



ED1 03: Arrays e Listas Ligadas.

Fabio Irigon Pereira

Arrays



- Estrutura de dados contígua na memória.
- Acesso direto a elementos via índice ($O(1)$).
- Tamanho fixo (deve ser definido no momento da alocação).
- Ineficiente para inserções e remoções no meio do array ($O(n)$).

Arrays



```
#include <stdio.h>

int main() {
    int arr[5] = {10, 20, 30, 40, 50}; // Array de tamanho fixo

    // Acesso direto ao terceiro elemento
    printf("Elemento na posição 2: %d\n", arr[2]); // O(1)

    return 0;
}
```

Listas Ligadas



- Conjunto de nós não contíguos armazenados **dinamicamente**.
- Cada nó contém:
 - **Dados**.
 - **Um ponteiro** para o próximo nó.
- Inserções e remoções são eficientes ($O(1)$ se feitas no início).
- Acesso a um elemento específico é mais lento ($O(n)$) pois requer percorrer a lista.

Estrutura de um nó de lista ligada



```
typedef struct No {  
    char nome[50];  
    int idade;  
    struct No * prox; // Ponteiro para o próximo nó  
} No;
```

Implementação: Arrays em C

Inserção em Arrays



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

#define TAMANHO_MAX 5

typedef struct {
    char nome[50];
    int idade;
} No;
```

```
int main() {
    No arr[TAMANHO_MAX];
    int tamanho = 0;

    // Inserindo elementos no array
    strcpy(arr[tamanho].nome, "Ana");
    arr[tamanho].idade = 25;
    tamanho++;

    strcpy(arr[tamanho].nome, "Carlos");
    arr[tamanho].idade = 30;
    tamanho++;

    // Imprimindo os elementos do array
    printf("Lista de Pessoas (Array):\n");
    for (int i = 0; i < tamanho; i++) {
        printf("Nome: %s, Idade: %d\n",
            arr[i].nome, arr[i].idade);
    }

    return 0;
}
```

Inserção no meio do Array



```
void inserirMeio(No arr[], int * tamanho, int pos, char nome[], int idade) {
    if (* tamanho >= TAMANHO_MAX || pos > * tamanho) {
        printf("Erro: Posição inválida ou array cheio.\n");
        return;
    }

    for (int i = * tamanho; i > pos; i--) {
        arr[i] = arr[i - 1]; // Desloca os elementos
    }

    strcpy(arr[pos].nome, nome);
    arr[pos].idade = idade;
    (* tamanho)++;
}
```

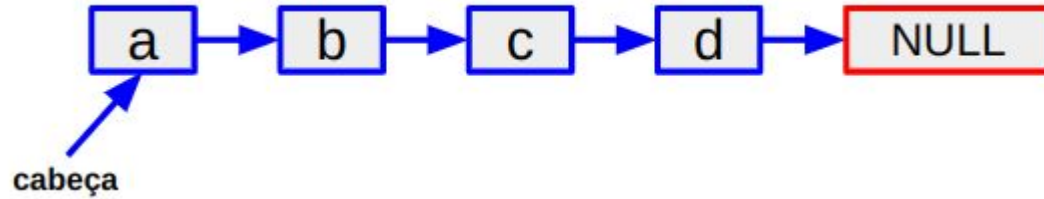

Exercício



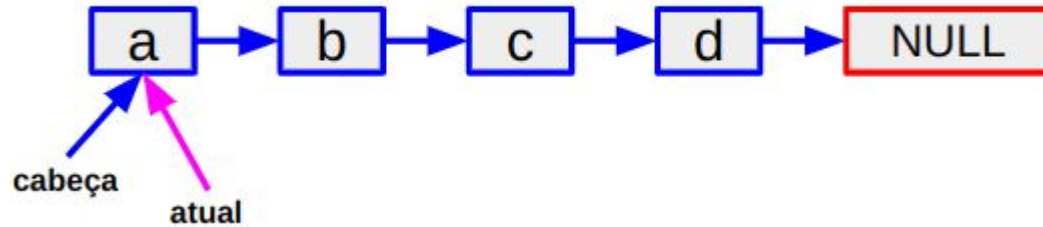
Tente criar uma função para remover um elemento no meio do array.

Implementação: Lista ligada

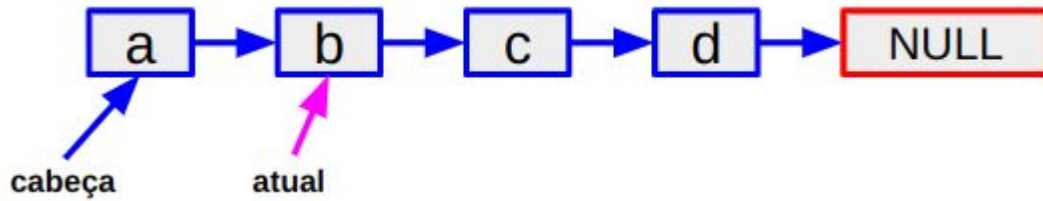
Imprimir (percorrer) lista ligada



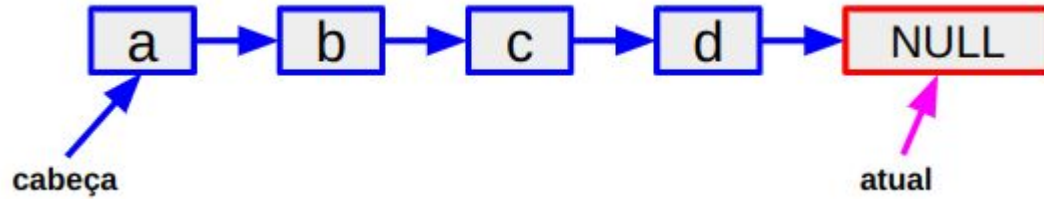
Imprimir (percorrer) lista ligada



Imprimir (percorrer) lista ligada



Imprimir (percorrer) lista ligada



Imprimir (percorrer) lista ligada



```
// Função para imprimir a lista
void imprimirLista(No * cabeca) {
    No * atual = cabeca;
    printf("Lista de Pessoas (Lista Ligada):\n");
    while (atual != NULL) {
        printf("Nome: %s, Idade: %d\n", atual -> nome, atual -> idade);
        atual = atual -> prox;
    }
}
```

Exercício:



Tente criar uma rotina para adicionar um elemento entre o primeiro e o segundo elemento de uma lista ligada.

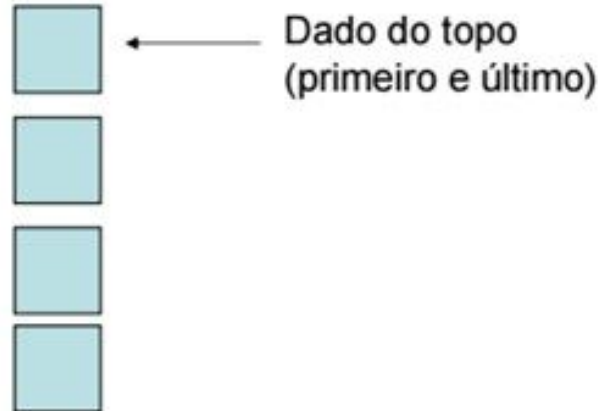
Pilhas (LIFO) e Filas (FIFO)

Pilhas e filas

FILA:



PILHA:



Pilhas e Filas

