laboratório 02: Listas encadeadas

tarefa 1: inicialização

Escreva um programa que cria uma lista encadeada de tamanho n, para algum n dado como entrada. Para isso você vai precisar implementar a classe `No` da seguinte forma:

```
class No {
        public:
        int chave;
        No* prox = NULL;
}
```

Escreva um método

```
No* nova_lista_encadeada(int n) {}
```

que cria a lista de tamanho n e retorna um ponteiro para o primeiro elemento da lista.

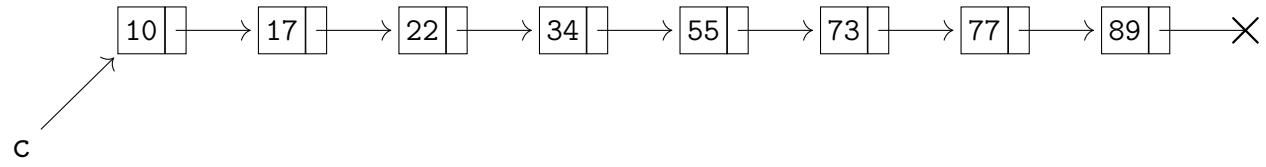
tarefa 2: desinicialização

Escreva um método que recebe uma lista (um ponteiro para `No`) e "apaga" a lista, isto é, desaloca cada nó. Para isso, você deverá usar o comando `delete` do C++:

```
        No* h;
        ...

        delete h;
```

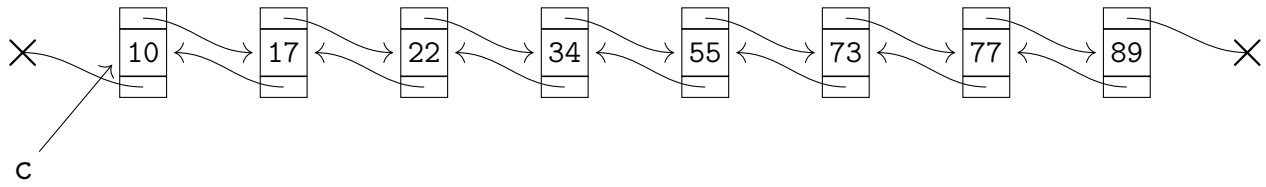
tarefa 3: impressão reversa

Escreva dois programas, um recursivo e um não recursivo, que imprime uma lista encadeada em ordem reversa. Por exemplo:



tarefa 4: lista duplamente encadeada

Uma lista duplamente encadeada é um lista onde cada elemento aponta para o próximo e para o anterior.



Para isso, o nó deve ser definido da seguinte maneira:

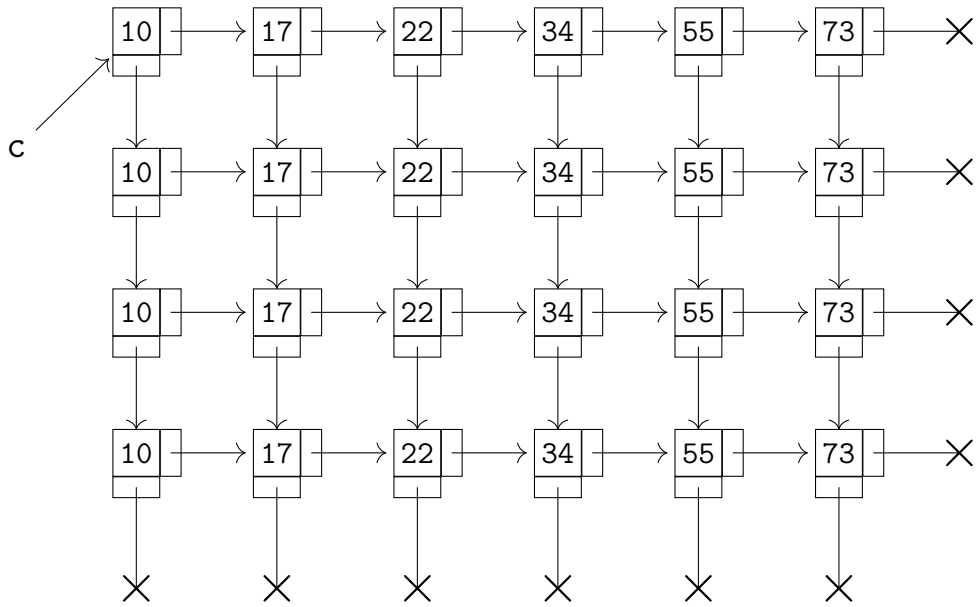
```
class NoDE {
        public:
        int chave;
        NoDE* prox = NULL;
        NoDE* ante = NULL;
}
```

Crie um classe `ListaDE` e escreva métodos para a inserção de um nó na lista encadeada em uma posição dada e para a remoção de um nó dada uma chave. A remoção retorna um ponteiro para o nó removido (não desaloca o espaço reservado ao nó).

```
class ListaDE {
        public:

        NoDE* cabeca = NULL;
        int cont = 0;

        void inserir(NoDE* no){
        ...
        }

        NoDE* remover(int chave){
        ...
        }
};
```

Matriz encadeada

Uma matriz encadeada é um conjunto de nós em que cada nó possui um vizinho à direita e um vizinho abaixo.

