



Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica



CSI488 – ALGORITMOS ESTRUTURAS DE DADOS I (TURMA 22)

TAD LISTA

Jhonatan Figueiredo Almeida

João Monlevade
Setembro de 2022

CSI488 – ALGORITMOS ESTRUTURAS DE DADOS I (TURMA 22)

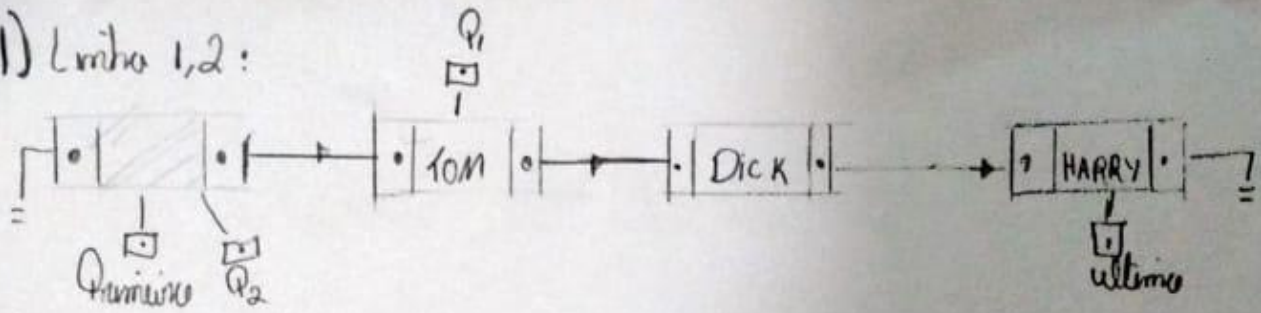
