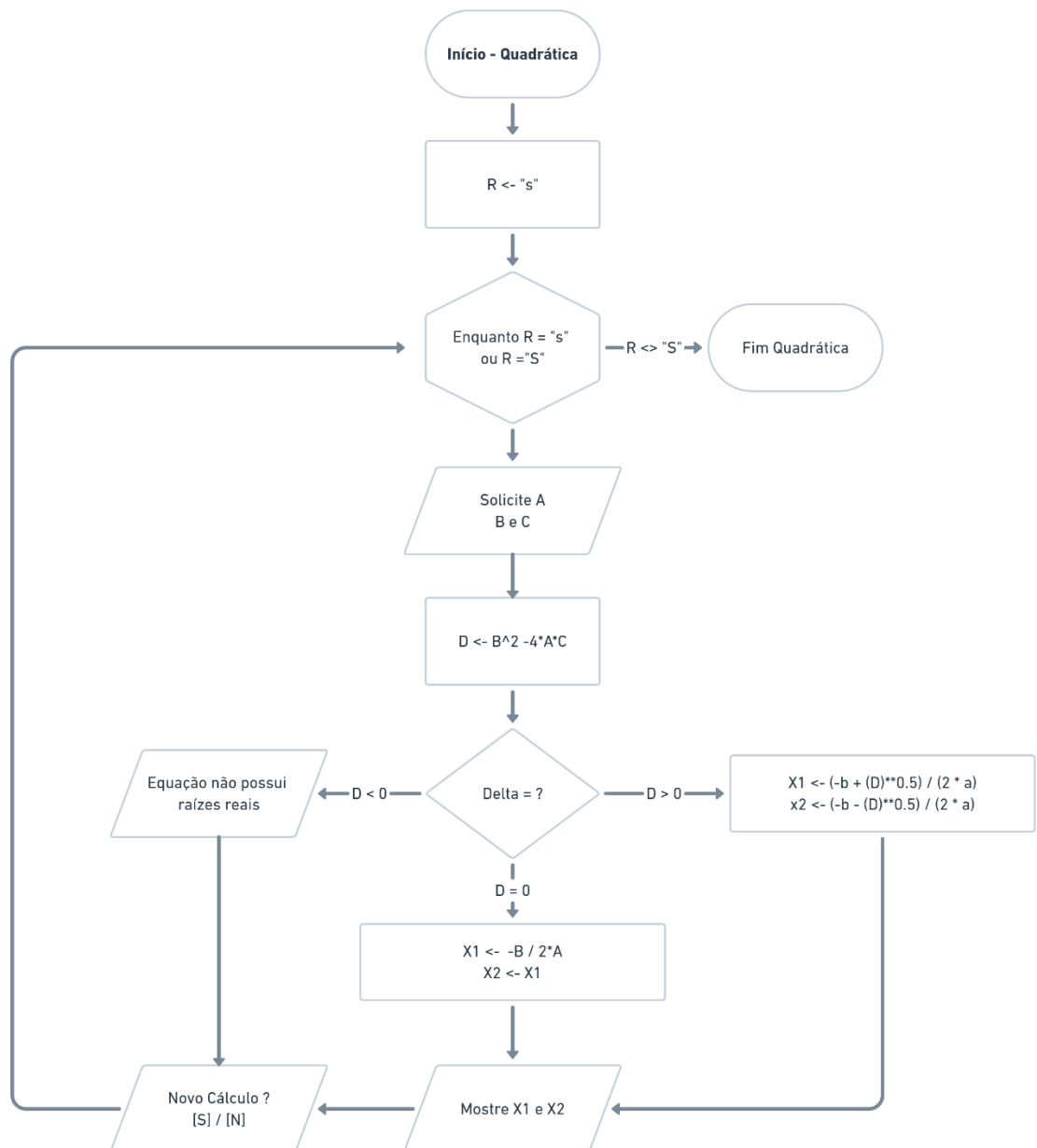


Questão 1

1. Início função quadrática (Algoritmo)
2. Defina $R \leftarrow "s"$
3. Enquanto $R = "s"$ ou $R = "S"$, faça:
 - 3.1. Escreva "Digite A"
 - 3.2. Leia a
 - 3.3. Escreva "Digite B"
 - 3.4. Leia b
 - 3.5. Escreva "Digite C"
 - 3.6. Leia c
 - 3.7. $D \leftarrow B^2 - 4 \cdot A \cdot C$
 - 3.8. Se $D > 0$ Então:
 - 3.8.1. Calcule $x_1 = (-b + (D)^{0.5}) / (2 \cdot a)$
 - 3.8.2. Calcule $x_2 = (-b - (D)^{0.5}) / (2 \cdot a)$
 - 3.8.3. Mostre X_1 e X_2
 - 3.9. Senão:
 - 3.10. Se $D = 0$ então:
 - 3.10.1. $X_1 \leftarrow -B / 2 \cdot A$
 - 3.10.2. $X_2 \leftarrow X_1$
 - 3.10.3. Mostre x_1 e x_2
 - 3.11. Senão:
 - 3.12. Se $D < 0$ Então:
 - 3.12.1. Escreva "Equação não possui raízes reais"
 - 3.13. Fim Se
 - 3.14. Escreva "Novo cálculo? [S]/[N]"
 - 3.15. Leia R
4. Fim enquanto
5. Fim quadrática

1. Início função quadrática (Linguagem natural)
2. Declare uma variável $R \leftarrow "S"$
3. Enquanto essa variável for igual a "s" ou "S", execute as operações
 - 3.1 Peça pelos valores de A, B e C
 - 3.2 Calcule o valor de delta por: $D \leftarrow B^2 - 4 \cdot A \cdot C$
 - 3.3 Se $D > 0$ então a equação possui duas raízes distintas
 - 3.3.1 Calcule X_1 pela formula: $x_1 = (-b + (D)^{0.5}) / (2 \cdot a)$

