

**INE5410 – Programação Concorrente**  
**Profs. Giovani Gracioli e Márcio Bastos Castro**

## **Exercícios de Programação com Linguagem C**

1) Retornando múltiplos valores de uma função com estrutura

As vezes será necessário retornar múltiplos valores de uma função. Nessa situação é possível usar uma estrutura e retornar os valores dentro de um ponteiro para essa estrutura. Escreva uma função que deve retornar os valores mínimos e máximos dentro de um vetor.

Para isso utilize uma estrutura MinMax conforme abaixo:

```
typedef struct MinMax
{
    int min;
    int max;
}MinMax;
```

A função tem o seguinte protótipo:

```
MinMax *getMinMax(int * array, const int SIZE);
```

O argumento **array** contém um vetor de inteiros com tamanho **SIZE** e deve retornar um ponteiro para uma estrutura alocada dinamicamente dentro da própria função. Lembre-se de desalocar a memória alocado ao final da função main (que deve ser escrita por você e conter a declaração de um vetor e a chamada a função **getMinMax**).

Para verificar se o código possui vazamento de memória, utilize a ferramenta valgrind:

```
valgrind --leak-check=full --show-leak-kinds=all ./main
```

2) Escreva um programa em C para permitir a ordenação de vetores.

Exemplo de entrada:

Elementos do vetor de entrada: 10 -1 0 4 2 100 15 20 24 -5

Exemplo de saída:

Vetor ordenado em ordem ascendente: -5, -1, 0, 2, 4, 10, 15, 20, 24, 100,

Vetor ordenado em ordem descendente: 100, 24, 20, 15, 10, 4, 2, 0, -1, -5,

Para isso, você deve implementar as seguintes funções:

```
//Lê os elementos do teclado e os coloca em arr. Size possui o tamanho do vetor
void inputArray(int * arr, int size);
```

```
//Imprime o conteúdo do vetor arr com tamanho size
void printArray(int * arr, int size);
```

//Função que compara dois inteiros. A função retorna a diferença entre o primeiro e o segundo

```
int sortAscending(int * num1, int * num2);
```

//Função que compara dois inteiros. A função retorna a diferença entre o segundo e o primeiro

```
int sortDescending(int * num1, int * num2);
```

//Função que realiza a ordenação. O terceiro argumento é um ponteiro para função que realiza a comparação entre dois inteiros do vetor (funções sortAscending ou sortDescending)

```
void sort(int * arr, int size, int (* compare)(int *, int *));
```

O conteúdo da função main é dado abaixo:

```
int main()
{
    int arr[MAX_SIZE];
    int size;

    /*
     * Input array size and elements.
     */
    printf("Enter array size: ");
    scanf("%d", &size);
    printf("Enter elements in array: ");
    inputArray(arr, size);

    printf("\n\nElements before sorting: ");
    printArray(arr, size);

    // Sort and print sorted array in ascending order.
    printf("\n\nArray in ascending order: ");
    sort(arr, size, sortAscending);
    printArray(arr, size);

    // Sort and print sorted array in descending order.
    printf("\n\nArray in descending order: ");
    sort(arr, size, sortDescending);
    printArray(arr, size);

    return 0;
}
```