

<pre>algoritmo "Avaliação da disciplina" var notatestes, notafichas, notaparticipacao, notafinal, nfinal_arred: <u>real</u> nome: <u>caracter</u> inicio escreval ("Digite o seu nome:") leia (nome) escreval ("Digite a nota do teste:") leia (notatestes) escreval ("Digite a nota da ficha:") leia(notafichas) escreval ("Digite a nota da participação:") leia (notaparticipacao) notafinal <- 0.5*notatestes + 0.4*notafichas + 0.1*notaparticipacao nfinal_arred <- int(notafinal + 0.5) escreval (nome, ", a sua nota final é de ", nfinal_arred:2:0, " valores.") se (nfinal_arred<5) entao escreval ("e tem a classificação de mau.") senao se (nfinal_arred<10) entao escreval ("e tem a classificação de insuficiente.") senao se (nfinal_arred<14) entao escreval ("e tem a classificação de suficiente.") senao se (nfinal_arred<18) entao escreval ("e tem a classificação de bom.") senao se (nfinal_arred<=20) entao escreval ("e tem a classificação de muito bom.") senao escreval ("e é inválida.") fimse fimse fimse fimse fimse fimalgoritmo</pre>	<pre>algoritmo "Avaliação da disciplina" var notatestes, notafichas, notaparticipacao, notafinal, nfinal_arred: <u>real</u> nome: <u>caracter</u> inicio escreval ("Digite o seu nome:") leia (nome) escreval ("Digite a nota do teste:") leia (notatestes) escreval ("Digite a nota da ficha:") leia(notafichas) escreval ("Digite a nota da participação:") leia (notaparticipacao) notafinal <- 0.5*notatestes + 0.4*notafichas + 0.1*notaparticipacao nfinal_arred <- int(notafinal + 0.5) escreval (nome, ", a sua nota final é de ", nfinal_arred:2:0, " valores.") se (nfinal_arred<6) entao escreval ("e tem a classificação de mau.") senao se (nfinal_arred<10) entao escreval ("e tem a classificação de insuficiente.") fimse se (nfinal_arred<14) entao escreval ("e tem a classificação de suficiente.") fimse se (nfinal_arred<18) entao escreval ("e tem a classificação de bom.") fimse se (nfinal_arred<=20) entao escreval ("e tem a classificação de muito bom.") senao escreval ("e é inválida.") fimse fimse // esta organização de blocos independentes dentro do primeiro IF // não serve para este tipo de problema visto as que as classificações não são // independentes entre si. Por exemplo uma nota de 13 valores é ao mesmo // tempo menor que 14, menor que 18 e menor que 20 e assim o resultado dirá // que a classificação é -ao mesmo tempo- suficiente, bom e muito bom, o que // não é correto. fimalgoritmo</pre>	<pre># Avaliação da disciplina # notatestes, notafichas, notaparticipacao, notafinal, nfinal_arred: float # nome: str # Início nome = input("Digite o seu nome: ") notatestes = float(input("Digite a nota dos testes: ")) notafichas = float(input("Digite a nota das fichas: ")) notaparticipacao = float(input("Digite a nota da participação: ")) notafinal = 0.5 * notatestes + 0.4 * notafichas + 0.1 * notaparticipacao nfinal_arred = int(notafinal + 0.5) print(nome, ", a sua nota final é de ", nfinal_arred, "valores.") if nfinal_arred < 6: print("E tem a classificação de mau.") else: if nfinal_arred < 10: print("E tem a classificação de insuficiente.") else: if nfinal_arred < 14: print("E tem a classificação de suficiente.") else: if nfinal_arred < 18: print("E tem a classificação de bom.") else: if nfinal_arred <= 20: print("E tem a classificação de muito bom.") else: print("E é inválida.") # Fim</pre>	<pre># Avaliação da disciplina # notatestes, notafichas, notaparticipacao, notafinal, nfinal_arred: float # nome: str # Início nome = input("Digite o seu nome: ") notatestes = float(input("Digite a nota dos testes: ")) notafichas = float(input("Digite a nota das fichas: ")) notaparticipacao = float(input("Digite a nota da participação: ")) notafinal = 0.5 * notatestes + 0.4 * notafichas + 0.1 * notaparticipacao nfinal_arred = int(notafinal + 0.5) print(nome, ", a sua nota final é de ", nfinal_arred, "valores.") if nfinal_arred < 6: print("E tem a classificação de mau.") else: if nfinal_arred < 10: print("E tem a classificação de insuficiente.") if nfinal_arred < 14: print("E tem a classificação de suficiente.") if nfinal_arred < 18: print("E tem a classificação de bom.") if nfinal_arred <= 20: print("E tem a classificação de muito bom.") else: print("E é inválida.") # esta organização de blocos independentes dentro do primeiro IF # não serve para este tipo de problema visto as que as classificações não são # independentes entre si. Por exemplo uma nota de 13 valores é ao mesmo # tempo menor que 14, menor que 18 e menor que 20 e assim o resultado dirá # que a classificação é -ao mesmo tempo- suficiente, bom e muito bom, o que # não é correto. # Fim</pre>
---	---	--	---

