if (listaCriada == TRUE){
                    retorno = Vazia(*lista);
                    if (retorno == TRUE){
                        printf("\n\n A lista está vazia! ");
                    }
                    else{
                        printf("\n\n A lista não está vazia! ");
                    }
                }
            }
        }
    }
}

```

```

        else{
            printf("\n\n A lista ainda não foi criada! ");
        }
        break;
    }
    case 3:{
        system("@cls||clear"); // clrscr();
        if (listaCriada == TRUE){
            retorno = Cheia(*lista);
            if (retorno == TRUE){
                printf("\n\n A lista está cheia! ");
            }
            else{
                printf("\n\n A lista não está cheia! ");
            }
        }
        else{
            printf("\n\n A lista ainda não foi criada! ");
        }
        break;
    }
    case 4:{
        system("@cls||clear"); // clrscr();
        if (listaCriada == TRUE){
            retorno = Quantidade(*lista);
            printf("\n\n A quantidade de elementos na lista é %d ",
retorno);
        }
        else{
            printf("\n\n A lista ainda não foi criada! ");
        }
        break;
    }
    case 5:{
        system("@cls||clear"); // clrscr();
        if (listaCriada == TRUE){
            printf("\n\n Forneça o nome que deseja inserir ");
            scanf("\n\n %s",e.nome);
            printf("\n\n Forneça a idade associada a este nome ");
            scanf("\n\n %d",&e.idade);
            retorno = InserirInicio(lista,e);
            if (retorno == TRUE){
                printf("\n\n Elemento inserido! ");
            }
            else{
                printf("\n\n Elemento não inserido! ");
            }
        }
        else{
            printf("\n\n A lista ainda não foi criada! ");
        }
        break;
    }

```

```

}
case 6:{
    system("@cls||clear"); // clrscr();
    if (listaCriada == TRUE){
        printf("\n\n Forneça o nome que deseja inserir ");
        scanf("\n\n %s",e.nome);
        printf("\n\n Forneça a idade associada a este nome ");
        scanf("\n\n %d",&e.idade);
        retorno = InserirFim(lista,e);
        if (retorno == TRUE){
            printf("\n\n Elemento inserido! ");
        }
        else{
            printf("\n\n Elemento não inserido! ");
        }
    }
    else{
        printf("\n\n A lista ainda não foi criada! ");
    }
    break;
}
case 7:{
    system("@cls||clear"); // clrscr();
    if (listaCriada == TRUE){
        Imprimir(*lista);
    }
    else{
        printf("\n\n A lista ainda não foi criada! ");
    }
    break;
}
case 8:{
    system("@cls||clear"); // clrscr();
    if (listaCriada == TRUE){
        printf("\n\n Forneça o nome que deseja pesquisar ");
        scanf("\n\n %s",e.nome);
        printf("\n\n Forneça a idade associada a este nome ");
        scanf("\n\n %d",&e.idade);
        retorno = Pesquisar(*lista,e);
        if ((retorno >= 0)&&(retorno <= lista->quantidade)){
            printf("\n\n Elemento encontrado ! ");
            printf("\n\n Nome: %s - ", e.nome);
        }
        else{
            printf("\n\n Elemento não encontrado ! ");
        }
    }
    else{
        printf("\n\n A lista não foi criada! ");
    }
    break;
}
}

```

```

case 9:{
    system("@cls||clear"); // clrscr();
    if (listaCriada == TRUE){
        retorno = RemoverInicio(lista,&e);
        if (retorno == TRUE){
            printf("\n\n Elemento %s removido !",e.nome );
        }
        else{
            printf("\n\n Elemento não removido ! ");
        }
    }
    else{
        printf("\n\n A lista não foi criada! ");
    }
    break;
}
case 10:{
    system("@cls||clear"); // clrscr();
    if (listaCriada == TRUE){
        retorno = RemoverFim(lista,&e);
        if (retorno == TRUE){
            printf("\n\n Elemento %s removido !",e.nome);
        }
        else{
            printf("\n\n Elemento não removido ! ");
        }
    }
    else{
        printf("\n\n A lista não foi criada! ");
    }
    break;
}
case 11:{
    system("@cls||clear"); // clrscr();
    if (listaCriada == TRUE){
        printf("\n\n Forneça o nome que deseja remover ");
        scanf("\n\n %s",e.nome);
        printf("\n\n Forneça a idade associada a este nome ");
        scanf("\n\n %d",&e.idade);
        retorno = Pesquisar(*lista,e);
        if ((retorno >= TRUE)&&(retorno <= lista->quantidade)){
            Remover(lista,retorno);
            printf("\n\n Elemento %s com idade %d removido !",
e.nome, e.idade);

        }
        else{
            printf("\n\n Elemento não encontrado ! ");
        }
    }
    else{
        printf("\n\n A lista não foi criada! ");
    }
}

```

```
        }
        break;
    }
    default:{
        system("@cls||clear"); // clrscr();
        if( escolha==12){
            continue;
        }
        printf("\n\n Nenhuma opcao foi escolhida ");
        break;
    }
}
}
```