

EXERCÍCIO 01

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <windows.h>

#define InicioArranjo 1
#define MaxTam 5
typedef int Apontador;

typedef struct
{
    int cod;
    int idade;
} TipoItem;

typedef struct
{
    TipoItem Item[MaxTam];
    Apontador Primeiro;
    Apontador Ultimo;
} TipoLista;

TipoLista Lista;
TipoItem x;
Apontador pos;

void FLVazia(TipoLista *Lista)
{
    Lista->Primeiro = InicioArranjo;
    Lista->Ultimo = Lista->Primeiro;
}

int Vazia(TipoLista Lista)
{
    return (Lista.Primeiro == Lista.Ultimo);
}

void Insere(TipoItem x, TipoLista *Lista)
{
    if (Lista->Ultimo > MaxTam)
        printf("\nA Lista está cheia\n");
}
```

```

    else
    {
        Lista->Item[Lista->Ultimo - 1] = x;
        Lista->Ultimo = Lista->Ultimo + 1;
    }
}

void Imprime(TipoLista Lista)
{
    printf("\n");
    for (int i = Lista.Primeiro - 1; i <= Lista.Ultimo - 2; i++)
    {
        printf("Cód: %d -- Idade: %d\n\n", Lista.Item[i].cod,
Lista.Item[i].idade);
    }
}

void Retira(Apontador pos, TipoLista *Lista, TipoItem *x)
{
    if (Vazia(*Lista) || pos >= Lista->Ultimo)
        printf("ERRO: Posição não existe\n");
    else
    {
        *x = Lista->Item[pos - 1];
        printf("O elemento retirado foi o de código %d\n", x->cod);
        Lista->Ultimo--;
        for (int i = pos; i < Lista->Ultimo - 1; i++)
            Lista->Item[i - 1] = Lista->Item[i];
    }
}

void RetiraPos(Apontador pos, TipoLista *Lista, TipoItem *x)
{
    if (Vazia(*Lista) || pos >= Lista->Ultimo)
        printf("ERRO: Posição não existe\n");
    else
    {
        *x = Lista->Item[pos];
        printf("O elemento retirado foi o de código %d\n", x->cod);
        Lista->Ultimo--;
        for (int i = pos; i < Lista->Ultimo - 1; i++)
            Lista->Item[i] = Lista->Item[i + 1];
    }
}

```

```

void Procura(TipoItem x, TipoLista Lista)
{
    if (Vazia(Lista))
    {
        printf("Lista está vazia\n");
        return;
    }
    int aux = 0;
    for (int i = Lista.Primeiro; i < Lista.Ultimo - 1; i++)
        if (Lista.Item[i].cod == x.cod)
            aux = 1;
    if (aux == 1)
        printf("Elemento encontrado\n");
    else
        printf("Elemento não encontrado\n");
}

void InserePrimeiraPosicao(TipoItem *x, TipoLista *Lista)
{
    if (Lista->Ultimo > MaxTam)
    {
        printf("\nA Lista está cheia\n");
        return;
    }
    else
    {
        Lista->Ultimo++;
        for (int i = Lista->Ultimo - 2; i > Lista->Primeiro - 1; i--)
            Lista->Item[i] = Lista->Item[i - 1];
        Lista->Item[Lista->Primeiro - 1] = *x;
    }
}

void InserePosEscUsu(TipoItem *x, TipoLista *Lista, Apontador pos)
{
    if (Lista->Ultimo > MaxTam)
    {
        printf("\nA Lista está cheia\n");
        return;
    }
    else
    {
        Lista->Ultimo++;
        for (int i = Lista->Ultimo - 2; i > pos - 1; i--)
            Lista->Item[i] = Lista->Item[i - 1];
    }
}

```