- 52-Explique qual seria a finalidade dos dois pseudocódigos/programas acima.
- 53-Experimente o primeiro pseudocódigo/programa (com SE/IF encadeados) e verifique que com funciona para todos os níveis de classificação.
- 54-Experimente o segundo pseudocódigo/programa (com SE/IF independentes) e verifique que não funciona devidamente.
- 55-O segundo pseudocódigo/programa apenas dá resultados certos para dois níveis de classificação. Quais são eles?
- 56-No segundo pseudocódigo/programa se introduzir notas para classificação 12 deveria ter como resultado “suficiente”. Isto acontece? Ou aparece ao mesmo tempo, suficiente, bom e muito bom? Faz sentido para um aluno com 12 aparecer como resultado suficiente, bom e muito bom?
- 57-Explique a razão para que não se resolva este tipo de problemas com SE/IF independentes, mas apenas com SE/IF encadeados.

<pre>algoritmo "É triângulo de que tipo?" var lado1,lado2,lado3:inteiro Inicio escreval("Introduza os 3 lados de um possível triângulo") leia(lado1) leia(lado2) leia(lado3) se (lado1+lado2>lado3) e (lado1+lado3>lado2) e (lado2+lado3>lado1) entao escreval("Os lados digitados formam um triângulo") se (lado1 = lado2) e (lado1 = lado3) e (lado2 = lado3) entao escreval("e é equilátero, isto é, os três lados são iguais.") senao se (lado1 <> lado2) e (lado1 <> lado3) e (lado2 <> lado3) entao escreval("e é escaleno, isto é, os três lados são diferentes.") senao se ((lado1 = lado2) e (lado1 <> lado3)) ou ((lado1 = lado3) e (lado1 <> lado2)) ou ((lado2 = lado3) e (lado2 <> lado1)) entao escreval("e é isósceles, isto é, dois dos seus lados têm a") escreval("mesma medida que é diferente do restante lado.") fimse fimse fimse senao escreval("Os lados inseridos não formam um triângulo.") fimse fimalgoritmo</pre>	<pre>algoritmo "É triângulo de que tipo?" var lado1,lado2,lado3:inteiro Inicio escreval("Introduza os 3 lados de um possível triângulo") leia(lado1) leia(lado2) leia(lado3) se (lado1+lado2>lado3) e (lado1+lado3>lado2) e (lado2+lado3>lado1) entao escreval("Os lados digitados formam um triângulo") se (lado1 = lado2) e (lado1 = lado3) e (lado2 = lado3) entao escreval("e é equilátero, isto é, os três lados são iguais.") fimse se (lado1 <> lado2) e (lado1 <> lado3) e (lado2 <> lado3) entao escreval("e é escaleno, isto é, os três lados são diferentes.") fimse se ((lado1 = lado2) e (lado1 <> lado3)) ou ((lado1 = lado3) e (lado1 <> lado2)) ou ((lado2 = lado3) e (lado2 <> lado1)) entao escreval("e é