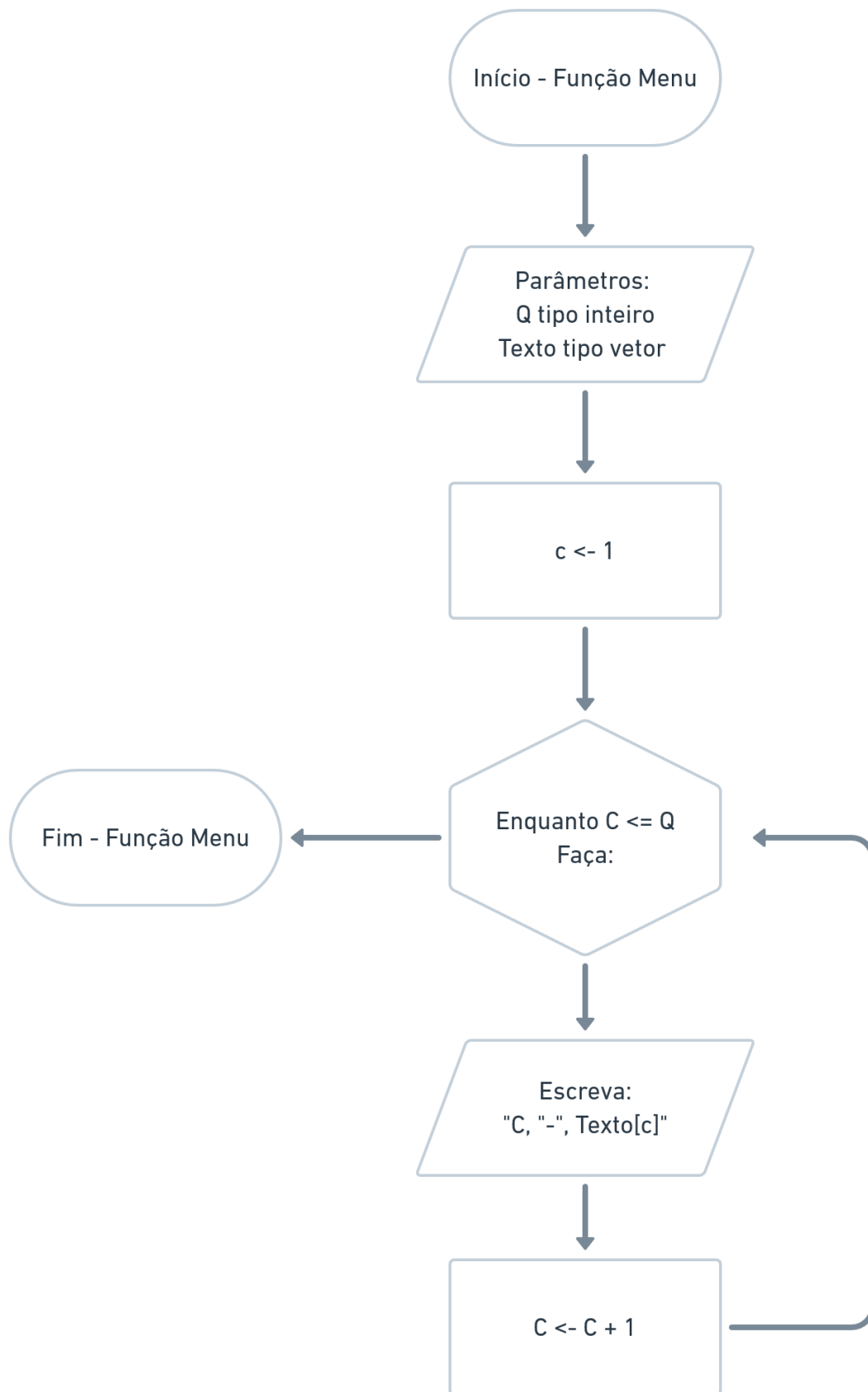
- 3.3.2 Calcule X2 pela formula: $x_2 = (-b - (D)^{0.5}) / (2 * a)$
- 3.3.3 Mostre X1 e X2
- 3.4 SE delta for igual a 0, ambas as raízes são iguais
 - 3.4.1 Calcule X1 por $-B / 2 * A$
 - 3.4.2 $X_2 \leftarrow X_1$
 - 3.4.3 Mostre X1 e X2
- 3.5 Se Delta for menor que 0, não existem raízes reais
 - 3.5.1 Escreva “Equação não possui raízes reais”
- 3.6 Mostre a mensagem “Novo cálculo? [S]/[N]”
- 3.7 Leia R
4. Fim do enquanto
5. Fim



Questão 3

1. Função Menu
2. Parâmetros: Q, tipo inteiro e Texto, do tipo vetor
3. $C \leftarrow 1$
4. Enquanto $C \leq Q$ faça:
 - a. Escreva Contador, “-”, Texto[C]
 - b. $C \leftarrow C + 1$
5. Mostre “S – Sair”
6. Fim Menu

