UTFPR - Universidade Tecnológica Federal do Paraná
DACOM - Departamento de Computação
BCC1001 :: Introdução à Programação

Lista de Exercícios no. 05 :: Laços/Loops Aninhados

Instruções Gerais

- Os exercícios são de resolução individual.
- Sempre tente resolver os exercícios por conta própria, mesmo aqueles que já tenham sido feitos pelo professor em aula.
- Crie um arquivo (ex: lista.py) e faça cada exercício em uma ou mais funções.
- Utilize a extensão .py e o editor VS Code (ou outro de sua preferência).
- **Não é permitido o uso de recursos da linguagem ou bibliotecas que ainda não foram abordados na disciplina até o momento da publicação desta lista.**

1. Escreva uma função que imprime a tabuada completa, de 1 a 10. Pense em uma solução que se beneficia do uso de laços aninhados.

   Função: **def mult_table()**

2. Dados dois parâmetros naturais m e n (linhas x colunas), escreva uma função que exibe um retângulo formado por caracteres '[]', com $m_x n$ caracteres.

   Função: **def rect(m: int, n: int)**

   Para m=3 e n=6:

   ```
   [][][][][][]
   [][][][][][]
   [][][][][][]
   ```

3. Dados dois parâmetros naturais m e n, escreva uma função que exibe um retângulo torcido, formado por caracteres '[]' e com $m_x n$ caracteres.

   Função: **def skewed_rect(m: int, n: int)**

   Para m=5 e n=4:

   ```
   [][][][]
     [][][][]
       [][][][]
         [][][][]
           [][][][]
   ```

4. Dados dois parâmetros naturais m e n, escreva uma função que exibe um retângulo torcido, formado por caracteres '[]' e com $m_x n$ caracteres.

   Função: **def skewed_rect2(m: int, n: int)**

   Para m=5 e n=4:

   ```
               [][][][]
             [][][][]
           [][][][]
         [][][][]
       [][][][]
   ```

5. Dados dois parâmetros naturais m e n, escreva uma função que exibe um retângulo formado por caracteres ‘X’ intercalados com ‘-’, tendo n caracteres ‘X’ de largura. As linhas devem estar contidas entre colchetes.

   Função: **def xbox(m: int, n: int)**

   Exemplo: m=3 e n=6
   ```
   [X-X-X-X-X-X]
   [X-X-X-X-X-X]
   [X-X-X-X-X-X]
   ```

6. Escreva uma função que desenha uma caixa de tamanho $m_x n$, com estilo igual ao mostrado no exemplo abaixo.

   Função: **def box(m: int, n: int)**

   ```
   Informe as dimensões: 5 10
   +--------+
   |        |
   |        |
   |        |
   +--------+
   ```

7. Escreva uma função que exibe um triângulo retângulo formado por caracteres ‘[]’, com m caracteres de altura.

   Função: **def right_triangle(m: int)**

   Para m=5:
   ```
   [][][][][]
     [][][][]
       [][][]
         [][]
           []
   ```

8. Escreva uma função que imprime um Triângulo de Floyd de m linhas. Observe o padrão numérico:
   Função: **def floyd(m: int)**

   Para m=6:
   ```
   1
   2 3
   4 5 6
   7 8 9 10
   11 12 13 14 15
   16 17 18 19 20 21
   ```

9. Escreva uma função que exibe um triângulo centralizado formado por caracteres ‘[]’, com m caracteres de altura.

   Função: **def centered_triangle(m: int, n: int)**

   Para m=6:
   ```
       []
     [][]
   [][][]
 [][][][]
[][][][][]
   ```

10. Escreva uma função que exibe um MENU com 5 opções para executar operações. O MENU deve permitir operar sobre duas variáveis, A e B, que serão lidas por meio das opções [1] e [2]. Dica: utilize um laço que só termina quando a opção 5 for digitada. Observe o exemplo:

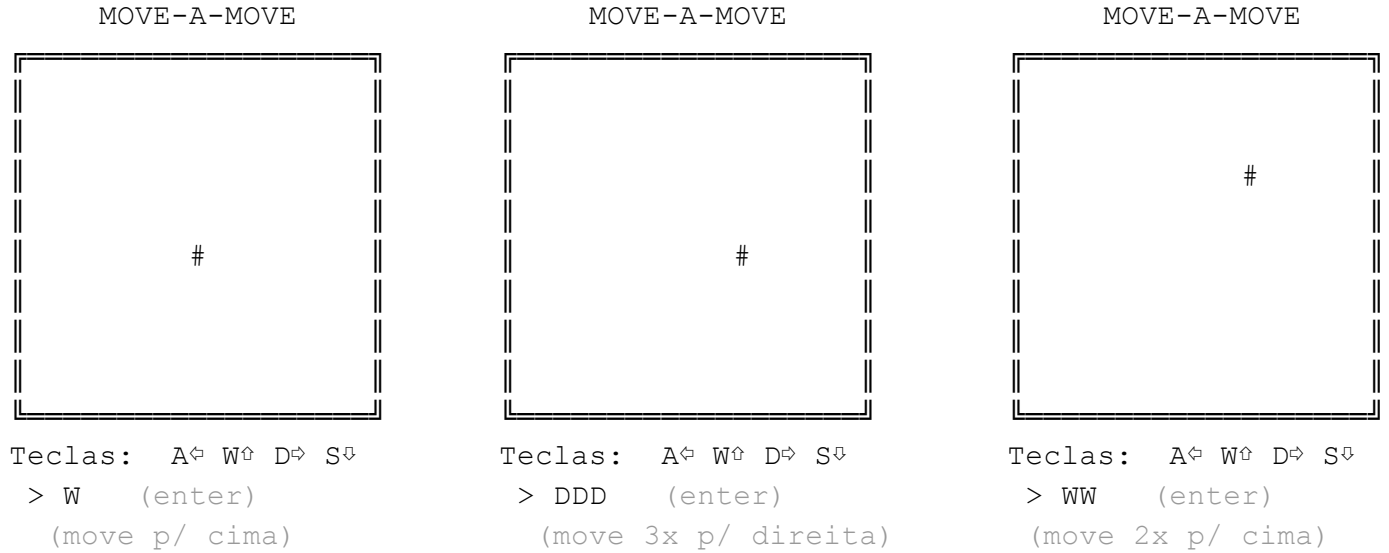
```
Função: def run_menu()
-------------------------------------------------
SUPREME SUM!                    A: 0    B: 0        // mostra A e B
-------------------------------------------------
1 - Set A                                           // ler da entrada
2 - Set B                                           // ler da entrada
3 - Show A+B                                        // soma e mostra
4 - Show AxB                                        // multiplica e mostra
5 - Exit
```

11. Escreva uma função que converte um número binário (int) para decimal (int). O processo envolve somar as multiplicações dos dígitos binários por potências de 2, com expoente crescente da direita para a esquerda. Observe o exemplo:

    Função: **def bin_to_dec(bin: int) -> int:**

    $10010_2 = 1x2^4 + 0x2^3 + 0x2^2 + 1x2^1 + 0x2^0 = 18_{10}$

12. Escreva uma função que imprime um tablado de tamanho **20_x10**, contendo um personagem '**#**', que deverá ser controlado pela entrada dos caracteres WASD. O personagem inicia no centro do trabalho - posição (10, 10). O código deve controlar a posição (linha x coluna) do personagem e alterá-la conforme os caracteres WASD forem informados. Note que, ao imprimir o tablado, é necessário considerar a posição em que se encontra o personagem para imprimí-lo na posição adequada. O código deve impedir que o personagem seja movido além dos limites do tablado. De forma geral, o programa consiste em:
    1. Ler caractere do teclado (W-A-S-D ou Q);
        a. Cada um dos caracteres WASD corresponde à uma direção;
        b. O caractere Q ("quit") deve encerrar o programa.
    2. De acordo com o caractere informado (WASD), deve alterar (adicionar/subtrair) a linha/coluna correspondente, de forma a reposicionar o personagem;
        a. W: linha -= 1, S: linha += 1, A: coluna -= 1, D: coluna += 1
    3. Reimprimir o tablado após cada entrada, conforme exemplo abaixo. Considerar a posição em que deve ser impresso o caractere do personagem ('#').



Função: **def move_a_move(m: int)**