

## Atividade B2-6 – Lista Ordenada – Inserir Dados

### Enunciado:

#### Instruções

Desenvolver 2 algoritmos na linguagem C.

Supondo um array com os elementos 10, 25, 40, 80 e também a lista ligada com a sequência de nós em ordem crescente; 10, 25, 40 e 80.

- Algoritmo 1 - Inserir no Array  
o algoritmo deverá inserir um determinado valor (30 por exemplo), no meio do array
- Algoritmo 2 - Inserir em lista ligada  
o algoritmo deverá inserir um nó, com o valor 30 no meio da lista ligada.

Ao término de cada algoritmo deverá ser informado o tempo de processamento do programa. para isto, considere a primeira instrução feita no programa (tempo inicial) e o última instrução efetuada pelo programa.

### Código Fonte:

**Link:** [link do repositório no Github](#)

- **Algoritmo 1:** Neste programa foi adicionada a opção ao usuário de inserir um número no intervalo de 26 até 39 (devido a sequência da ordem crescente dos números) ou adicionar automaticamente o número 30.

```
/*-----*/
/* FATEC - São Caetano do Sul    Estrutura de Dados */
/*                               */
/*           Camille Guillen      */
/*      Objetivo: Lista Ordenada - Inserir Dados    */
/*                               */
/*                               Data: 25/05/2024     */
/*-----*/
```

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

```
#define ARRAY_SIZE 4
#define MIDDLE_INDEX 2
```

```
void printArray(int *array, int size) {
```

```

    for (int i = 0; i < size; i++) {
        printf("%d ", array[i]);
    }
    printf("\n");
}

int main() {
    int array[ARRAY_SIZE + 1] = {10, 25, 40, 80}; // Inicia com o espaço
    extra
    int size = ARRAY_SIZE;
    int value;
    char option;

    clock_t start_time, end_time;
    double cpu_time_used;

    // Início do tempo de processamento
    start_time = clock();

    while (1) {
        printf("Escolha uma opção:\n");
        printf("\n1. Digitar um número entre 26 e 39\n");
        printf("2. Adicionar automaticamente o número 30\n");
        printf("\nOpção: ");
        scanf(" %c", &option);

        if (option == '1') {
            printf("\nDigite um número entre 26 e 39: ");
            scanf("%d", &value);
            if (value < 26 || value > 39) {
                printf("\nErro: Número fora do intervalo permitido. Tente
    novamente.\n");
                continue;
            }
        } else if (option == '2') {
            value = 30;
        } else {
            printf("\nErro: Opção inválida. Escolha novamente.\n");
            continue;
        }
        break;
    }

    // Move os elementos para inserir o novo valor no meio
    for (int i = size; i > MIDDLE_INDEX; i--) {
        array[i] = array[i - 1];
    }
    array[MIDDLE_INDEX] = value;

```

```

size++;

// Imprime o array resultante
printf("\nArray atualizado: ");
printArray(array, size);

// Fim do tempo de processamento
end_time = clock();
cpu_time_used = ((double)(end_time - start_time)) /
CLOCKS_PER_SEC;

printf("\nTempo de processamento: %f segundos\n",
cpu_time_used);
printf("\nFim do programa...\n");

return 0;
}

```

- **Algoritmo 2:** Neste programa, é inserido o nó com o número 30 no meio da lista ligada.

```

/*-----*/
/* FATEC - São Caetano do Sul      Estrutura de Dados */
/*                                */
/*          Camille Guillen        */
/*      Objetivo: Lista Ordenada - Inserir Dados      */
/*                                */
/*                                Data: 25/05/2024      */
/*-----*/

```

```

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

```

```

typedef struct No {
    int valor;
    struct No* proximo;
} No;

```

```

No* criarNo(int valor) {
    No* novoNo = (No*)malloc(sizeof(No));
    novoNo->valor = valor;
    novoNo->proximo = NULL;
    return novoNo;
}

```

```

void inserirNaLista(No** cabeca, int valor) {

```

```

    No* novoNo = criarNo(valor);

    // Se a lista estiver vazia ou o novo valor for menor que o valor do
    primeiro nó
    if (*cabeca == NULL || (*cabeca)->valor >= valor) {
        novoNo->proximo = *cabeca;
        *cabeca = novoNo;
        return;
    }

    No* atual = *cabeca;
    // Encontrar o ponto de inserção
    while (atual->proximo != NULL && atual->proximo->valor < valor) {
        atual = atual->proximo;
    }

    novoNo->proximo = atual->proximo;
    atual->proximo = novoNo;
}

void imprimirLista(No* no) {
    while (no != NULL) {
        printf("\033[0;34m%d\033[0m ", no->valor); // Cor azul
        no = no->proximo;
    }
    printf("\n");
}

int main() {
    No* cabeca = criarNo(10);
    cabeca->proximo = criarNo(25);
    cabeca->proximo->proximo = criarNo(40);
    cabeca->proximo->proximo->proximo = criarNo(80);

    int valor = 30; // Valor a ser inserido

    clock_t inicio, fim;
    double tempo_gasto;

    inicio = clock();
    inserirNaLista(&cabeca, valor);
    fim = clock();

    tempo_gasto = ((double)(fim - inicio)) / CLOCKS_PER_SEC;

    // Imprimindo a lista atualizada
    imprimirLista(cabeca);

```

```
printf("\nTempo de processamento: %f segundos\n",  
tempo_gasto);
```

```
// Liberando a memória alocada
```

```
No* atual = cabeca;
```

```
No* proximo;
```

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

```
    proximo = atual->proximo;
```

```
    free(atual);
```

```
    atual = proximo;
```

```
}
```

```
return 0;
```

```
}
```