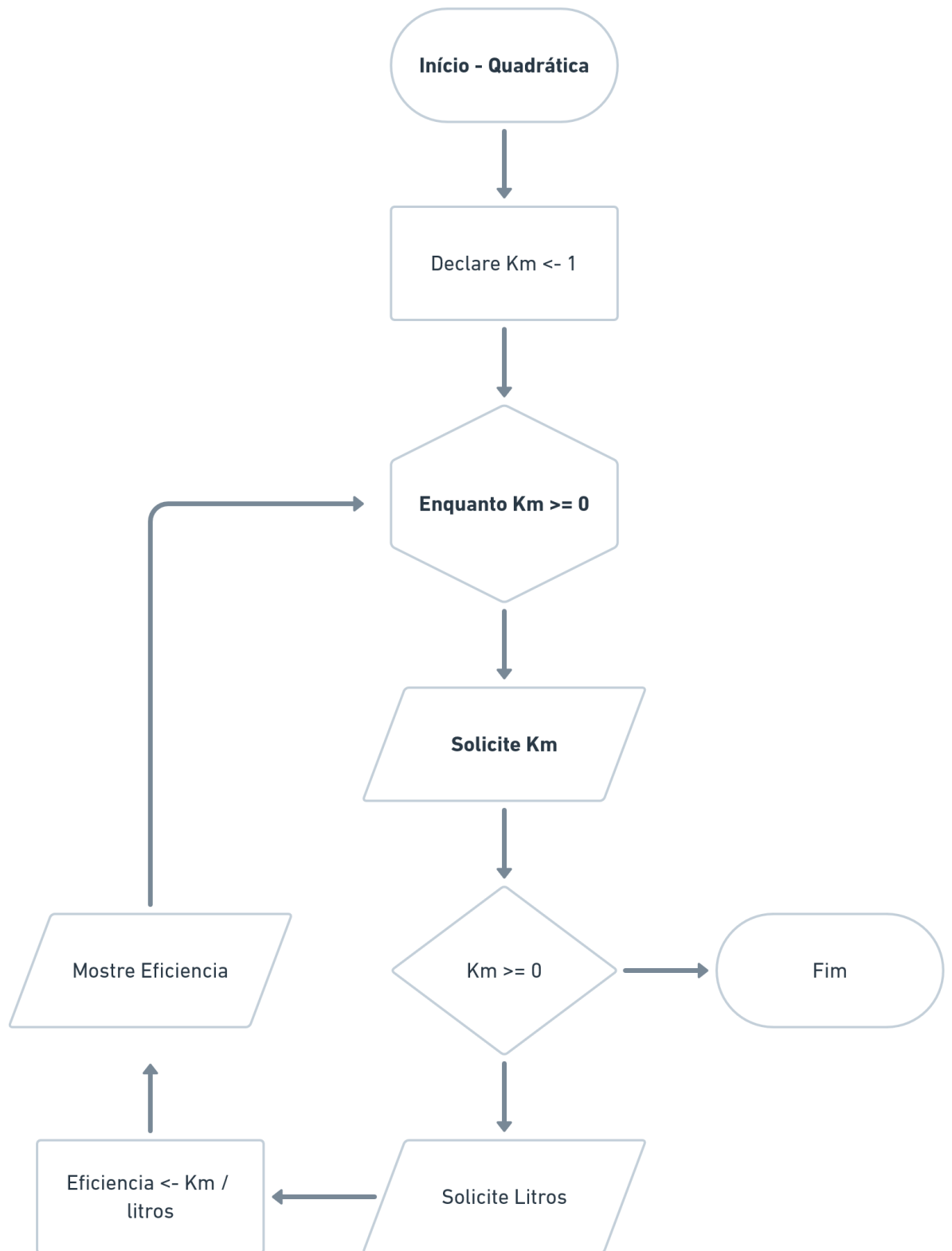
1. Função menu (linguagem natural)
2. Parâmetros: Q, do tipo inteiro e Texto do tipo vetor
3. Crie uma variável C e atribua 1
4. Enquanto $C \leq Q$, faça:
 - a. Escreva Contador, “-”, Texto[C]
 - b. Some 1 ao contador ($C \leftarrow C + 1$)
5. Mostre “S – Sair”
6. Fim Menu



Questão 4

1. Início Km/l (Algoritmo)
2. Declare Km <- 1
3. Enquanto Km > 0 faça:
 - 3.1 Escreva “Quantos quilômetros foram percorridos?”
 - 3.2 Leia Km
 - 3.3 Se Km < 0 então:
 - 3.3.1 continue
 - 3.4 Escreva “Qual foi o total de litros abastecidos?”
 - 3.5 Leia litros
 - 3.6 Eficiência <- Km / litros
 - 3.7 Escreva “Eficiência: “ Eficiência.
4. Fim

1. Início Km/ l (linguagem natural)
2. Declare uma variável Km <- 1
3. Enquanto Km não for negativo, faça:
 - 3.1 Solicite Km
 - 3.2 Se Km for menor que 0, finalize o programa
 - 3.3 Se não for:
 - 3.4. Solicite Litros
 - 3.5 Calcule Eficiência <- Km / litros
 - 3.6 Mostre Eficiência
4. Fim Enquanto
5. Fim