```

        Lista->Item[pos] = *x;
    }
}

int main()
{
    int opcao, resposta;
    SetConsoleOutputCP(CP_UTF8);
    FLVazia(&Lista);
    do
    {
        printf("\n\nMENU\n\n");
        printf("Digite a opção desejada:\n");
        printf("0 - Sair\n");
        printf("1 - Esvaziar Lista\n");
        printf("2 - Verificar se a lista está vazia\n");
        printf("3 - Inserir elemento na lista\n");
        printf("4 - Imprimir os elementos da lista\n");
        printf("5 - Retirar elemento da lista antes da posição
determinada\n");
        printf("6 - Retirar elemento da lista\n");
        printf("7 - Pesquisar elemento na lista\n");
        printf("8 - Inserir na primeira posição\n");
        printf("9 - Inserir um elemnto na posição determinada pelo
usuário\n");
        scanf("%d", &opcao);
        switch (opcao)
        {
            case 0:
            {
                printf("PROGRAMA ENCERRADO!\n");
                break;
            }
            case 1:
            {
                FLVazia(&Lista);
                printf("\nLISTA VAZIA\n");
                break;
            }
            case 2:
            {
                resposta = Vazia(Lista);
                if (resposta == 1)
                    printf("\n Lista está vazia\n");
                else

```

```
        printf("\n Lista não está vazia\n");
        break;
    }
    case 3:
    {
        printf("Digite o código\n");
        scanf("%d", &x.cod);
        printf("Digite a idade\n");
        scanf("%d", &x.idade);
        Insere(x, &Lista);
        break;
    }
    case 4:
    {
        Imprime(Lista);
        break;
    }
    case 5:
    {
        printf("Digite a posição\n");
        scanf("%d", &pos);
        Retira(pos, &Lista, &x);
        break;
    }
    case 6:
    {
        printf("Digite a posição\n");
        scanf("%d", &pos);
        RetiraPos(pos, &Lista, &x);
        break;
    }
    case 7:
    {
        printf("Digite o código do elemento\n");
        scanf("%d", &x.cod);
        Procura(x, Lista);
        break;
    }
    case 8:
    {
        printf("Digite o código\n");
        scanf("%d", &x.cod);
        printf("Digite a idade\n");
        scanf("%d", &x.idade);
        InserePrimeiraPosicao(&x, &Lista);
```

```

        break;
    }
    case 9:
    {
        printf("Digite o código\n");
        scanf("%d", &x.cod);
        printf("Digite a idade\n");
        scanf("%d", &x.idade);
        printf("Digite a posição\n");
        scanf("%d", &pos);
        InserePosEscUsu(&x, &Lista, pos);
        break;
    }
    default:
        printf("\n\nOpção inválida\n\n");
    }
    getch();
    system("cls");
} while (opcao != 0);

return 0;
}

```

EXERCÍCIO 02

```

#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <windows.h>

#define InicioArranjo 1
#define MaxTam 10
typedef int Apontador;

typedef struct
{
    int notas;
} TipoItem;

typedef struct
{
    TipoItem Item[MaxTam];
    Apontador Primeiro;
    Apontador Ultimo;
    int qtdeValores, ValAcimaMedia;
    float Media;
}

```

```

} TipoLista;

TipoLista Lista;
TipoItem x;
Apontador pos;

void FLVazia(TipoLista *Lista)
{
    Lista->Primeiro = InicioArranjo;
    Lista->Ultimo = Lista->Primeiro;
}

void Insere(TipoItem *x, TipoLista *Lista)
{
    if (Lista->Ultimo > MaxTam)
    {
        printf("\nA Lista está cheia\n");
        x->notas = -1;
        return;
    }
    Lista->Item[Lista->Ultimo - 1] = *x;
    Lista->Ultimo++;
}

void Calcula(TipoLista *Lista)
{
    Lista->Media = 0;
    Lista->qtdeValores = 0;
    Lista->ValAcimaMedia = 0;
    printf("\n");
    for (int i = Lista->Primeiro - 1; i <= Lista->Ultimo - 2; i++)
    {
        Lista->Media += Lista->Item[i].notas;
        Lista->qtdeValores++;
    }
    Lista->Media /= Lista->Ultimo - 1;
    for (int i = Lista->Primeiro - 1; i <= Lista->Ultimo - 2; i++)
        if (Lista->Item[i].notas > Lista->Media)
            Lista->ValAcimaMedia++;
}

void Imprime(TipoLista *Lista)
{
    printf("Quantidade de valores lidos: %d\n", Lista->qtdeValores);
    printf("Valores na ordem que foram informados:\n");
}

```

