

# Laboratório 3

20/08/2020

# Dicas de código

## 1 - Sempre iniciar as variáveis na declaração.

```
int soma=0, contador=0;
```

## O mesmo vale para vetores

```
int vetor[10];  
  
vetor[0] += 5;  
  
printf("%d", vetor[0]);
```



A posição 0 não tem valor. Deve imprimir um lixo de memória.

# Dicas de código

## 2 - Prestar atenção nas variáveis dentro da repetição.

```
int i, j;  
  
int matriz[10][10];  
  
for(i=0; i<10;i++){  
  
    for(j=0;j<10;j++){  
  
        printf("%d", matriz[i,j]);  
  
    }  
  
}
```

# Dicas de código

## 3- É preciso especificar o tamanho do vetor.



Vetor criado sem  
especificar o tamanho.

```
int tamanho;  
  
int vetor[tamanho];  
  
scanf("%d", tamanho);
```

# Dicas de código

## 4 - Criação de vetor em tempo de execução.



Nova característica  
adicionada na versão  
ISO C99.

```
int tamanho_vetor = 0;  
scanf("%d", &tamanho_vetor );  
int vetor[tamanho_vetor];
```

**Arrays of Variable Length**

**Link:** <http://gcc.gnu.org/onlinedocs/gcc-4.6.1/gcc/Variable-Length.html>

# Exercícios

3.3 - Ordem inversa: **vetor** + comando de repetição

3.4 - Matrícula em comum: **vetor** + comando de repetição

3.5 - Maior da matriz: **matriz** + comando de repetição

## 3.3 - Ordem inversa

Ler 10 elementos e gerar um outro vetor com esses 10 elementos em ordem inversa

Entrada	Saída
1	10
2	9
3	8
4	7
5	6
6	5
7	4
8	3
9	2
10	1

# Solução: Luiz Daniel

```
int x[10], y[10];

for (int i = 0; i < 10; i++)
{
    scanf("%d", &x[i]);
}

for (int i = 0; i < 10; i++)
{
    y[i] = x[9 - i];
    printf("%d\n", y[i]);
}

return 0;
```



## 3.4 - Matrícula em Comum

Ler 2 vetores e imprimir as matrículas em comum

Entrada	Saída
5 //número de alunos em AEDS 1 123 321 456 654 888 2 //número de alunos em Cálculo 1 123 321	123 321

# Solução: Luiza Araujo Martins

```
scanf("%i",&x);  
int aeds[x];  
for(i=0; i < x; i++){  
    scanf("%i",&aeds[i]);  
}  
scanf("%i",&y);  
int calc[y];  
for(i=0; i < y; i++){  
    scanf("%i",&calc[i]);  
}  
//imprime a interseção  
for(i=0; i < x; i++){  
    for(j=0; j < y; j++){  
        if(aeds[i]==calc[j]) printf("%i\\n",aeds[i]);  
    }  
}
```

## 3.5 - Maior da Matriz

Ler uma matriz e encontrar o maior valor.

Entrada	Saída
3 //qtd linhas 4 //qtd colunas 1 1 8 7 0 8 6 4 5 5 2 4	8

# Solução: Fernando Gomes.

```
int qtd_linhas, qtd_colunas, i, j, maior;
scanf("%d%d", &qtd_linhas, &qtd_colunas);
int matriz [qtd_linhas][qtd_colunas];

for (i=0; i < qtd_linhas; i++){
    for (j=0; j < qtd_colunas; j++)
        scanf("%d",&matriz [i][j]);
}
maior= matriz[0][0];
for(i=0; i < qtd_linhas; i++){
    for (j=0; j < qtd_colunas; j++)
        if (matriz [i][j] > maior)
            maior = matriz[i][j];
}
printf("\n%d",maior);
}
```

# Exercícios

3.1 - Primos num intervalo: comando de repetição

3.2 - Série e Arrays - Fibonacci - **veter** + comando de repetição

3.6 - Matriz Oposta: **matriz** + comando de repetição

## 3.1 - Primos num Intervalo

Faça um programa para imprimir todos os números primos num intervalo entre A e B.

Entrada	Saída
50 150	53 59 .. 149

# Solução: Bruna Borges

```
int num1, num2, i, count;
scanf("%d%d", &num1, &num2);
if(num1 < 1 || num2 > 10000 || num1 >= num2)
    return 0;

while(num1 <= num2){
    count = 0;
    for(i=1; i <= num1; i++){
        if(num1 % i == 0)
            count++;
    }
    if(count == 2)
        printf("\n%d", num1);
    num1++;
}
```

## 3.2 - Série e Arrays - Fibonacci - Vetor

$$\text{fib}(0) = 0$$

$$\text{fib}(1) = 1$$

$$\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2), \text{ para } n \geq 2$$

Entrada	Saída
0	0
2	1
6	8
-1 // valor negativo para finalizar	

### Série de Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21...



# Solução: Mateus Aranha

```
int fib[1000];  
int contador=0, x=0;  
fib[0]=0;  
fib[1]=1;  
  
for(contador=2; contador <= 1000; contador++){  
    fib[contador] = fib[contador-1] + fib[contador-2];  
}  
while(x >= 0 && x <= 1000){  
    scanf("%d", &x);  
    if(x >= 0 && x <= 1000)  
        printf("%d\n", fib[x]);  
}  
return 0;
```

## 3.6 - Matriz Oposta

Entrada

$$A = \begin{bmatrix} 2 & -3 \\ -1 & 4 \end{bmatrix}$$

Oposta

$$-A = \begin{bmatrix} -2 & 3 \\ 1 & -4 \end{bmatrix}$$

Resposta

$$\begin{bmatrix} -2 & 3 \\ 1 & -4 \end{bmatrix} + \begin{bmatrix} 2 & -3 \\ -1 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Entrada	Saída
2 //qtd linhas 2 //qtd colunas 1 1 1 1	-1 -1 -1 -1

# Solução: Gabriel Nascimento

```
int nl, nc, a, b;  
scanf("%i %i", &nl, &nc);  
int matriz[nl][nc];  
  
for (a = 0 ; a < nl ; a++) {  
    for (b = 0 ; b < nc ; b++)  
        scanf("%i", &matriz[a][b]);  
}  
  
for (a = 0 ; a < nl ; a++) {  
    for (b = 0 ; b < nc; b++)  
        printf("%i ", (-1) * matriz[a][b]);  
    printf("\n");  
}
```

Dúvidas?