TAD LISTA

Discente: Jhonatan Figueiredo Almeida
Professor: Alexandre Magno de Souza

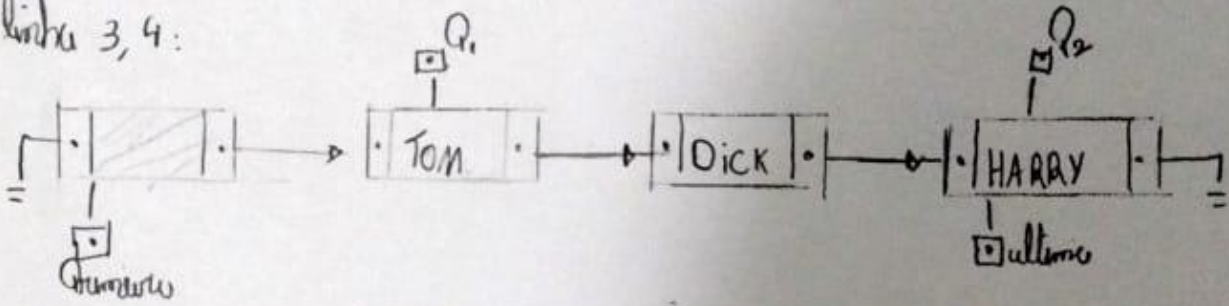
Matrícula: 20.1.8164

João Monlevade
Setembro de 2022

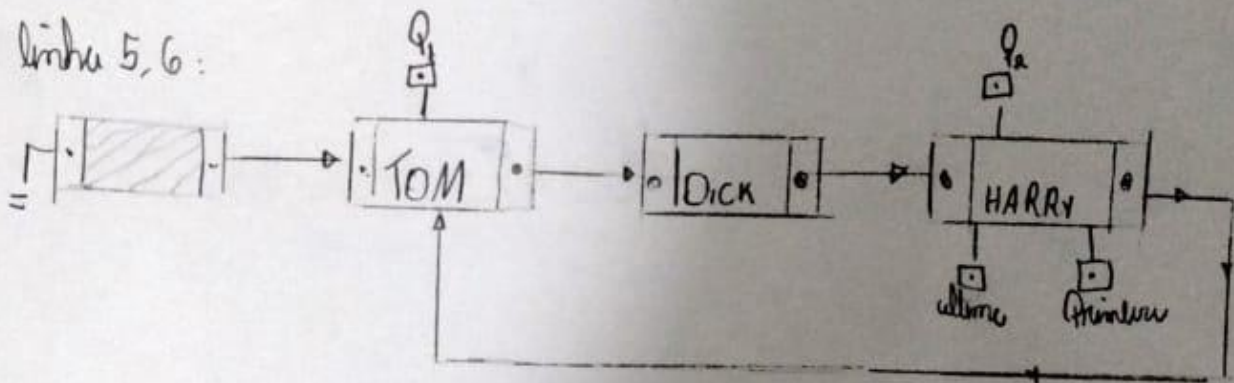
1) Linha 1,2:



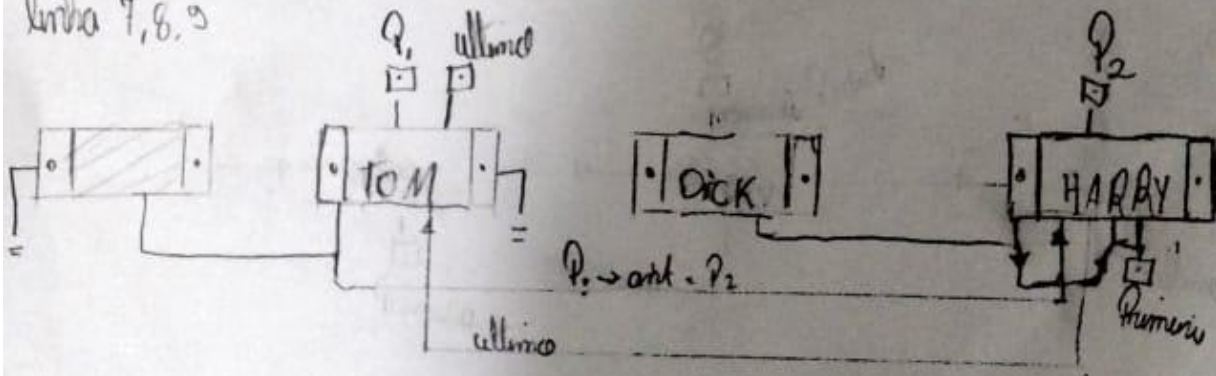
linha 3,4:

