

Programação de Computadores

Atividade 4

Resolva os exercícios abaixo utilizando a linguagem de programação Python. Elabore um arquivo ‘.py’ para cada um dos exercícios.

Após finalizar os exercícios, compacte todos os arquivos em um único arquivo ‘.zip’ e envie pelo BlackBoard na atividade correspondente.

1. Elaborar um programa Python para gerar a sequência de Fibonacci. O usuário deverá informar qual a quantidade de termos da sequência que quer ver. Exemplo: usuário escolhe o número 10 - saída: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55.
2. Desenvolva um programa em Python que exiba a tabuada completa de multiplicação NxN ($N = 1 \dots 10$) no formato de uma tabela organizada, indo de 1×1 , 1×2 até 10×10 , conforme o exemplo abaixo. O programa deve utilizar estruturas de repetição aninhadas para gerar a tabuada de forma eficiente. Exemplo:

Tabuada de 1 a 10:

 $1 \times 1 = 1 \quad 1 \times 2 = 2 \quad 1 \times 3 = 3 \quad \dots \quad 1 \times 10 = 10$

$2 \times 1 = 2 \quad 2 \times 2 = 4 \quad 2 \times 3 = 6 \quad \dots \quad 2 \times 10 = 20$

...

$10 \times 1 = 10 \quad 10 \times 2 = 20 \quad 10 \times 3 = 30 \quad \dots \quad 10 \times 10 = 100$

3. Crie um programa que funcione com base em laços de repetição, onde sempre após executar uma tarefa ele irá voltar para a parte inicial até que seja pressionado “0”, além disso o mesmo deve receber a entrada de dois números inteiros que irão ser utilizados no programa, veja o exemplo a seguir:

Saída/Impressão:

CALCULADORA:

1- somar

2- subtrair

3- multiplicar

4- dividir

0- sair

Insira sua opção: “1”

Opcão – SOMAR

Insira o número desejado: “2”

Insira o próximo número: “2”

Resultado = 4
(voltando para o menu....)

CALCULADORA:

- 1- somar
- 2- subtrair
- 3- multiplicar
- 0- sair Insira sua opção: “0”

Até logo!....

4. Em uma eleição presidencial existem quatro candidatos. Os votos são informados por meio de código. Os códigos utilizados são:

- 1 , 2, 3, 4 - Votos para os respectivos candidatos
(você deve montar a tabela ex: 1 - Jose/ 2- João/etc)
- 5 - Voto Nulo
- 6 - Voto em Branco

Faça um programa que calcule e mostre:

- O total de votos para cada candidato;
- O total de votos nulos;
- O total de votos em branco;
- A percentagem de votos nulos sobre o total de votos;
- A percentagem de votos em branco sobre o total de votos. Para finalizar o conjunto de votos tem-se o valor zero.

5. Elabore um programa em Python que simule o jogo Jokenpô (Pedra, Papel ou Tesoura) com as seguintes funcionalidades:

- a. Interação com o Usuário:
 - i. Solicite ao jogador que escolha uma das opções: Pedra, Papel ou Tesoura.
 - ii. Valide a entrada para garantir que seja uma opção válida.
 - iii. Sorteio da Jogada do Computador:
- b. O programa deve sortear aleatoriamente a jogada do adversário (computador) entre Pedra, Papel ou Tesoura (Dica: Utilize a biblioteca random para sortear a jogada do computador)
- c. Regras e Resultados:
 - i. Compare as jogadas do usuário e do computador, aplicando as regras clássicas do Jokenpô:
 - 1. Pedra vence Tesoura
 - 2. Tesoura vence Papel
 - 3. Papel vence Pedra
 - ii. Exiba o resultado de cada rodada (vitória, derrota ou empate).
- d. Placar e Melhor de Cinco:

- i. Mantenha um placar atualizado com as pontuações do jogador e do computador.
 - ii. O jogo deve ser disputado em melhor de cinco rodadas, ou seja, encerre quando um dos participantes atingir 3 vitórias.
- e. Finalização e Replay:
- i. Ao término da partida, declare o vencedor final.
 - ii. Pergunte ao usuário se deseja jogar novamente. Em caso positivo, reinicie o placar e comece uma nova partida.

Exemplo de Saída:

--Rodada 1--

Você escolheu: Papel

Computador escolheu: Pedra

Resultado: Você venceu!

Placar: Você 1 x 0 Computador

--Rodada 2--

Você escolheu: Tesoura

Computador escolheu: Pedra

Resultado: Você perdeu!

Placar: Você 1 x 1 Computador