```

        for (int i = Lista->Primeiro - 1; i <= Lista->Ultimo - 2; i++)
        {
            printf("%d ", Lista->Item[i].notas);
        }
        printf("\nValores na ordem inversa:\n");
        for (int i = Lista->Ultimo - 2; i >= Lista->Primeiro - 1; i--)
        {
            printf("%d\n", Lista->Item[i].notas);
        }
        printf("Média dos valores: %.2f\n", Lista->Media);
        printf("Quantidade de valores acima da média: %d\n",
Lista->ValAcimaMedia);
    }

int main()
{
    SetConsoleOutputCP(CP_UTF8);
    FLVazia(&Lista);
    system("cls");
    do
    {
        printf("Insira o %dº número: ", Lista.Ultimo);
        scanf("%d", &x.notas);
        if (x.notas == -1)
            break;
        Insere(&x, &Lista);

    } while (x.notas != -1);
    Calcula(&Lista);
    Imprime(&Lista);

    return 0;
}

```

EXERCÍCIO 03

```

#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <windows.h>

#define InicioArranjo 1
#define MaxTam 5
typedef int Apontador;

```

```

typedef struct
{
    int numero;
} TipoItem;

typedef struct
{
    TipoItem Item[MaxTam];
    Apontador Primeiro;
    Apontador Ultimo;
    Apontador Menor, Maior;
    float Media;
} TipoLista;

TipoLista Lista;
TipoItem x;
Apontador pos;

void FLVazia(TipoLista *Lista)
{
    Lista->Primeiro = InicioArranjo;
    Lista->Ultimo = Lista->Primeiro;
}

void Insere(TipoItem *x, TipoLista *Lista)
{
    if (Lista->Ultimo > MaxTam)
        printf("\nA Lista está cheia\n");
    else
    {
        Lista->Item[Lista->Ultimo - 1] = *x;
        Lista->Ultimo++;
    }
}

void Imprime(TipoLista *Lista)
{
    Lista->Maior = Lista->Primeiro - 1;
    Lista->Menor = Lista->Primeiro - 1;
    Lista->Media = 0;
    printf("\nNúmeros: ");
    for (int i = Lista->Primeiro - 1; i <= Lista->Ultimo - 2; i++)
    {
        printf("%d ", Lista->Item[i].numero);
        Lista->Media += Lista->Item[i].numero;
    }
}

```

```

        if (Lista->Item[i].numero > Lista->Item[Lista->Maior].numero)
            Lista->Maior = i;
        if (Lista->Item[i].numero < Lista->Item[Lista->Menor].numero)
            Lista->Menor = i;
    }
    Lista->Media /= Lista->Ultimo - 1;
    printf("\nMaior número: %d\n", Lista->Item[Lista->Maior].numero);
    printf("Menor número: %d\n", Lista->Item[Lista->Menor].numero);
    printf("Média dos números: %f\n", Lista->Media);
}

int main()
{
    int opcao, resposta;
    SetConsoleOutputCP(CP_UTF8);
    FLVazia(&Lista);
    do
    {
        printf("Insira o %dº número: ", Lista.Ultimo);
        scanf("%d", &x);
        Insere(&x, &Lista);

    } while (Lista.Ultimo <= MaxTam);
    Imprime(&Lista);

    return 0;
}

```

EXERCÍCIO 04

```

#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <windows.h>

#define InicioArranjo 1
#define MaxTam 10
typedef int Apontador;

typedef struct
{
    int numero;
} TipoItem;

typedef struct
{

```

```

    TipoItem Item[MaxTam];
    Apontador Primeiro;
    Apontador Ultimo;
} TipoLista;

