

# Laços encaixados

---

Disciplina de Programação de Computadores I  
Universidade Federal de Ouro Preto

# Agenda

---

- Laços encaixados
- Funções com laços encaixados



# Exercício 1

---

- Imprima um quadrado formado por asteriscos na tela.

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

# Exercício 1: Resposta (lado com 4 asteriscos)

```
int main (int argc, char const *argv[])
{
    int linha, coluna;

    for(linha = 1; linha <= 4; ++linha)
    {
        for(coluna = 1; coluna <= 4; ++coluna)
        {
            printf(" * ");
        }
        printf("\n");
    }
    return 0;
}
```

linha	coluna
1	1
	2
	3
	4
2	1
	2
	3
	4
...	...

# Exercício 1: Resposta (lado variável)

```
int main (int argc, char const *argv[]){
    int linha, coluna, lados;

    printf("Quantos lados tem o quadrado?\n");
    scanf("%d", &lados);

    for(linha = 1; linha <= lados; ++linha) {
        for(coluna = 1; coluna <= lados; ++coluna) {
            printf(" * ");
        }
        printf("\n");
    }
    return 0;
}
```

linha	coluna
1	1
	2
	...
	lados
2	1
	2
	...
	lados
...	...

## Exercício 2: Triângulo retângulo invertido

---

- Imprima um triângulo retângulo invertido formado por asteriscos na tela.

\* \* \* \*

\* \* \*

\* \*

\*

# Código: Triângulo retângulo invertido (base = 4\*)

```
int main (int argc, char const *argv[]) {
    int linha, coluna;

    for(linha = 1; linha <= 4; ++linha)
    {
        for(coluna = 1; coluna <= 4 - linha + 1;
            ++coluna)
        {
            printf(" * ");
        }
        printf("\n");
    }
    return 0;
}
```

linha	coluna	4 - linha + 1
1	1	4
	2	
	3	
	4	
2	1	3
	2	
	3	
3	1	2
	2	
4	1	1

## Exercício 3: Procedimentos com laços encaixados

---

- Codifique um procedimento sem parâmetros que imprima as tabuadas de 1 a 10 utilizando laços encaixados.
- Apenas este procedimento **deverá** ser chamado na função main.



## Exercício 4: Procedimentos com laços encaixados

---

- Codifique uma função com a assinatura  
`void resulta_em(int numero),`  
em que `numero` pode ser um inteiro positivo ou negativo,  
que imprima todos os valores `x` e `y` (inclusive negativos)  
tais que:  
$$x * y = \text{numero}.$$

# Exercício 5

---

- Codifique uma função com a assinatura  
`int potencia(int base, int expoente)`  
que calcule a potência  $\text{base}^{\text{expoente}}$ .
- Não utilize a biblioteca `math.h` ou qualquer outra biblioteca.
- Utilizando esta função, crie um programa que imprima as 11 primeiras potências de 2 ( $2^0, 2^1, \dots, 2^{10}$ ), de 4 ( $4^0, 4^1, \dots, 4^{10}$ ), de 6 ( $6^0, 6^1, \dots, 6^{10}$ ) e de 8 ( $8^0, 8^1, \dots, 8^{10}$ ).

# Referências Bibliográficas

---

- Material de aula do Prof. Ricardo Anido, da UNICAMP:  
<http://www.ic.unicamp.br/~ranido/mc102/>
- Material de aula da Profa. Virgínia F. Mota:  
<https://sites.google.com/site/virginiaferm/home/disciplinas>
- DEITEL, P; DEITEL, H. *C How to Program*. 6a Ed. Pearson, 2010.