isósceles, isto é, dois dos seus lados têm a") escreval("mesma medida que é diferente da do restante lado.") fimse senao escreval("Os lados inseridos não formam um triângulo.") fimse fimalgoritmo</pre>	<pre># é triângulo de que tipo? # lado1, lado2, lado3: inteiro # Início print("Introduza os 3 lados de um possível triângulo") lado1 = int(input()) lado2 = int(input()) lado3 = int(input()) if (lado1 + lado2 > lado3) and (lado1 + lado3 > lado2) and (lado2 + lado3 > lado1): print("Os lados digitados formam um triângulo,") if (lado1 == lado2) and (lado1 == lado3) and (lado2 == lado3): print("que é equilátero, isto é, os três lados são iguais.") else: if (lado1 != lado2) and (lado1 != lado3) and (lado2 != lado3): print("que é escaleno, isto é, os três lados são diferentes.") else: if ((lado1 == lado2) and (lado1 != lado3)) or ((lado1 == lado3) and (lado1 != lado2)) or ((lado2 == lado3) and (lado2 != lado1)): print("que é isósceles, isto é, dois dos seus lados têm a") print("mesma medida que é diferente do restante lado.") else: print("Os lados inseridos não formam um triângulo.") # Fim</pre>	<pre># é triângulo de que tipo? # lado1, lado2, lado3: inteiro # Início print("Introduza os 3 lados de um possível triângulo") lado1 = int(input()) lado2 = int(input()) lado3 = int(input()) if (lado1 + lado2 > lado3) and (lado1 + lado3 > lado2) and (lado2 + lado3 > lado1): print("Os lados digitados formam um triângulo,") if (lado1 == lado2) and (lado1 == lado3) and (lado2 == lado3): print("que é equilátero, isto é, os três lados são iguais.") if (lado1 != lado2) and (lado1 != lado3) and (lado2 != lado3): print("que é escaleno, isto é, os três lados são diferentes.") if ((lado1 == lado2) and (lado1 != lado3)) or ((lado1 == lado3) and (lado1 != lado2)) or ((lado2 == lado3) and (lado2 != lado1)): print("que é isósceles, isto é, dois dos seus lados têm a") print("mesma medida que é diferente do restante lado.") # ao contrário do exercício das Classificações, esta organização de blocos # independentes dentro do primeiro IF pode ser utilizado neste tipo de problema # visto que as condições para os tipos de triângulo são independentes por si, # seja, um triângulo não pode ser ao mesmo tempo equilétero, escaleno e isósceles # (ao contrário do exercício das classificações em que uma nota, 13 por exemplo, # é ao mesmo tempo menor que 14, menor que 18 e menor que 20. else: print("Os lados inseridos não formam um triângulo.") # Fim</pre>
---	---	--	---