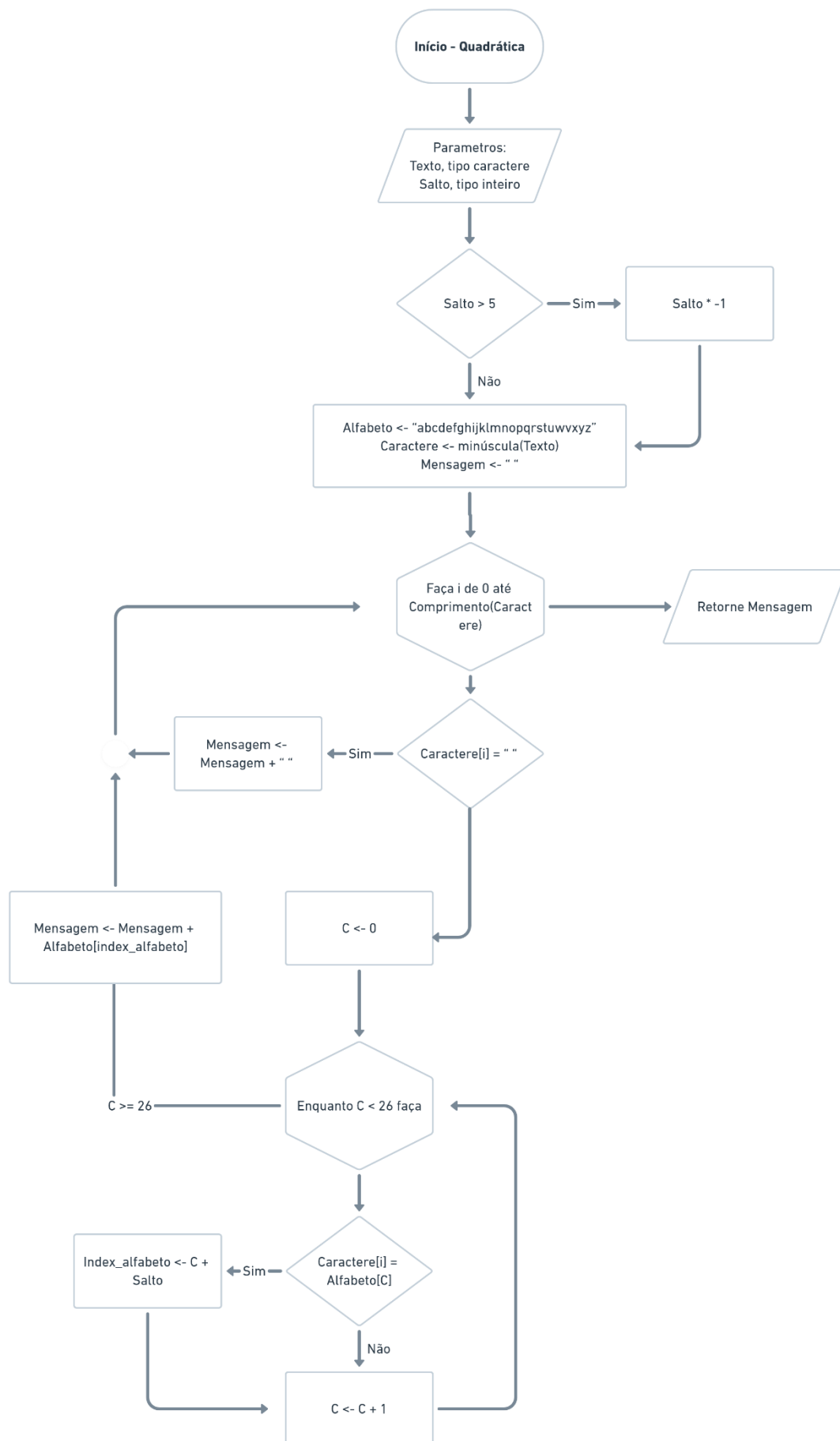


Questão 5

1. Início função criptografia
2. Parâmetros: Texto, tipo caractere. Salto, tipo inteiro
3. Se Salto > 5 então:
 - 3.1 Salto <- Salto * -1
4. Alfabeto <- "abcdefghijklmnopqrstuvwxyz"
5. Caractere <- minúscula(Texto)
6. Mensagem <- ""
7. Faça i de 0 até Comprimento(Caractere)
 - 7.1 Se Caractere[i] = "" então:
 - 7.1.1. Mensagem <- Mensagem + ""
 - 7.1.2. Continue
 - 7.2. Senão:
 - 7.3. C <- 0
 - 7.4. Enquanto C < 26 faça:
 - 7.4.1. Se Caractere[i] = Alfabeto[C] então:
 - 7.4.1.1 Index_alfabeto <- C + Salto
 - 7.4.2. C <- C + 1
 - 7.5. Fim Enquanto
 - 7.6. Mensagem <- Mensagem + Alfabeto[index_alfabeto]
8. Retorne Mensagem

1. Início função criptografia (linguagem natural)
2. Parâmetros: Texto e salto. O primeiro do tipo caractere e o segundo inteiro
3. Se salto for maior que 5, então faça:
 - 4.1. Salto <- Salto * -1
4. Defina Alfabeto <- "abcdefghijklmnopqrstuvwxyz"
5. Caractere <- minúscula(Texto)

