LISTA IV  
Aluno: Roberto da CostaProfessor: Marcio Marcelo PiferIFC – Instituto Federal Catarinense, Campus Araquari, SC.Curso redes, Turma RCB0203 – Programação I – T01(2019.1)

|  |
| --- |
|  |
|  | ESTA ATIVIDADE VALERÁ NOTA E QUEM A ENTREGAR TAMBÉM TERÁ PRESENÇA NA AULA |
|  | ADICIONAL (SÁBADO) DO DIA 15/06/2019.  Você verá como se trabalha com strings. |
|  | # Jogo da velha |
|  | # Programa Fila de banco |
|  | ############################################################################## |
|  |  |
|  | ''' |
|  |  |
|  |  |
|  |  |
|  | 1) Analise o programa abaixo. Execute-o e veja os resultados. |
|  | Você verá como se trabalha com strings. |
|  | ''' |
|  |  |
|  | # Programa Jogo da forca |
|  | palavra = input("Digite a palavra secreta:").lower().strip() |
|  | for x in range(100): |
|  | print() |
|  | digitadas = [] |
|  | acertos = [] |
|  | erros = 0 |
|  | while True: |
|  | senha = "" |
|  | for letra in palavra: |
|  | senha += letra if letra in acertos else "." |
|  | print(senha) |
|  | if senha == palavra: |
|  | print("Você acertou!") |
|  | break |
|  | tentativa = input("\nDigite uma letra:").lower().strip() |
|  | if tentativa in digitadas: |
|  | print("Você já tentou esta letra!") |
|  | continue |
|  | else: |
|  | digitadas += tentativa |
|  | if tentativa in palavra: |
|  | acertos += tentativa |
|  | else: |
|  | erros += 1 |
|  | print("Você errou!") |
|  | print("X==:==\nX : ") |
|  | print("X O " if erros >= 1 else "X") |
|  | linha2 = "" |
|  | if erros == 2: |
|  | linha2 = " | " |
|  | elif erros == 3: |
|  | linha2 = " \| " |
|  | elif erros >= 4: |
|  | linha2 = " \|/ " |
|  | print(f"X{linha2}") |
|  | linha3 = "" |
|  | if erros == 5: |
|  | linha3 += " / " |
|  | elif erros >= 6: |
|  | linha3 += " / \ " |
|  | print(f"X{linha3}") |
|  | print("X\n===========") |
|  | if erros == 6: |
|  | print("Enforcado!") |
|  | break |
|  |  |
|  | ''' |
|  | 2) Modifique o programa acima de forma a escrever a palavra secreta caso o jogador a perca. |
|  |  |
|  | palavra = input("Digite a palavra secreta:").lower().strip() |
|  | for x in range(100): |
|  | print() |
|  | digitadas = [] |
|  | acertos = [] |
|  | erros = 0 |
|  | while True: |
|  | senha = "" |
|  | for letra in palavra: |
|  | senha += letra if letra in acertos else "." |
|  | print(senha) |
|  | if senha == palavra: |
|  | print("Você acertou!") |
|  | break |
|  | tentativa = input("\nDigite uma letra:").lower().strip() |
|  | if tentativa in digitadas: |
|  | print("Você já tentou esta letra!") |
|  | continue |
|  | else: |
|  | digitadas += tentativa |
|  | if tentativa in palavra: |
|  | acertos += tentativa |
|  | else: |
|  | erros += 1 |
|  | print("Você errou!") |
|  |  |
|  | print("X==:==\nX : ") |
|  | print("X O " if erros >= 1 else "X") |
|  | linha2 = "" |
|  | if erros == 2: |
|  | linha2 = " | " |
|  | elif erros == 3: |
|  | linha2 = " \| " |
|  | elif erros >= 4: |
|  | linha2 = " \|/ " |
|  | print(f"X{linha2}") |
|  | linha3 = "" |
|  | if erros == 5: |
|  | linha3 += " / " |
|  | elif erros >= 6: |
|  | linha3 += " / \ " |
|  | print(f"X{linha3}") |
|  | print("X\n===========") |
|  | if erros == 6: |
|  | print("Enforcado!") |
|  | print("A palavra secreta é: ", palavra) |
|  |  |
|  | break |
|  |  |
|  |  |
|  |  |
|  |  |
|  | 3) Modifique o programa acima de forma a utilizar uma lista de palavras. No início, pergunte |
|  | um número e calcule o índice da palavra a se utilizar pela fórmula: |
|  | índice = (número \* 567) % len(lista\_de\_palavras). |
|  |  |
|  |  |
|  | 4) Modifique o programa de forma a utilizar uma lista de strings para desenhar o boneco da |
|  | forca. Você poderá utilizar uma lista para cada linha e organizá-las em uma lista de listas. |
|  | Em vez de controlar quando o imprimir cada parte, desenhe nessas listas, substituindo o |
|  | elemento a desenhar como mostrado no exemplo: |
|  |  |
|  | >>> linha = list("X------") |
|  | >>> linha |