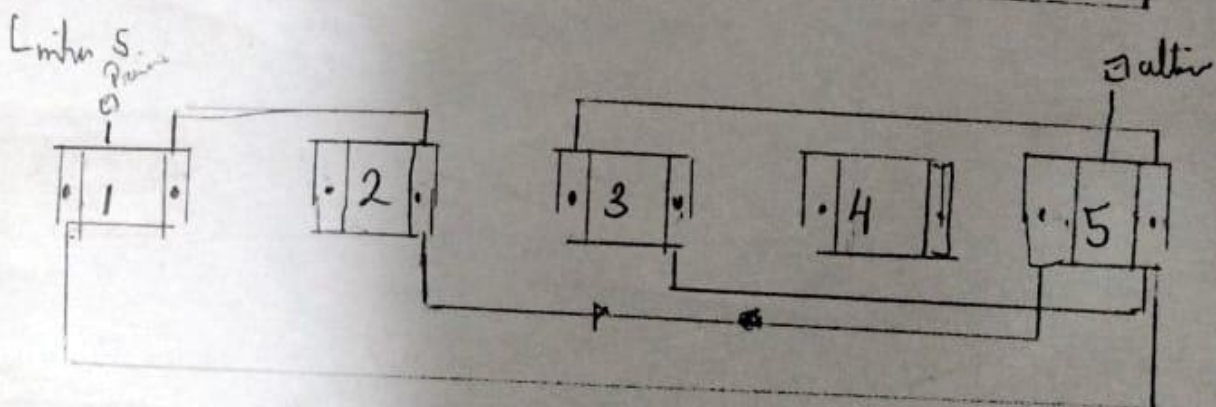
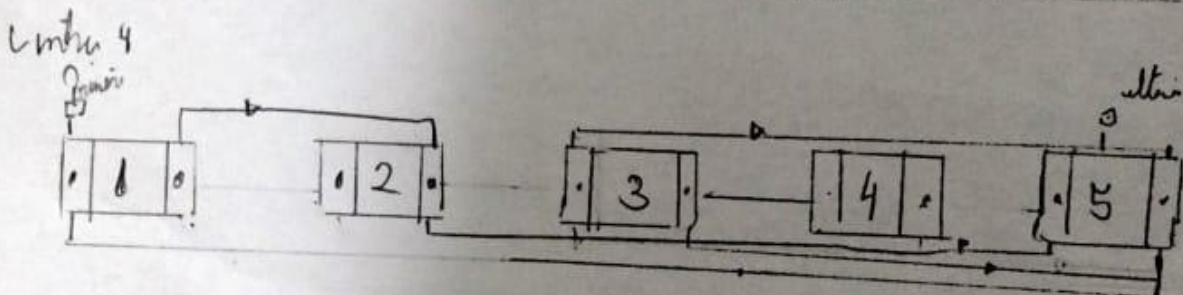
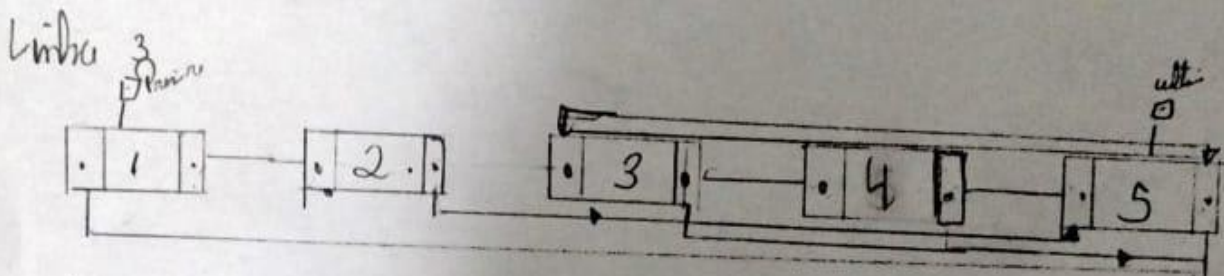
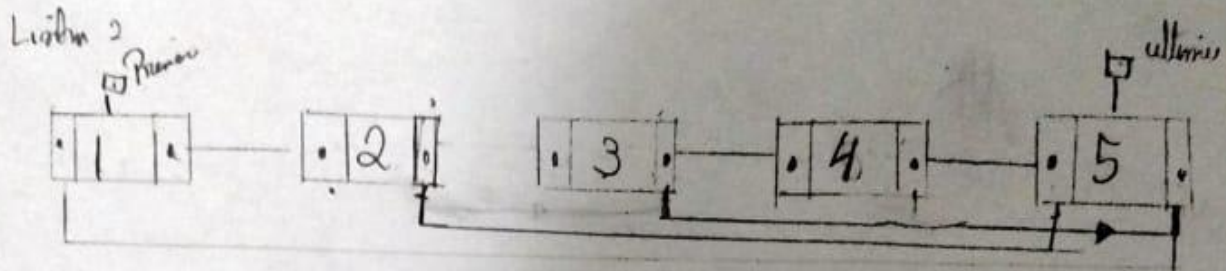
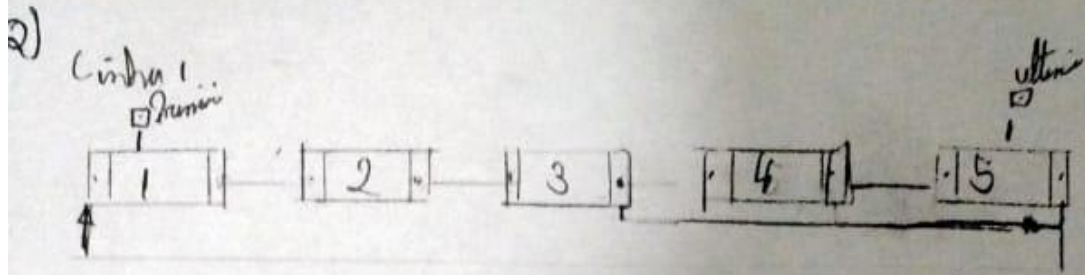