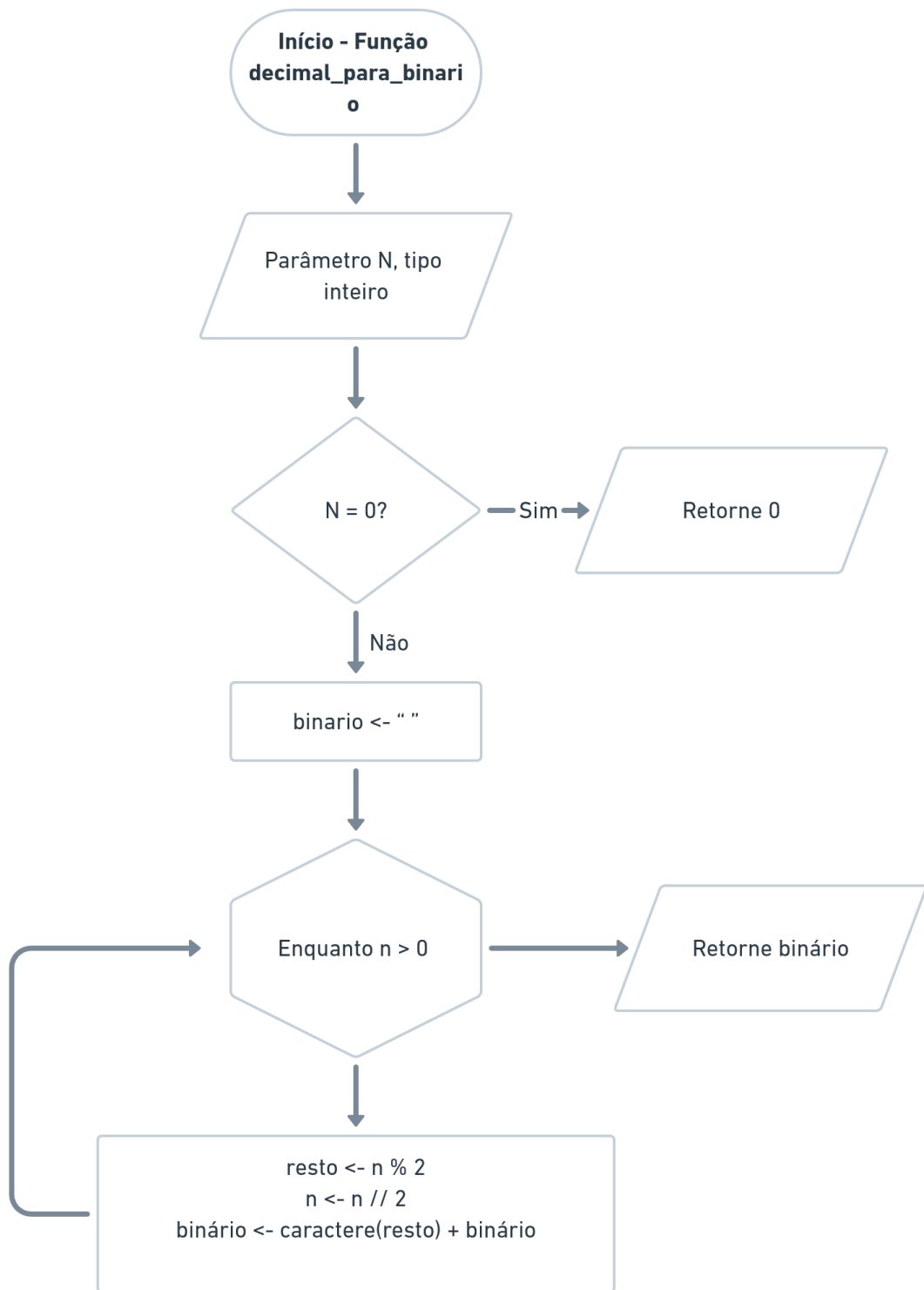
6. Mensagem <- " "
7. Faça as seguintes operações de i 0 até comprimento(caractere)
8. Se Caractere[i] for igual a " ", então:
 - 8.1. Mensagem <- Mensagem + " "
 - 8.2. Continue
9. Defina uma variável C <- 0
10. Enquanto C < 26, faça:
 - 10.1 Se Caractere[i] = Alfabeto[C] então
 - 10.1.1. Index_alfabeto <- C + Salto
 - 10.2 C < C + 1
11. Fim enquanto
12. Mensagem <- Mensagem + Alfabeto[index_alfabeto]
13. Fim faça
14. Retorne Mensagem



Questão 6

1. Início função decimal_para_binario (algoritmo)
2. Parâmetro: n, tipo inteiro
3. Se $n = 0$ então:
 - 3.1 binario <- "0"
4. Senão:
5. Binário <- ""
6. Enquanto $n > 0$ faça:
 - 6.1. resto <- $n \% 2$
 - 6.2. $n <- n // 2$
 - 6.3. binário <- caractere(resto) + binário
7. Retorne binário
8. Fim decimal_para_binario

