

MAC2166 - Introdução à Computação

Prof. Dr. Helder Oliveira



Agenda

- **Laços aninhados (repetições encaixadas)**
- **Exercícios**

Laços Encaixados

- Tanto os comandos de execução condicional quanto os de repetição podem ser “encaixados”, ou seja, um comando de repetição pode ser colocado dentro de outra repetição, como por exemplo:

```
while condição_1:  
    # bloco do while 1  
    while condição_2:  
        # bloco do while 2  
        while condição_3:  
            # bloco do while 3  
        # instruções após fim do while condição_3  
    # instruções após fim do while condição_2  
# instruções após fim do while condição_1
```

Exemplo 1

- Imprimir a tabuada de 1 a 10

```
1 linha = 1 #inicializa o laço externo
2 while linha <= 10:
3     coluna = 1 #inicializa o laço interno
4     while coluna <= 10:
5         print(linha*coluna, end="\t")
6         coluna += 1
7     print() # pula linha
8     linha += 1
```

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Exemplo 2

- Faça um programa que lê uma sequência de números inteiros não negativos, exibindo para cada número da sequência o seu fatorial

```
1 n = int(input("Digite um num da seq: "))
2 while n >= 0:
3     fat = 1 #inicializa o laço interno
4     cont = 2
5     while cont <= n:
6         fat *= cont
7         cont += 1
8     print(n, "! = ", fat, sep="") # imprime resultado
9     n = int(input("Digite um num da seq: "))
```

Exercício

- Escreva um programa que leia um número n e imprima um triângulo de asteriscos com n linhas.
- Ex para $n=3$:

*

**

Exercício

- Escreva um programa que leia um número n e imprima um triângulo de asteriscos com n linhas.

```
1 n = int(input("Digite o número de linhas: "))
2
3 linha = 1
4 while linha <= n:
5     asteriscos = 1
6     while asteriscos <= linha:
7         print("*", end="")
8         asteriscos += 1
9     print()
10    linha += 1
```

Exercício

- Escreva um programa que imprima a tabela de multiplicação de 1 a 10.
- Ex:

$$1 \times 1 = 1$$

$$1 \times 2 = 2$$

...

$$10 \times 10 = 100$$

Exercício

- Escreva um programa que imprima a tabela de multiplicação de 1 a 10.

```
1 numero = 1
2 while numero <= 10:
3     multiplicador = 1
4     while multiplicador <= 10:
5         print(f"{numero} x {multiplicador} = {numero * multiplicador}")
6         multiplicador += 1
7     numero += 1
```

Exercício

- Escreva um programa que leia um número n e imprima uma pirâmide de números até n .
- Ex para $n = 4$:

1

12

123

1234

Exercício

- Escreva um programa que leia um número n e imprima uma pirâmide de números até n .

```
1 n = int(input("Digite o número de linhas: "))
2
3 linha = 1
4 while linha <= n:
5     numero = 1
6     while numero <= linha:
7         print(numero, end="")
8         numero += 1
9     print()
10    linha += 1
11
```

Exercício

- Mostrar uma pirâmide semelhante a abaixo, sendo que o maior valor da pirâmide é definido pelo usuário. Ex: $n=9$

9 8 7 6 5 4 3 2 1

8 7 6 5 4 3 2 1

7 6 5 4 3 2 1

6 5 4 3 2 1

5 4 3 2 1

4 3 2 1

3 2 1

2 1

1

Exercício

- Mostrar uma pirâmide semelhante a abaixo, sendo que o maior valor da pirâmide é definido pelo usuário. Ex: n=9

```
9 8 7 6 5 4 3 2 1
8 7 6 5 4 3 2 1
7 6 5 4 3 2 1
6 5 4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

```
1 # Solicita o valor de n ao usuário
2 n = int(input("Digite o valor de n: "))
3 # Inicializa a variável de controle para o maior valor da pirâmide
4 i = n
5 # Loop para imprimir as linhas da pirâmide
6 while i >= 1:
7     j = i
8     # Loop para imprimir os números de cada linha
9     while j >= 1:
10        print(j, end=" ")
11        j -= 1
12    print() # Quebra de linha para a próxima linha
13    i -= 1 # Diminui o valor de i para a próxima linha
```

Exercício

- Escreva um programa que leia um número n e imprima um quadrado de asteriscos com n linhas e n colunas.
- Para $n=4$

Exercício

- Escreva um programa que leia um número n e imprima um quadrado de asteriscos com n linhas e n colunas.

```
1 n = int(input("Digite o tamanho do quadrado: "))
2
3 linha = 1
4 while linha <= n:
5     coluna = 1
6     while coluna <= n:
7         print("*", end="")
8         coluna += 1
9     print()
10    linha += 1
```

Exercício

- Escreva um programa que leia um número n e imprima uma pirâmide de asteriscos invertida com n linhas.
- Ex para $n=4$

**

*

Exercício

- Escreva um programa que leia um número n e imprima uma pirâmide de asteriscos invertida com n linhas.

