Questão 02

-Linguagem Natural

1. Início do Algoritmo
2. Solicitar a cadeia de caractere
3. Verificar se a cadeia está vazia ou não
4. Mostrar Verdadeiro se estiver vazia ou falso se não estiver
5. Fim do Algoritmo

-Linguagem Estruturada

1. **Função ÉVAZIOOUNAO**
2. Parâmetros: **N** tipo caractere
3. Declare **V** tipo inteiro
4. **V <-** Évazio(**N**)
5. Retorne **V**
6. **Fim ÉVAZIOOUNAO**
7. **Início Algoritmo**
8. Declare **X** tipo caractere
9. Leia **X**
10. Mostre **évazioounao**(X)
11. **Fim Algoritmo**

Questão 03

-Linguagem Natural

1. Início do Algoritmo
2. Solicitar as duas cadeias de caractere
3. Verificar se uma cadeia está contida na outra
4. Mostrar a posição inicial de uma cadeia dentro da outra se estiver contida ou mostrar -1 se não estiver.
5. Fim do Algoritmo

-Linguagem Estruturada

1. **Função S1CONTEMS2**
2. Parâmetros: **S1, S2** tipo caractere
3. Declare **C** tipo caractere
4. **C** <- Contém (**S1, S2**)
5. Retorne **C**
6. **Fim S1CONTEMS2**
7. **Início Algoritmo**
8. Declare **P1**, **P2** tipo caractere
9. Leia **P1, P2**
10. Mostre **s1contems2** (P1, P2)
11. **Fim Algoritmo**

Questão 04

-Linguagem Natural

1. Início do Algoritmo
2. Solicitar a cadeia de caracteres
3. Inverter a cadeia de caracteres
4. Fim do Algoritmo

-Linguagem Estruturada

1. **Função INVERSÃO**
2. Parâmetros: **N** tipo caractere
3. Declarar **C** tipo inteiro
4. PARA **C <-** Comprimento(**N**) ATÉ 1 PASSO -1 FACA
   4.1. MOSTRE(Repetir(**N**), **C,** 1)
5. FIMPARA
6. Retorne **C**
7. **Fim INVERSÃO**
8. **Início Algoritmo**
9. Declare **P** tipo caractere
10. Leia **P**
11. Mostre **inversão**(P)
12. **Fim Algoritmo**

Questão 07

-Linguagem Natural

1. Início do Algoritmo
2. Solicitar a temperatura em Farenheit
3. Converter a temperatura em Farenheit para Centígrados
4. Mostrar a temperatura em Centígrados
5. Fim do Algoritmo

-Linguagem Estruturada

1. **Função CONVERSOR**
2. Parâmetros: **F** tipo real
3. Declarar **C** tipo real
4. **C <-** 5*(**F** - 32) / 9
5. Retorne **C**
6. **Fim CONVERSOR**
7. **Início Algoritmo**
8. Declare **TF** tipo real
9. Leia **TF**
10. Mostre **conversor** (TF)
11. **Fim Algoritmo**

Questão 08

-Linguagem Natural

1. Início do Algoritmo
2. Solicitar a quantidade de notas e moedas
3. Calcular o valor total das notas
4. Calcular o valor total das moedas
5. Somar o valor das notas e das moedas
6. Mostrar a soma das notas e das moedas
7. Fim do Algoritmo

-Linguagem Estruturada

1. **Função CONTADINHEIRO**
2. Parâmetros: **M** e **N** tipo vetor
3. **M**[6] →{0.01, 0.05, 0.10, 0.25, 0.50, 1.00}
4. **N**[7] → {2, 5, 10, 20, 50, 100, 200}
5. Declare **TotM**[6], **TotN**[7] tipo vetor
6. Declare **QuantM**, **QuantN**, **C** tipo inteiro
7. Declare **SomaM**, **SomaN**, **Total** tipo real
8. PARA **C <-** 1 ATE 6 FACA
   - 8.1. LEIA(**QuantM**)
   - 8.2. **TotM**[**C**] **<- M**[**C**] * **QuantM**
   - 8.3. PARA **C <-** 1 ATE 6 FACA
     - 8.3.1 **SomaM <- SomaM** + **TotM[C]**
   - 8.4. FIMPARA
9. FIMPARA
10. PARA **C <-** 1 ATE 7 FACA
    - 10.1. LEIA(**QuantN**)
    - 10.2. **TotN**[**C**] **<- N**[**C**] * **QuantN**
    - 10.3. PARA **C <-** 1 ATE 7 FACA
      - **10.3.1 SomaN <- SomaN + TotN[C]**
    - 10.4. FIMPARA
11. FIMPARA
12. **Total <- SomaM + SomaN**
13. **Fim CONTADINHEIRO**
14. **Início Algoritmo**
15. Declare **QM** e **QN** tipo inteiro
16. LEIA QM e QN
17. Mostre **CONTADINHEIRO** (**QM**, **QN**)
18. **Fim Algoritmo**

Questão 10

-Linguagem Natural

1. Início do Algoritmo
2. Solicitar as coordenadas dos pontos que deseja calcular a distancia
3. Calcular a distância entre os pontos
4. Mostrar a distância entre os pontos apresentados
5. Fim do Algoritmo

-Linguagem Estruturada

1. **Função DISTANCIAENTREPONTOS**
2. Parâmetros: **a1, a2, b1, b2** tipo inteiro
3. Declarar **D** tipo real
4. **D <-** RaizQ((**b1 - a1**)^2 + (**b2 - a2**)^2))
5. Retorne **D**
6. **Fim DISTANCIAENTREPONTOS**
7. **Início Algoritmo**
8. Declare **XP1, YP1, XP2, YP2** tipo inteiro
9. Leia **XP1, YP1, XP2, YP2**
10. Mostre **distanciaentrepontos(XP1, YP1, XP2, YP2)**
11. **Fim Algoritmo**