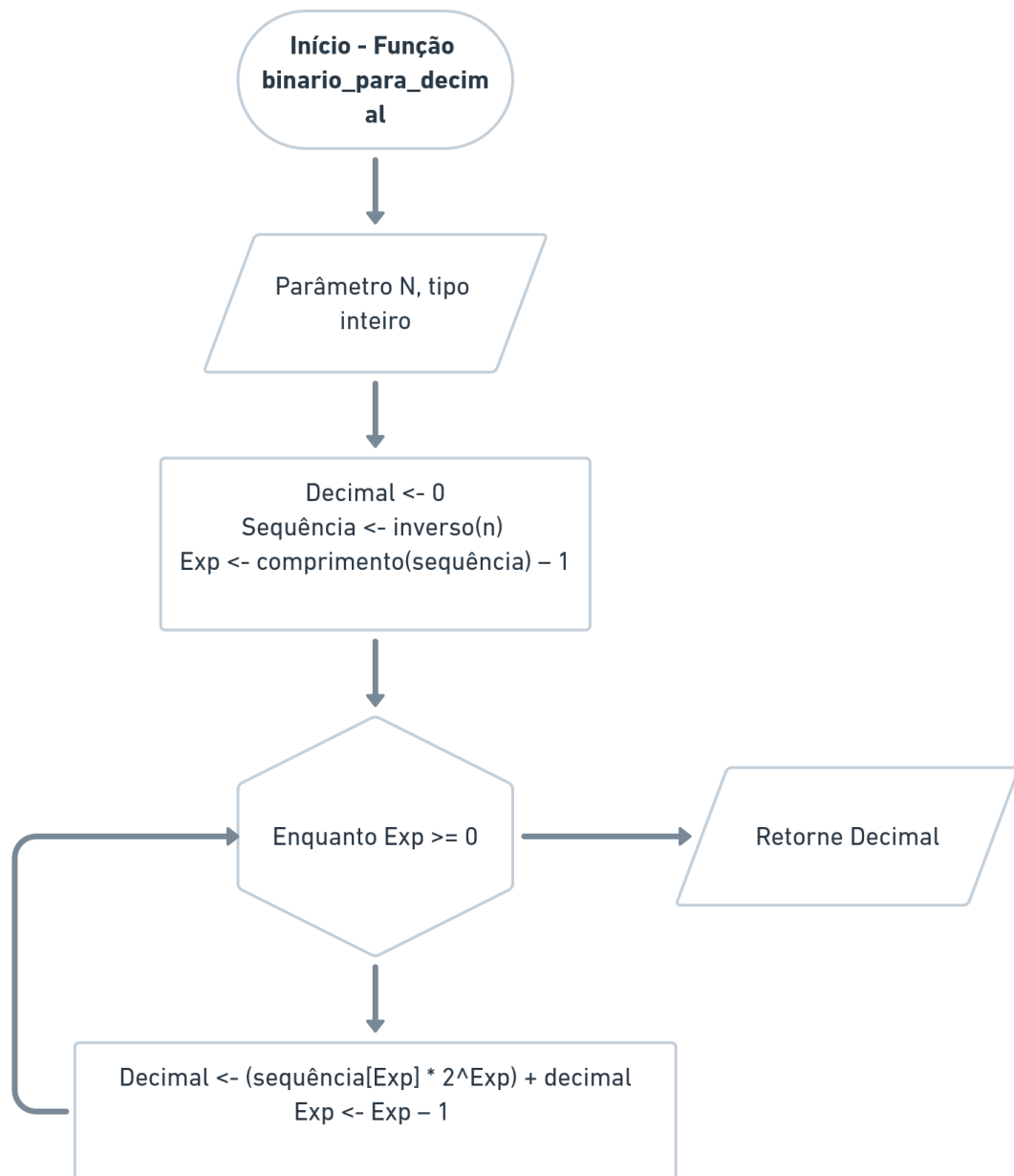
1. Início função decimal_para_binario (Linguagem natural)
2. Parâmetro: n, tipo inteiro
3. Se n for igual a zero, retorne zero e finalize
4. Senão:
5. Uma variável Binário recebe ""
6. Enquanto n for maior que 0, faça:
 - a. Uma variável Resto receberá o módulo da divisão $n / 2$ ($n \% 2$)
 - b. n recebe o valor inteiro de sua divisão por dois ($n // 2$)
 - c. Binário recebe o resto(como cadeia de caractere) + binário
7. Retorne binário
8. Fim decimal_para_binario



Questão 7

1. Início binário_para_decimal (algoritmo)
2. Parâmetro: n, tipo inteiro
3. Decimal <- 0
4. Sequência <- inverso(n)
5. Exp <- comprimento(sequência) – 1
6. Enquanto Exp >= 0 faça:
 - 6.1. Decimal <- (sequência[Exp] * 2^{Exp}) + decimal
 - 6.2. Exp <- Exp – 1
7. Retorne Decimal

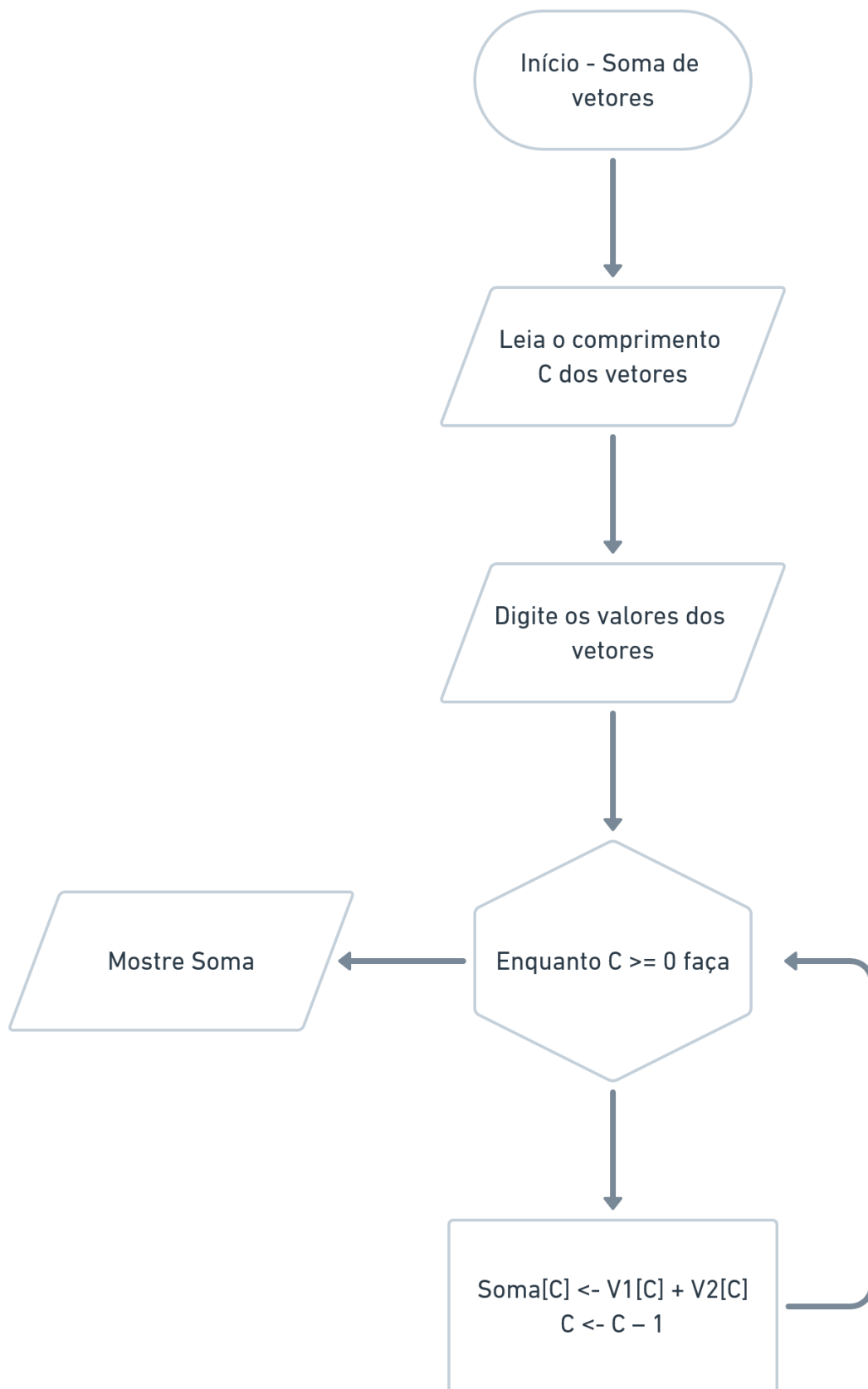
1. Início binário_para_decimal (Linguagem natural)
2. Parâmetro: n, tipo inteiro
3. Uma variável Decimal recebe 0
4. A variável sequência recebe o inverso de n como caractere
5. A variável Exp recebe o comprimento da sequência – 1
6. Enquanto Exp for maior ou igual a 0, faça:
 - a. Decimal recebe a (posição Exp da sequência vezes 2 elevado à Exp) + Decimal
 - b. Exp recebe Exp – 1
7. Retorne Decimal