58-Explique qual seria a finalidade dos dois pseudocódigos/programas acima.

59-Experimente o primeiro pseudocódigo/programa (com SE/IF encadeados) e verifique que funciona para todos os tipos de triângulo.

60-Experimente o segundo pseudocódigo/programa (com SE/IF independentes) e verifique que funciona para todos os tipos de triângulo..

61-Explique a razão para que este tipo de problemas (ao contrário do da página anterior) possa ser resolvido quer com SE/IF encadeados, quer com SE/IF independentes. Na sua resposta compare com o exercício 57-.

<pre>algoritmo "Avaliação da disciplina" var notatestes, notafichas, notaparticipacao, notafinal, nfinal_arred: real nome: caracter inicio escreval ("Digite o seu nome:") leia (nome) escreval ("Digite a nota do teste:") leia (notatestes) escreval ("Digite a nota da ficha:") leia(notafichas) escreval ("Digite a nota da participação:") leia (notaparticipacao) notafinal <- 0.5*notatestes + 0.4*notafichas + 0.1*notaparticipacao nfinal_arred <- int(notafinal + 0.5) escreval (nome, ", a sua nota final é de ", nfinal_arred:2:0, " valores.") se (nfinal_arred<5) entao escreval ("e tem a classificação de mau.") senao se (nfinal_arred<10) entao escreval ("e tem a classificação de insuficiente.") senao se (nfinal_arred<14) entao escreval ("e tem a classificação de suficiente.") senao se (nfinal_arred<18)entao escreval ("e tem a classificação de bom.") senao se (nfinal_arred<=20)entao escreval ("e tem a classificação de muito bom.") senao escreval ("e é inválida.") fimse fimse fimse fimse fimse fimse fimse fimalgoritmo</pre>	<pre>algoritmo "Avaliação da disciplina" var notatestes, notafichas, notaparticipacao, notafinal, nfinal_arred: real nome: caracter inicio escreval ("Digite o seu nome:") leia (nome) escreval ("Digite a nota do teste:") leia (notatestes) escreval ("Digite a nota da ficha:") leia(notafichas) escreval ("Digite a nota da participação:") leia (notaparticipacao) notafinal <- 0.5*notatestes + 0.4*notafichas + 0.1*notaparticipacao nfinal_arred <- int(notafinal + 0.5) escreval (nome, ", a sua nota final é de ", nfinal_arred:2:0, " valores.") se (nfinal_arred<6) entao escreval ("e tem a classificação de mau.") senao se (nfinal_arred<10) entao escreval ("e tem a classificação de insuficiente.") fimse se (nfinal_arred<14) entao escreval ("e tem a classificação de suficiente.") fimse se (nfinal_arred<18) entao escreval ("e tem a classificação de bom.") fimse se (nfinal_arred<=20)entao escreval ("e tem a classificação de muito bom.") senao escreval ("e é inválida.") fimse fimse // esta organização de blocos independentes dentro do primeiro IF // não serve para este tipo de problema visto as que as classificações não são // independentes entre si. Por exemplo uma nota de 13 valores é ao mesmo // tempo menor que 14, menor que 18 e menor que 20 e assim o resultado dirá // que a classificação é -ao mesmo tempo- suficiente, bom e muito bom, o que // não é correto. Fimalgoritmo</pre>	<pre># Avaliação da disciplina # notatestes, notafichas, notaparticipacao, notafinal, nfinal_arred: float # nome: str # Início nome = input("Digite o seu nome: ") notatestes = float(input("Digite a nota dos testes: ")) notafichas = float(input("Digite a nota das fichas: ")) notaparticipacao = float(input("Digite a nota da participação: ")) notafinal = 0.5 * notatestes + 0.4 * notafichas + 0.1 * notaparticipacao nfinal_arred = int(notafinal + 0.5) print(nome, ", a sua nota final é de ", nfinal_arred, "valores.") if nfinal_arred < 6: print("E tem a classificação de mau.") else: if nfinal_arred < 10: print("E tem a classificação de insuficiente.") else: if nfinal_arred < 14: print("E tem a classificação de suficiente.") else: if nfinal_arred < 18: print("E tem a classificação de bom.") else: if nfinal_arred <= 20: print("E tem a classificação de muito bom.") else: print("E é inválida.") # Fim</pre>	<pre># Avaliação da disciplina # notatestes, notafichas, notaparticipacao, notafinal, nfinal_arred: float # nome: str # Início nome = input("Digite o seu nome: ") notatestes = float(input("Digite a nota dos testes: ")) notafichas = float(input("Digite a nota das fichas: ")) notaparticipacao = float(input("Digite a nota da participação: ")) notafinal = 0.5 * notatestes + 0.4 * notafichas + 0.1 * notaparticipacao nfinal_arred = int(notafinal + 0.5) print(nome, ", a sua nota final é de ", nfinal_arred, "valores.") if nfinal_arred < 6: print("E tem a classificação de mau.") else: if nfinal_arred < 10: print("E tem a classificação de insuficiente.") if nfinal_arred < 14: print("E tem a classificação de suficiente.") if nfinal_arred < 18: print("E tem a classificação de bom.") if nfinal_arred <= 20: print("E tem a classificação de muito bom.") else: print("E é inválida.") # esta organização de blocos independentes dentro do primeiro IF # não serve para este tipo de problema visto as que as classificações não são # independentes entre si. Por exemplo uma nota de 13 valores é ao mesmo # tempo menor que 14, menor que 18 e menor que 20 e assim o resultado dirá # que a classificação é -ao mesmo tempo- suficiente, bom e muito bom, o que # não é correto. # Fim</pre>
--	---	--	---