```
1 n = int(input("Digite o número de linhas: "))
2
3 linha = n
4 while linha >= 1:
5     asteriscos = 1
6     while asteriscos <= linha:
7         print("*", end="")
8         asteriscos += 1
9     print()
10    linha -= 1
```

Exercício

- Crie um programa que desenhe um triângulo de asteriscos (*) de forma crescente. O número de linhas deve ser informado pelo usuário. Em cada linha, o programa deve exibir a quantidade de asteriscos correspondente ao número da linha e depois completar a linha com o caractere +, até o número total de caracteres na linha ser igual ao número da linha mais o número total de linhas.
- Ex: Para $n = 5$

```
*++++  
**+++  
***++  
****+  
*****
```

Exercício

- Ex: Para $n = 5$

```
*++++
**+++
***++
****+
*****
```

```
1 # Solicita o número de linhas ao usuário
2 n = int(input("Digite o número de linhas: "))
3 linha = 1
4 while linha <= n:
5     asteriscos = 0
6     mais = 0
7     # Laço interno para contar a quantidade de asteriscos
8     while asteriscos < linha:
9         print("*", end="")
10        asteriscos += 1
11    # Laço interno para contar a quantidade de "+"
12    while mais < (n - linha):
13        print("+", end="")
14        mais += 1
15    # Quebra de linha ao final de cada linha
16    print()
17
18    # Passa para a próxima linha
19    linha += 1
```

Exercício

- Dado um número inteiro n , $n > 1$, imprimir sua decomposição em fatores primos, indicando também a multiplicidade de cada fator.
- A **decomposição em fatores primos** de um número é o processo de quebrá-lo em números primos que, multiplicados entre si, resultam no número original.
- Por exemplo, para $n=600$, a saída deverá ser:
 - i. fator 2 multiplicidade 3
 - ii. fator 3 multiplicidade 1
 - iii. fator 5 multiplicidade 2

Exercício

- Dado um número inteiro n , $n > 1$, imprimir sua decomposição em fatores primos, indicando também a multiplicidade de cada fator.

```
1 n = int(input(" Entre com o número: "))
2 div = 2
3 while n > 1:
4     mult = 0
5     while n % div == 0:
6         mult += 1
7         n = n // div
8     if mult > 0:
9         print("fator", div, "com multiplicidade", mult)
10    div += 1
```

Exercício

- Dado um número inteiro n ($n > 1$), e um número inteiro m ($m > 1$), faça um programa que calcula o resultado da soma:

$$S = \sum_{i=1}^n \sum_{j=1}^m \frac{i^2 * j}{3^i (j * 3^i + i * 3^j)}$$

Exercício

- Dado um número inteiro n ($n > 1$), e um número inteiro m ($m > 1$), faça um programa que calcula o resultado da soma:

```
1 n = int(input("Digite n: "))
2 m = int(input("Digite m: "))
3
4 soma = 0
5 i = 1
6 while i <= n:
7     j = 1
8     while j <= m:
9         # numerador:
10        num = i*i*j
11        # denominador:
12        den = (3**i)*(j*(3**i) + i*(3**j))
13        soma += num/den
14        j += 1
15    i += 1
16
17 print("Soma:", soma)
```

$$S = \sum_{i=1}^n \sum_{j=1}^m \frac{i^2 * j}{3^i(j * 3^i + i * 3^j)}$$

Exercício

- Sabe-se que cada número da forma n^3 é igual a soma de n ímpares consecutivos.
 - Exemplos: $1^3 = 1$, $2^3 = 3 + 5$, $3^3 = 7 + 9 + 11$, $4^3 = 13 + 15 + 17 + 19$.
 - Dado um número inteiro m , $m > 0$, determinar os ímpares consecutivos cuja soma é igual a n^3 , para n assumindo valores de 1 a m .

Exercício

- Dado um número inteiro m , $m > 0$, determinar os ímpares consecutivos cuja soma é igual a n^3 , para n assumindo valores de 1 a m .

```
1 n = int(input("Digite n para elevar ao cubo: "))
2 i = 0
3 soma = 0
4 achei = False
5 while soma < n**3 and not achei:
6     soma = 0
7     j = 1
8     impar = 2*i+1
9     while j <= n:
10         soma += impar
11         impar += 2
12         j += 1
13
14     if soma == n**3:
15         achei = True
16         primeiro_impar = 2*i+1
17     i += 1
```

```
18 if achei:
19     print("Soma de ímpares encontrada é: ", end="")
20     j = 1
21     valor = primeiro_impar
22     while j <= n:
23         print(valor, end=" ")
24         j += 1
25         valor += 2
26     print()
```

Dúvidas