Questão 8

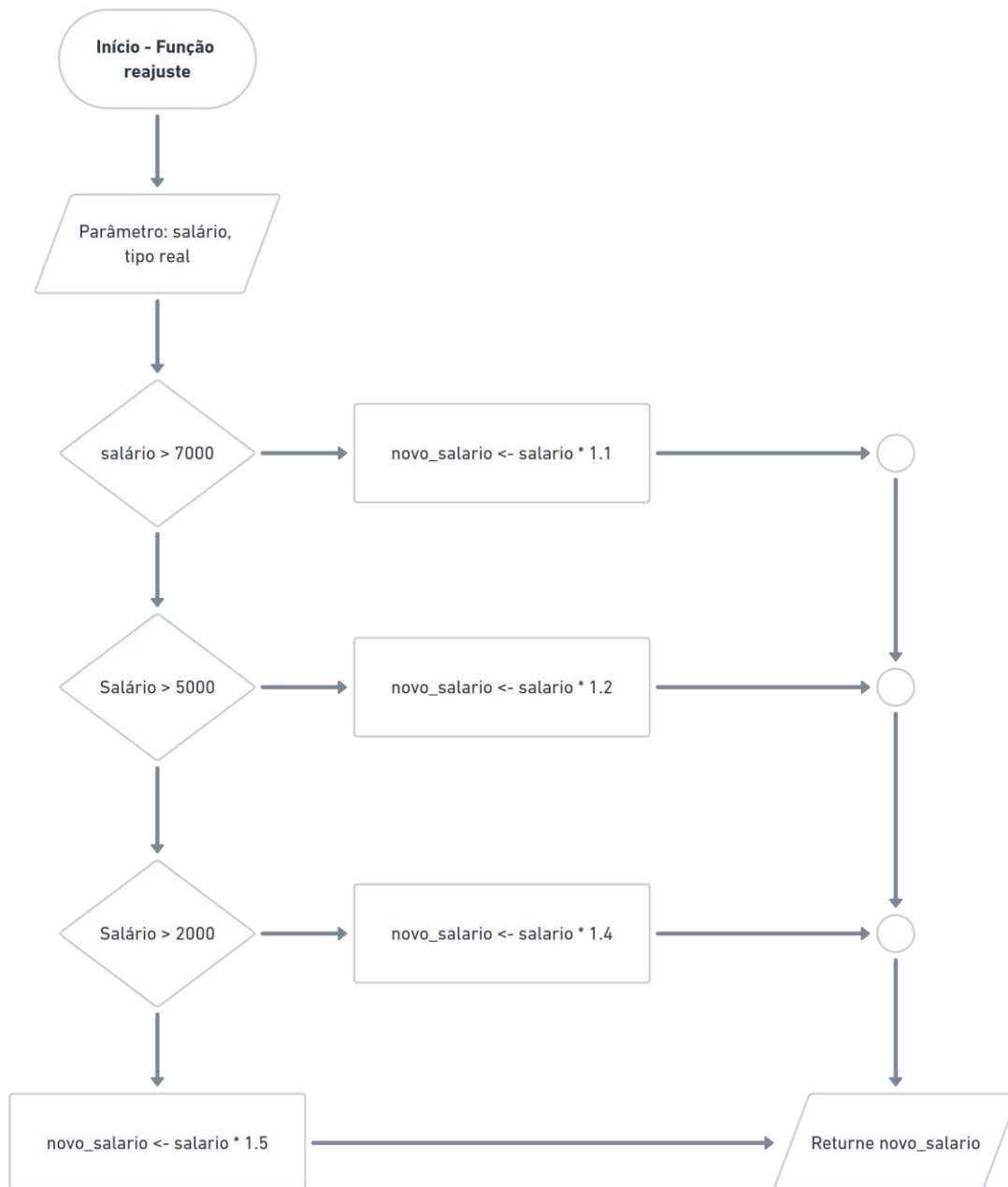
1. Início Soma de vetores
2. Escreva “Digite o comprimento dos vetores”
3. Leia C
4. Escreva “Agora digite os valores”
5. Leia V1, V2
6. Enquanto C >= 0 Faça
 - a. $Soma[C] \leftarrow V1[C] + V2[C]$
 - b. $C \leftarrow C - 1$
7. Fim Enquanto
8. Mostre Soma
9. Fim Soma de vetores

