

Iniciado em	domingo, 21 mai. 2023, 19:26
Estado	Finalizada
Concluída em	domingo, 21 mai. 2023, 21:52
Tempo empregado	2 horas 26 minutos
Avaliar	10,00 de um máximo de 10,00(100%)



Questão 1

Correto

Atingiu 1,00 de 1,00

Gêmeos Pitá e Goras

Pitá e Goras são irmãos gêmeos que amam matemática e gostam de aplicá-la em tudo o que vão fazer. Até mesmo na hora de comer eles utilizam números para definir o que vão jantar. Sua tarefa é ajudar os irmãos a implementarem um [programa](#) que os ajude a tomar a decisão do que irão jantar conforme o critério que eles estabeleceram. Para isso, os gêmeos decidiram que se a quantidade de minutos do relógio for um número par na hora que eles forem comer, então eles irão jantar arroz e feijão, mas se for um número ímpar eles irão fazer apenas um lanche.

A Entrada consiste de:

- Uma [variável](#) `M` do tipo inteiro representando os minutos do relógio.

A Saída deve apresentar:

- 1 linha contendo a frase **"Fome de comida! Queremos arroz e feijão"** para casos com números pares ou a frase **"Só um lanchinho cai bem!"** para os casos ímpares, sem as aspas conforme apresentado nos exemplos.

Observações:

- Para a [variável](#) `M` serão considerados apenas valores entre 0 e 60 inclusive.

Descrição dos Exemplos:

- No primeiro exemplo, a saída corresponde ao caso da entrada par conforme solicitado no enunciado.

For example:

Input	Result
0	Fome de comida! Queremos arroz e feijão
2	Fome de comida! Queremos arroz e feijão
15	Só um lanchinho cai bem!

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```
1 num = int(input())
2 res = num%2
3 if res == 0:
4     print("Fome de comida! Queremos arroz e feijão")
5 else:
6     print("Só um lanchinho cai bem!")
7
```



	Input	Expected	Got	
✓	0	Fome de comida! Queremos arroz e feijão	Fome de comida! Queremos arroz e feijão	✓
✓	2	Fome de comida! Queremos arroz e feijão	Fome de comida! Queremos arroz e feijão	✓
✓	15	Só um lanchinho cai bem!	Só um lanchinho cai bem!	✓
✓	25	Só um lanchinho cai bem!	Só um lanchinho cai bem!	✓
✓	40	Fome de comida! Queremos arroz e feijão	Fome de comida! Queremos arroz e feijão	✓
✓	59	Só um lanchinho cai bem!	Só um lanchinho cai bem!	✓

Passou em todos os teste! ✓

Para resolver essa questão atente para o uso da divisão pelo [módulo](#).

Correto

Notas para este envio: 1,00/1,00.



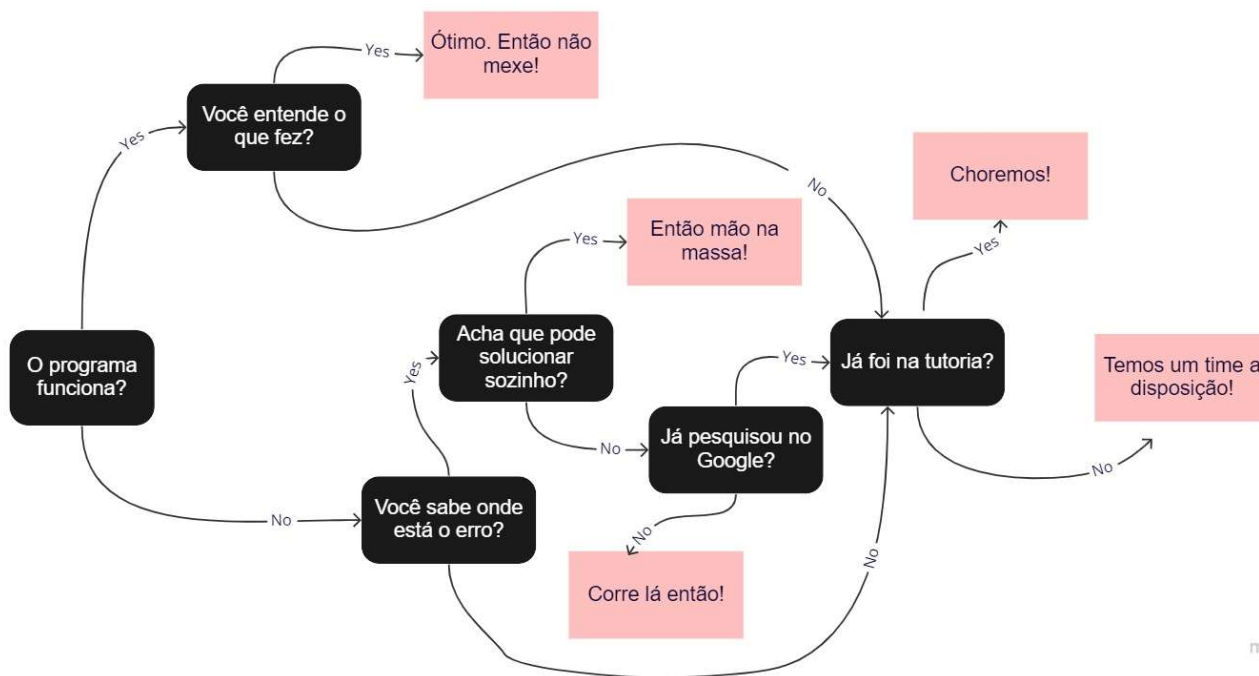
Questão 2

Correto

Atingiu 2,00 de 2,00

Filtro de Códigos

Para programar existem alguns requisitos que nossos códigos devem atender independente do problema. Para isso, algumas perguntas podem ser feitas durante a implementação de um programa, conforme demonstra o guia abaixo. Sua tarefa é implementar um programa que ajude os programadores a avaliarem seus códigos da mesma forma que o guia. A figura apresenta a ordem das perguntas que deverão ser feitas.



A Entrada consiste de:

- Após cada pergunta é lida uma String que pode ser do tipo 'SIM' ou 'NÃO'.

A Saída deve apresentar:

- Todas as perguntas de acordo com as respostas do usuário e por fim uma das cinco possíveis respostas finais.

Observações:

- Não é necessário validar se os valores de entrada são do tipo definido.

Descrição dos Exemplos:

- Os exemplo são auto explicativos.

For example:

Input	Result
SIM SIM	O programa funciona? Você entende o que fez? Ótimo. Então não mexe!
SIM NÃO SIM	O programa funciona? Você entende o que fez? Já foi na tutoria? Choremos!
SIM NÃO NÃO	O programa funciona? Você entende o que fez? Já foi na tutoria? Temos um time a disposição!



Answer: (penalty regime: 0, 0, 10, 20, ... %)