|  | >>> ['X', '-', '-', '-', '-', '-', '-'] |
|  | >>> linha[6] = "|" |
|  | >>> linha |
|  | >>> ['X', '-', '-', '-', '-', '-', '|'] |
|  | >> "".join(linha) |
|  | X------| |
|  | ========================================================================================================================= |
|  | 5#Escreva um jogo da velha para dois jogadores. O jogo deve perguntar onde você que jogar |
|  | #e alternar entre os jogadores. A cada jogada verifique se a posição está livre. Verifique |
|  | #também quando um jogador venceu a partida. Um jogo da velha pode ser visto como uma lista |
|  | #de três (03) elementos, onde cada elemento é outra lista, também com três elementos. |
|  | #Exemplodo jogo: |
|  | velha=""" Posições |
|  | | | 7 | 8 | 9 |
|  | ---+---+--- ---+---+--- |
|  | | | 4 | 5 | 6 |
|  | ---+---+--- ---+---+--- |
|  | | | 1 | 2 | 3 |
|  | """ |
|  | #Onde cada posição pode ser vista como um número. Confira abaixo um exemplo das posições mapeadas |
|  | #para a mesma posição de seu teclado numérico: |
|  | # 7 | 8 | 9 |
|  | # ---+---+--- |
|  | # 4 | 5 | 6 |
|  | # ---+---+--- |
|  | # 1 | 2 | 3 |
|  |  |
|  | posições = [ |
|  | None, |
|  | (5, 1), # 1 |
|  | (5, 5), # 2 |
|  | (5, 9), # 3 |
|  | (3, 1), # 4 |
|  | (3, 5), # 5 |
|  | (3, 9), # 6 |
|  | (1, 1), # 7 |
|  | (1, 5), # 8 |
|  | (1, 9), # 9 |
|  | ] |
|  |  |
|  | ganho = [ |
|  | [ 1, 2, 3], # Linhas |
|  | [ 4, 5, 6], |
|  | [ 7, 8, 9], |
|  | [ 7, 4, 1], # Colunas |
|  | [ 8, 5, 2], |
|  | [ 9, 6, 3], |
|  | [ 7, 5, 3], # Diagonais |
|  | [ 1, 5, 9] |
|  | ] |
|  |  |
|  | tabuleiro = [] |
|  | for linha in velha.splitlines(): |
|  | tabuleiro.append(list(linha)) |
|  |  |
|  | jogador = "X" |
|  | jogando = True |
|  | jogadas = 0 |
|  | while True: |
|  | for t in tabuleiro: |
|  | print("".join(t)) |
|  | if not jogando: |
|  | break |
|  | if jogadas == 9: |
|  | print("Deu velha! Ninguém ganhou.") |
|  | break |
|  | jogada = int(input("Digite a posição a jogar de 1 a 9 (jogador %s):" % jogador)) |
|  | if jogada<1 or jogada>9: |
|  | print("Posição inserida inválida") |
|  | continue |
|  |  |
|  | if tabuleiro[posições[jogada][0]][posições[jogada][1]] != " ": |
|  | print("Posição inserida ocupada."); |
|  | continue |
|  |  |
|  | tabuleiro[posições[jogada][0]][posições[jogada][1]] = jogador |
|  |  |
|  | for p in ganho: |
|  | for x in p: |
|  | if tabuleiro[posições[x][0]][posições[x][1]] != jogador: |
|  | break |
|  | else: |
|  | print("O jogador %s ganhou (%s): "%(jogador, p)) |
|  | jogando = False |
|  | break |
|  | jogador = "X" if jogador == "O" else "O" |
|  | jogadas +=1 |
|  | =============================================================================================== |
|  | 6) Analise o código do programa abaixo. |
|  |  |
|  | # Programa: Simulação de uma fila de banco |
|  |  |
|  | último = 10 |
|  | fila = list(range(1, último + 1)) |
|  | while True: |
|  | print(f"\nExistem {len(fila)} clientes na fila") |
|  | print(f"Fila atual: {fila}") |
|  | print("Digite F para adicionar um cliente ao fim da fila,") |
|  | print("ou A para realizar o atendimento. S para sair.") |
|  | operação = input("Operação (F, A ou S):") |
|  | if operação == "A": |
|  | if len(fila) > 0: |
|  | atendido = fila.pop(0) |
|  | print(f"Cliente {atendido} atendido") |
|  | else: |
|  | print("Fila vazia! Ninguém para atender.") |
|  | elif operação == "F": |
|  | último += 1 # Incrementa o ticket do novo cliente |
|  | fila.append(último) |
|  | elif operação == "S": |
|  | break |
|  | else: |
|  | print("Operação inválida! Digite apenas F, A ou S!") |
|  |  |
|  | Altere o programa para que você possa trabalhar com vários comandos digitados de uma só vez. |
|  | Atualmente, apenas um comando pode ser inserido por vez. A alteração deve considerar a operação |
|  | como uma string. |
|  | Exemplo: |
|  | AAAFFFS significaria três atendimentos, três chegadas de novos clientes e, finalmente, a saída |
|  | do programa. |
|  | ''' |