linha 5,6:



linha 7,8,9





```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
typedef struct
```

```
{
```

```
    int codigo;
```

```
    char nome[200];
```

```
    int quantidade ;
```

```
    float preco;
```

```
} Tproduto;
```

```
typedef struct celula
```

```
{
```

```
    Tproduto item;
```

```
    struct celula * prox;
```

```
} Tcelula;
```

```
typedef struct
```

```
{  
  
    Tcelula * primeiro;  
  
    Tcelula * ultimo;  
  
    int tamanho;  
  
} Tlista;
```

```
void FLVazia (Tlista *lista )
```

```
{  
  
    lista ->primeiro= (Tcelula*) malloc (sizeof (Tcelula));  
  
    lista ->ultimo = lista ->primeiro;  
  
    lista ->primeiro->prox = NULL;  
  
    lista->tamanho=0;  
  
}
```

```
int Vazia (Tlista lista)
```

```
{  
  
    return (lista.primeiro== lista.ultimo);  
  
}
```

```
int Tamanho ( Tlista lista)
```

```
{  
  
    return lista.tamanho;  
  
}
```

```
void inserir (Tproduto x, Tlista *lista)
```

```
{  
  
    lista->ultimo->prox= (Tcelula*) malloc (sizeof ( Tcelula));  
  
    lista->ultimo= lista->ultimo->prox;  
  
    lista->ultimo->item= x;  
  
    lista->ultimo->prox=NULL;  
  
    lista->tamanho++;  
  
}
```

```
void Ler (Tproduto *h){
```

```
    printf("nome:");  
  
    fflush(stdin);  
  
    fgets(h ->nome , 200, stdin);  
  
  
    printf("Codigo:");  
  
    fflush(stdin);  
  
    scanf("%d",&h ->codigo);
```

```
printf("preco:");
```

```
fflush(stdin);
```

```
scanf("%f",&h ->preco);
```

```
}
```

```
void ImprimirProduto ( Tproduto h){
```

```
    printf("\n=====\\n");
```

```
    printf("nome:%s",h.nome);
```

```
    printf("Codigo:%d\\n",h.codigo);
```

```
    printf("preco:%.2f\\n",h.preco);
```

```
    printf("\n=====\\n");
```

```
}
```

```
void Imprimir (Tlista Lista){
```

```
    Tcelula *aux;
```

```
    aux= Lista.primeiro->prox;
```

```
    while (aux != NULL){
```

```
        ImprimirProduto (aux->item );
```

```
        aux= aux->prox;
```

```
    }
```



```

}

Tcelula * Pesquisar (Tlista lista , Tproduto Item){

    Tcelula * aux;

    aux =lista.primeiro;

    while (aux->prox != NULL){

        if(aux->prox->item.codigo==Item.codigo){

            return aux;

        }

        aux=aux->prox;

    }

    return NULL;

}

```

```

void Excluir (Tlista *Lista , Tproduto *Item){

    Tcelula *aux1 , *aux2;

    aux1=Pesquisar (*Lista , *Item);

    if(aux1!=NULL){

        aux2 = aux1->prox;

        aux1->prox = aux2->prox;

        *Item =aux2 ->item;
    }
}