```
1 resp1 = input("O programa funciona?\n")
2 if resp1 == "SIM":
3     resp2 = input("Você entende o que fez?\n")
4     if resp2 == "SIM":
5         print("Ótimo. Então não mexe!\n")
6     else:# resp2 != SIM
7         resp6 = input("Já foi na tutoria?\n")
8         if resp6 == "SIM":
9             print("Choremos!\n")
10        else: #resp 6 != "SIM"
11            print("Temos um time a disposição!\n")
12
13 else:#resp1 != SIM
14     resp3 = input("Você sabe onde está o erro?\n")
15     if resp3 == "SIM":
16         resp4 = input("Acha que pode solucionar sozinho?\n")
17         if resp4 == "SIM":
18             print("Então mão na massa!\n")
19         else:#resp4 != "SIM"
20             resp5 = input("Já pesquisou no Google?\n")
21             if resp5 == "SIM":
22                 resp6 = input("Já foi na tutoria?\n")
23                 if resp6 == "SIM":
24                     print("Choremos!\n")
25                 else: #resp 6 != "SIM"
26                     print("Temos um time a disposição!\n")
27             else: #resp5 != "SIM"
28                 print("Corre lá então!\n")
29         else: #resp3 == "SIM"
30             resp6 = input("Já foi na tutoria?\n")
31             if resp6 == "SIM":
32                 print("Choremos!\n")
33             else: #resp 6 != "SIM"
34                 print("Temos um time a disposição!\n")
35
36
37
38
```

	Input	Expected	Got	
✓	SIM SIM	O programa funciona? Você entende o que fez? Ótimo. Então não mexe!	O programa funciona? Você entende o que fez? Ótimo. Então não mexe!	✓
✓	SIM NÃO SIM	O programa funciona? Você entende o que fez? Já foi na tutoria? Choremos!	O programa funciona? Você entende o que fez? Já foi na tutoria? Choremos!	✓
✓	SIM NÃO NÃO	O programa funciona? Você entende o que fez? Já foi na tutoria? Temos um time a disposição!	O programa funciona? Você entende o que fez? Já foi na tutoria? Temos um time a disposição!	✓
✓	NÃO NÃO SIM	O programa funciona? Você sabe onde está o erro? Já foi na tutoria? Choremos!	O programa funciona? Você sabe onde está o erro? Já foi na tutoria? Choremos!	✓
✓	NÃO NÃO NÃO	O programa funciona? Você sabe onde está o erro? Já foi na tutoria? Temos um time a disposição!	O programa funciona? Você sabe onde está o erro? Já foi na tutoria? Temos um time a disposição!	✓
✓	NÃO SIM SIM	O programa funciona? Você sabe onde está o erro? Acha que pode solucionar sozinho? Então mão na massa!	O programa funciona? Você sabe onde está o erro? Acha que pode solucionar sozinho? Então mão na massa!	✓



Passou em todos os teste! ✓

Correto

Notas para este envio: 2,00/2,00.



Questão 3

Correto

Atingiu 1,00 de 1,00

Carêncio está carente

Nem sempre é fácil encontrar o amor da sua vida. Para Carêncio, um jovem solteiro que está se sentindo sozinho, aparentemente o mar não está para peixe. Como ele não possui uma vida movimentada o suficiente para encontrar alguém facilmente, seu amigo lhe indicou um aplicativo de namoro, o revolucionário "LovePertin". O aplicativo mostra sua localização e a de pessoas em uma raio de 1 km que podem se interessar pelo seu perfil. Porém, Carêncio acredita que o amor deve ser algo fácil e não tem tanta paciência para lutar por uma paixão, então decidiu que, a depender da distância que a pessoa estiver, não valeria a pena investir em um encontro. Sendo D a distância entre ele e a pessoa, ele determinou os seguintes critérios:

- $D \leq 100$: "É o amor da minha vida!"
- $100 < D \leq 200$: "Talvez dê certo"
- $D > 200$: "Não vale a pena investir"

Como Carêncio tem feito sucesso no aplicativo, ele pediu sua ajuda na classificação de cada match que recebeu. Para isso implemente um programa que de acordo com a distância entre eles determina se vale ou não a pena investir no encontro respeitando os critérios estabelecidos por Carêncio.

A Entrada consiste de:

- 4 linhas, todas com variáveis do tipo inteiro, sendo as duas primeiras (x_1, y_1) a localização de Carêncio e as duas últimas (x_2, y_2) a localização de sua mais nova paquera.

A Saída deve apresentar:

- Uma das três frases ("É o amor da minha vida!", "Talvez dê certo", "Não vale a pena investir") de acordo com a distância.

Observações:

- Não é necessário validar se os valores de entrada são do tipo definido.
- Para solucionar a questão atente para o uso da fórmula da distância entre dois pontos
- Em python você pode utilizar a função `math.sqrt(x)` para retirar raiz quadrada

Descrição dos Exemplos:

- No primeiro exemplo, a distância entre Carêncio e sua paquera é de aproximadamente 145m, por ser um valor entre 100 e 200 a saída é a segunda frase.

For example:

Input	Result
10 5 15 150	Talvez dê certo
3 7 15 20	É o amor da minha vida!
0 0 100 15	Talvez dê certo

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```
1 def distancia(x1,y1,x2,y2):
2     D = ((x1-x2)**2 +(y1-y2)**2)**(0.5)
3     return D
4 #inicio do programa
5 x1 =int(input())
6 y1 =int(input())
7 x2 =int(input())
```



```

8 | y2 =int(input())
9 |
10 | d = distancia(x1,y1,x2,y2)
11 | if d <= 100:
12 |     print("É o amor da minha vida!")
13 | elif (100 < d) and (d <= 200):
14 |     print("Talvez dê certo")
15 | else:
16 |     print("Não vale a pena investir")

