Programação Imperativa e Funcional

Estruturas de Repetição e Matrizes

erico.souza.teixeira
est@cesar.school

Multiplique duas Matrizes

```
while (condicional){
  //faça algo
}

do {
  //faça algo
} while (condicional);

for(inicialização; teste; atualização){
  //faça algo
}
```

Arrays

- **Coleção de variáveis de mesmo tipo**
- **Referenciada por um nome comum**
- **Associada a um índice**
- **Relação com ponteiros**
- **Declaração**
  - `tipo identificador[tamanho];`

Arrays

- **Inicialização**

```
int point[6]={0,0,1,0,0,0};
int point[]={0,0,1,0,0,0};
int number[2000]={1234};
```

Arrays

- **Acesso**

```
int point[6]={0,0,1,0,0,0};
int x=point[3];
```

- **Índice: 0 a tamanho-1**

```
size = sizeof(point)/sizeof(int)
size = sizeof(point)/sizeof point[0]
```

Matrizes

- **Declaração e acesso**

```
int point[3][2];
point[2][1] = 3;
```

- **Inicialização**

```
point={{0,0},
       {1,0},
       {0,0}};
point={{0,0},{1,0},{0,0}};
point={0,0,1,0,0,0};
```

Matrizes

- **Declaração e inicialização (linhas omitidas)**

```
int point[][2]={{0,0},
                {1,0},
                {0,0}};
```

Strings

- ***Arrays* de caracteres**

```
char nome[6] = "CESAR";
char nome[6] = {'C','E','S','A','R','\0'};
char string[58] = "This is a very, very long "
                  "string that requires two lines.";
```

- `strings.h`

Array de *strings*

- Declaração e inicialização

```
char ch_arr[3][10] = {
        {'s', 'p', 'i', 'k', 'e', '\0'},
        {'t', 'o', 'm','\0'},
        {'j', 'e', 'r', 'r', 'y','\0'}
                    };
```

Exercício

Escreve um programa que leia uma sequência de 10 números inteiros e retorne a subsequência de elementos adjacentes que tenha o maior valor quando todos os elementos da mesma são somados. Em caso de duas ou mais subsequências com o maior valor, retorne aquela de menor média.