<pre>algoritmo "é triangulo de que tipo?" var lado1,lado2,lado3:inteiro Inicio escreval("Introduza os 3 lados de um possível triângulo") leia(lado1) leia(lado2) leia(lado3) se (lado1+lado2>lado3) e (lado1+lado3>lado2) e (lado2+lado3>lado1) entao escreval("Os lados digitados formam um triângulo") se (lado1 = lado2) e (lado1 = lado3) e (lado2 = lado3) entao escreval("e é equilátero, isto é, os três lados são iguais.") senao se (lado1 <> lado2) e (lado1 <> lado3) e (lado2 <> lado3) entao escreval("e é escaleno, isto é, os três lados são diferentes.") senao se ((lado1 = lado2) e (lado1 <> lado3)) ou ((lado1 = lado3) e (lado1 <> lado2)) ou ((lado2 = lado3) e (lado2 <> lado1)) entao escreval("e é isósceles, isto é, dois dos seus lados têm a") escreval("mesma medida que é diferente do restante lado.") fimse fimse fimse senao escreval("Os lados inseridos não formam um triângulo.") fimse fimalgoritmo</pre>	<pre>algoritmo "é triangulo de que tipo?" var lado1,lado2,lado3:inteiro Inicio escreval("Introduza os 3 lados de um possível triângulo") leia(lado1) leia(lado2) leia(lado3) se (lado1+lado2>lado3) e (lado1+lado3>lado2) e (lado2+lado3>lado1) entao escreval("Os lados digitados formam um triângulo") se (lado1 = lado2) e (lado1 = lado3) e (lado2 = lado3) entao escreval("e é equilátero, isto é, os três lados são iguais.") fimse se (lado1 <> lado2) e (lado1 <> lado3) e (lado2 <> lado3) entao escreval("e é escaleno, isto é, os três lados são diferentes.") fimse se ((lado1 = lado2) e (lado1 <> lado3)) ou ((lado1 = lado3) e (lado1 <> lado2)) ou ((lado2 = lado3) e (lado2 <> lado1)) entao escreval("e é isósceles, isto é, dois dos seus lados têm a") escreval("mesma medida que é diferente da do restante lado.") fimse senao escreval("Os lados inseridos não formam um triângulo.") fimse fimalgoritmo</pre>	<pre># é triangulo de que tipo? # lado1, lado2, lado3: inteiro # Início print("Introduza os 3 lados de um possível triângulo") lado1 = int(input()) lado2 = int(input()) lado3 = int(input()) if (lado1 + lado2 > lado3) and (lado1 + lado3 > lado2) and (lado2 + lado3 > lado1): print("Os lados digitados formam um triângulo,") if (lado1 == lado2) and (lado1 == lado3) and (lado2 == lado3): print("que é equilátero, isto é, os três lados são iguais.") else: if (lado1 != lado2) and (lado1 != lado3) and (lado2 != lado3): print("que é escaleno, isto é, os três lados são diferentes.") else: if ((lado1 == lado2) and (lado1 != lado3)) or ((lado1 == lado3) and (lado1 != lado2)) or ((lado2 == lado3) and (lado2 != lado1)): print("que é isósceles, isto é, dois dos seus lados têm a") print("mesma medida que é diferente do restante lado.") else: print("Os lados inseridos não formam um triângulo.") # Fim</pre>	<pre># é triangulo de que tipo? # lado1, lado2, lado3: inteiro # Início print("Introduza os 3 lados de um possível triângulo") lado1 = int(input()) lado2 = int(input()) lado3 = int(input()) if (lado1 + lado2 > lado3) and (lado1 + lado3 > lado2) and (lado2 + lado3 > lado1): print("Os lados digitados formam um triângulo,") if (lado1 == lado2) and (lado1 == lado3) and (lado2 == lado3): print("que é equilátero, isto é, os três lados são iguais.") if (lado1 != lado2) and (lado1 != lado3) and (lado2 != lado3): print("que é escaleno, isto é, os três lados são diferentes.") if ((lado1 == lado2) and (lado1 != lado3)) or ((lado1 == lado3) and (lado1 != lado2)) or ((lado2 == lado3) and (lado2 != lado1)): print("que é isósceles, isto é, dois dos seus lados têm a") print("mesma medida que é diferente do restante lado.") # ao contrário do exercício das Classificações, esta organização de blocos # independentes dentro do primeiro IF pode ser utilizado neste tipo de problema # visto que as condições para os tipos de triângulo são independentes por si, # seja, um triângulo não pode ser ao mesmo tempo equilétro, escaleno e isósceles # (ao contrário do exercício das classificações em que uma nota, 13 por exemplo, # é ao mesmo tempo menor que 14, menor que 18 e menor que 20. else: print("Os lados inseridos não formam um triângulo.") # Fim</pre>
---	---	--	---