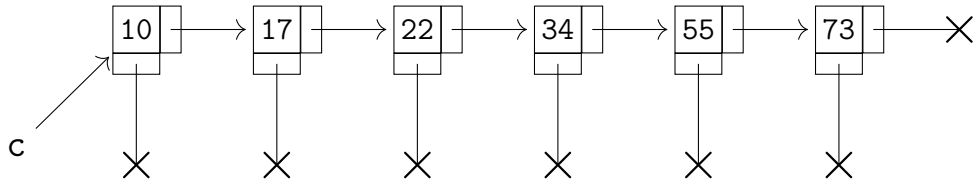


Um nó de matriz encadeada pode ser definido assim:

```
class NoMat {
        public:
        int chave;
        No* direita = NULL;
        No* abaixo = NULL;
}
```

tarefa 5: matriz linha

Escreva um programa que, dado n, gera um matriz linha, ou seja, apenas uma lista de tamanho n de nós de matriz encadeada.




```
NoMat* matriz_linha(int n){
...
}
```

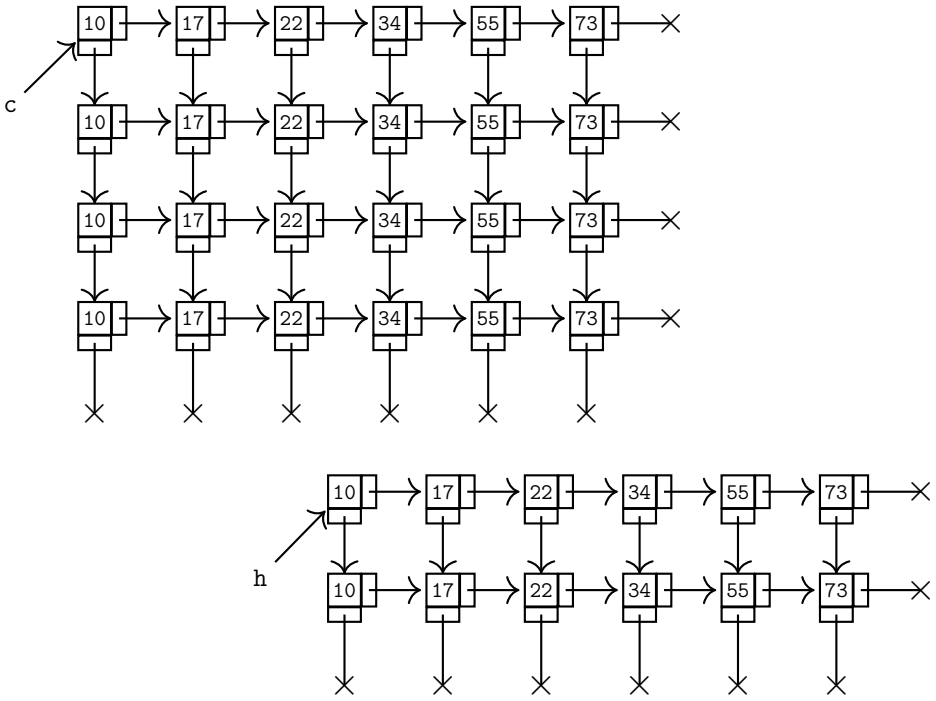
tarefa 6: imprimindo matriz encadeada

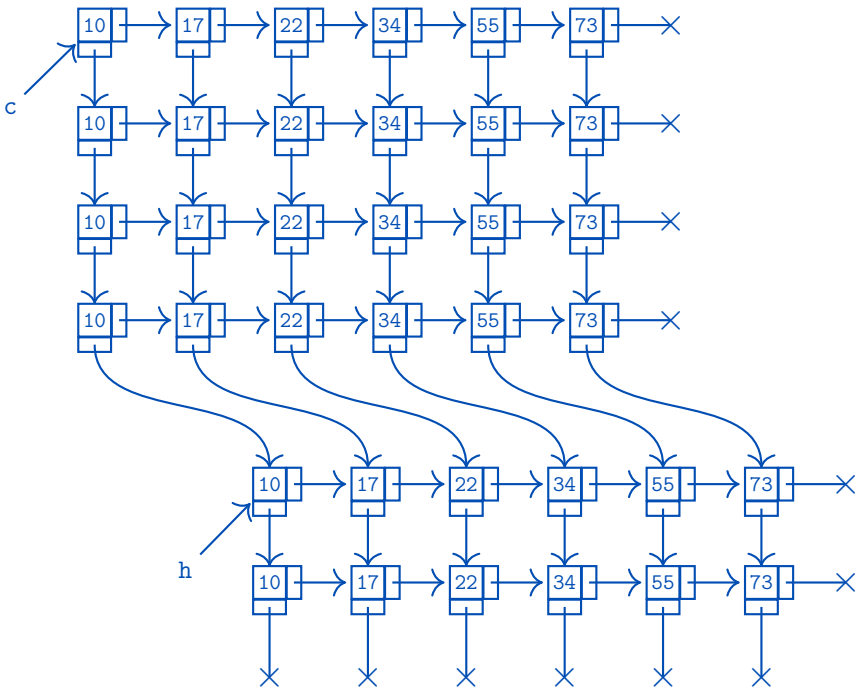
Escreva um método que, dada uma matriz encadeada, imprime a matriz. Seu método deve receber apenas a matriz, não recebe as dimensões.

```
void matriz_linha(NoMat* m){
...
}
```

tarefa 7: costurando matrizes

Faça um algoritmo que, dadas duas matrizes encadeadas com o mesmo número de colunas, uma de dimensões $n \times m$ e outra de dimensões $l \times m$, anexa uma matriz à outra produzindo uma matriz de dimensões $(n + l) \times m$.





tarefa 8: usando uma matriz encadeada como... uma matriz

Escreva um método que, dados inteiros i e j, retorna o elemento da posição (i, j) de uma matriz encadeada.