```

```
if(aux1->prox==NULL){  
  
    Lista->ultimo = aux1;  
  
}  
  
free(aux2);  
  
Lista->tamanho--;  
  
}  
  
else {  
  
    Item->codigo=-1;  
  
}  
  
}
```

```
int main (){
```

```
    Tlista  L1, L2;
```

```
    Tproduto x;
```

```
    FLVazia(&L1);
```

```
    FLVazia(&L2);
```

```
    for (int i =0; i<10;i++){
```

```
        Ler(&x);
```

```
        printf("\n");
```

```
inserir(x,&L1);

inserir(x,&L2);

}

printf("\n===== \n Lista1:");

Imprimir(L1);

printf("\n===== \n Lista2;");

Imprimir(L2);
```

```
x = L1.primeiro->prox->prox->prox->item;
```

```
Excluir(&L1, &x);
```

```
x = L1.primeiro->prox->item;
```

```
Excluir(&L1, &x);
```

```
x= L1.primeiro->prox->prox->item;
```

```
Excluir(&L1, &x);
```

```
x= L1.primeiro->prox->item;
```

```
Excluir(&L1, &x);
```

```
printf("\n=====");
```

```
printf("Lista 1 dps da exclu");
```

```
Imprimir(L1);
```

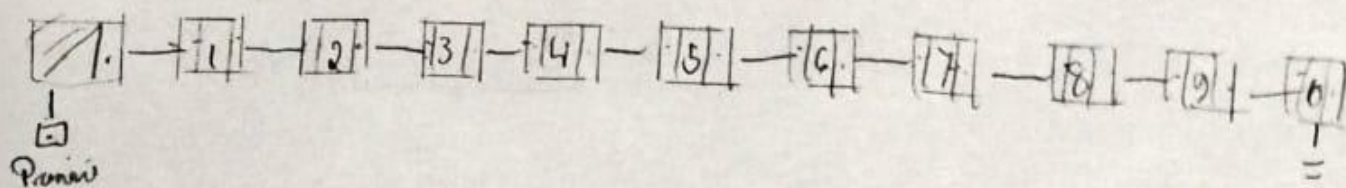
```
printf("\n===== \n Lista2 dps da ex");
```

```
Imprimir(L2);
```

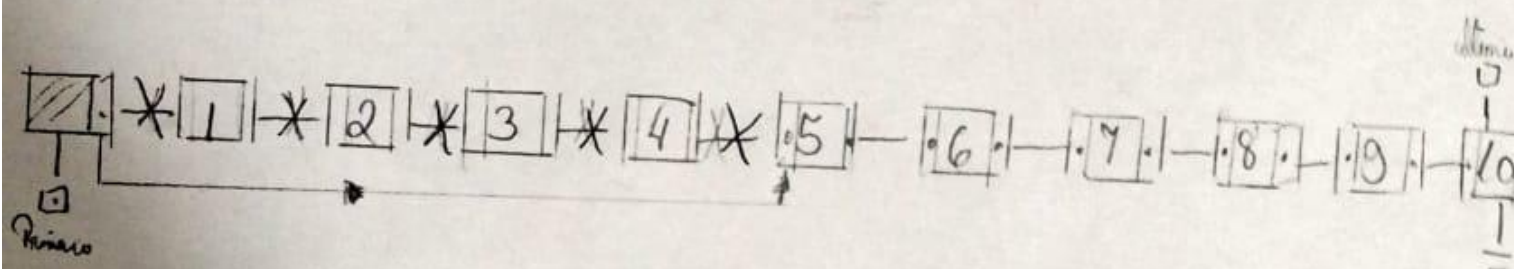
```
return 0;
```

```
}
```

Lista 2 9)



Lista 1



A cada linha de código anterior vai excluindo uma célula, ficando então a lista 1 menor que a lista 2.