TipoLista Lista;
TipoItem x;
Apontador pos;

void FLVazia(TipoLista *Lista)
{
    Lista->Primeiro = InicioArranjo;
    Lista->Ultimo = Lista->Primeiro;
}

void Insere(TipoItem *x, TipoLista *Lista)
{
    if (Lista->Ultimo > MaxTam)
        printf("\nA Lista está cheia\n");
    else
    {
        Lista->Item[Lista->Ultimo - 1] = *x;
        Lista->Ultimo++;
    }
}

void OrdenaDec(TipoLista *Lista)
{
    TipoItem aux;
    for (int i = MaxTam - 1; i >= 0; i--)
    {
        for (int j = 0; j < i; j++)
        {
            if (Lista->Item[j].numero < Lista->Item[j + 1].numero)
            {
                aux = Lista->Item[j];
                Lista->Item[j] = Lista->Item[j + 1];
                Lista->Item[j + 1] = aux;
            }
        }
    }
}

void Imprime(TipoLista *Lista)

```

```

{
    printf("\nLista em ordem decrescente:\n");
    for (int i = Lista->Primeiro - 1; i < Lista->Ultimo - 1; i++)
        printf("%d ", Lista->Item[i].numero);
    printf("\n");
}

int main()
{
    SetConsoleOutputCP(CP_UTF8);
    FLVazia(&Lista);
    printf("\nInsira %d números\n\n", MaxTam);
    do
    {
        printf("Insira o %dº número: ", Lista.Ultimo);
        scanf("%d", &x);
        Insere(&x, &Lista);

    } while (Lista.Ultimo <= MaxTam);
    OrdenaDec(&Lista);
    Imprime(&Lista);

    return 0;
}

```

EXERCÍCIO 05

```

#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <windows.h>

#define InicioArranjo 1
#define MaxTam 5
typedef int Apontador;

typedef struct
{
    int resp;
} TipoItem;

typedef struct
{
    TipoItem Item[MaxTam];
    Apontador Primeiro;
}

```

```

Apontador Ultimo;
int Classificaco;
} TipoLista;

TipoLista Lista;
TipoItem x;
Apontador pos;

void FLVazia(TipoLista *Lista)
{
    Lista->Primeiro = InicioArranjo;
    Lista->Ultimo = Lista->Primeiro;
}

void Imprime(TipoLista *Lista)
{
    Lista->Classificaco = 0;
    printf("\n");
    for (int i = Lista->Primeiro - 1; i <= Lista->Ultimo - 2; i++)
    {
        if (Lista->Item[i].resp == 1)
            Lista->Classificaco++;
    }
    if (Lista->Classificaco == 2)
        printf("Suspeita\n");
    else if (Lista->Classificaco >= 3 && Lista->Classificaco <= 4)
        printf("Cúmplice\n");
    else if (Lista->Classificaco == 5)
        printf("Assassino\n");
    else
        printf("Inocente\n");
}

int main()
{
    SetConsoleOutputCP(CP_UTF8);
    FLVazia(&Lista);
    system("cls");
    printf("\nRESPOnda COM 1 PARA SIM OU 0 PARA NÃO\n\n");

    printf("Telefonou para a vítima?\n");
    scanf("%d", &Lista.Item[Lista.Ultimo - 1].resp);
    Lista.Ultimo++;

    printf("Esteve no local do crime?\n");

```

```
scanf("%d", &Lista.Item[Lista.Ultimo - 1].resp);
Lista.Ultimo++;

printf("Mora perto da vítima?\n");
scanf("%d", &Lista.Item[Lista.Ultimo - 1].resp);
Lista.Ultimo++;

printf("Devia para a vítima?\n");
scanf("%d", &Lista.Item[Lista.Ultimo - 1].resp);
Lista.Ultimo++;

printf("Já trabalhou com a vítima?\n");
scanf("%d", &Lista.Item[Lista.Ultimo - 1].resp);
Lista.Ultimo++;

Imprime(&Lista);

return 0;
}
```