1. Início Soma de vetores (Linguagem natural)
2. Escreva “Digite o comprimento dos vetores”
3. Leia e salve o valor numa variável C
4. Escreva “Digite os valores de cada vetor”
5. Leia V1 e V2
6. Enquanto C for maior ou igual a 0, faça o seguinte:
 - a. Um vetor Soma na posição C recebe a soma de V1 e V2, ambos na posição C
 - b. Subtraia 1 de C
7. Fim do Enquanto
8. Mostre Soma
9. Fim Soma de vetores



Questão 9

1. Início função Reajuste
2. Parâmetro: Salário, tipo real
3. Se salário \leq 2000.00:
 - 3.1.1 novo_salario = salario * 1.5
 - 3.1.2 retorne novo_salario
4. Senão:
5. Se salário \leq 5000.00:
 - 5.1.1 novo_salario = salario * 1.4
 - 5.1.2 retorne novo_salario
6. Senão:
7. Se salário \leq 7000.00:
 - 7.1.1 novo_salario = salario * 1.2
 - 7.1.2 retorne novo_salario
8. Senão:
 - 8.1.1 novo_salario = salario * 1.1
 - 8.1.2 retorne novo_salario



1. Início função reajuste (linguagem natural)
2. Parâmetro: salário, tipo real
3. Se o salário for maior que 7000, então:
 - 3.1. Multiplique salário por 1.1, atribua ao novo_salario
 - 3.2. Retorne novo_salario
4. Se salário estiver entre 5000 e 7000, então:
 - 4.1. Multiplique salário por 1.2, atribua ao novo_salario
 - 4.2. Retorne novo_salario

5. Se salário estiver entre 2000 e 5000, então:
 - 5.1. Multiplique salário por 1.4, atribua ao novo_salario
 - 5.2. Retorne novo_salario
6. Se salário for menor ou igual a 2000, então:
 - 6.1. Multiplique salário por 1.5, atribua ao novo_salario
 - 6.2. Retorne novo_salario
7. Fim reajuste

Questão 10

1. Início Lanchonete
2. Declare pedido, total, refri, misto_quente, pao_de_queijo, suco = 0, 0, 0, 0, 0, 0
3. Enquanto pedido >= 0:
 - a. Escreva “[100] Misto quente R\$4,50”
 - b. Escreva “[101] Refrigerante R\$5,00”
 - c. Escreva “[102] Pão de queijo R\$2,00”
 - d. Escreva “[103] Suco R\$6,00”
 - e. Leia pedido
 - f. Se pedido < 0 então
 - i. Continue
 - g. Escreva “Quantos?”
 - h. Leia quantidade
 - i. Se pedido = 100:
 - i. $\text{total} \leftarrow 4.5 * \text{quantidade} + \text{total}$
 - ii. $\text{misto_quente} \leftarrow \text{quantidade} + \text{misto_quente}$
 - j. Se pedido = 101:
 - i. $\text{total} \leftarrow 5 * \text{quantidade} + \text{total}$
 - ii. $\text{refri} \leftarrow \text{quantidade} + \text{refri}$
 - k. SE pedido = 102:
 - i. $\text{total} \leftarrow 2 * \text{quantidade} + \text{total}$
 - ii. $\text{pao_de_queijo} \leftarrow \text{quantidade} + \text{pao_de_queijo}$
 - l. Se pedido = 103:
 - i. $\text{total} \leftarrow 6 * \text{quantidade} + \text{total}$

- ii. `suco <- quantidade + suco`
- m. Fim enquanto
- n. Escreva "Misto quentes: " `misto_quente`
- o. Escreva "Refrigerantes: " `refri`
- p. Escreva "Pão de queijos: " `pao_de_queijo`
- q. Escreva "Sucos: " `suco`
- r. Escreva "Total: R\$" `total`
- s. Fim

1. Início Lanchonete
2. Declare `pedido`, `total`, `refri`, `misto_quente`, `pao_de_queijo`, `suco`: todos iguais a 0
3. Enquanto `pedido` for igual ou menor a 0, faça:
 - a. Escreva "[100] Misto quente R\$4,50"
 - b. Escreva "[101] Refrigerante R\$5,00"
 - c. Escreva "[102] Pão de queijo R\$2,00"
 - d. Escreva "[103] Suco R\$6,00"
 - e. Leia `pedido`
4. Se `pedido` for menor que 0, finalize o loop
5. Senão:
6. Solicite e leia `quantidade`
7. Se `pedido` for igual a 100:
 - i. `total <- 4.5 * quantidade + total`
 - ii. `misto_quente <- quantidade + misto_quente`
8. Se `pedido` for igual a 101:
 - i. `total <- 5 * quantidade + total`
 - ii. `refri <- quantidade + refri`
9. Se `pedido` for igual a 102:
 - i. `total <- 2 * quantidade + total`
 - ii. `pao_de_queijo <- quantidade + pao_de_queijo`
10. Se `pedido` for igual a 103:
 - i. `total <- 6 * quantidade + total`
 - ii. `suco <- quantidade + suco`
11. Fim enquanto
12. Mostre todas as variáveis diferentes de 0
13. Fim Lanchonete