```

	Input	Expected	Got	
✓	10 5 15 150	Talvez dê certo	Talvez dê certo	✓
✓	3 7 15 20	É o amor da minha vida!	É o amor da minha vida!	✓
✓	0 0 100 15	Talvez dê certo	Talvez dê certo	✓
✓	0 7 10 0	É o amor da minha vida!	É o amor da minha vida!	✓
✓	200 8 1000 0	Não vale a pena investir	Não vale a pena investir	✓
✓	185 5 385 5	Talvez dê certo	Talvez dê certo	✓

Passou em todos os teste! ✓

Correto

Notas para este envio: 1,00/1,00.



Questão 4

Correto

Atingiu 3,00 de 3,00

Média Aritmética, Ponderada e Harmônica

Escreva um programa que leia 3 números inteiros positivos e efetue o cálculo das médias Aritmética (A), Ponderada (P) e Harmônica (H) dependendo da letra dada pelo usuário, mostre qual o tipo de média e qual o valor da média. No caso do usuário digitar qualquer outro caractere, apresente a mensagem 'Operacao inexistente'.

A Entrada consiste de:

- Linha contendo as três notas que são três números reais positivos.
- Linha contendo um caractere (para determinar qual a média), sendo (P) Ponderada, (H) Harmônica e (A) Aritmética
- Caso o caractere seja 'P', deve-se solicitar os três pesos de cada nota enviada, que são números positivos inteiros.

A Saída deve apresentar:

- Na primeira linha, o tipo de média que ele fez ("Harmonica", "Ponderada", "Aritmetica" ou "Operacao inexistente")
- Na segunda linha, caso tenha sido digito um caractere válido, o resultado da média com precisão de 2 casas decimais.

Observações:

- Não é necessário validar se os valores de entrada estão dentro dos intervalos e tipos definidos.

Descrição dos Exemplos:

- Os exemplos são autoexplicativos.

For example:

Input	Result
1 2 3 P 5 6 7	Ponderada 2.11
1 4 3 G	Operacao inexistente
5 3 8 H	Harmonica 4.56

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```
1 def medP(x1,x2,x3,a,b,c):
2     return (x1*a+x2*b+x3*c)/(a+b+c)
3 def medA(x1,x2,x3):
4     return medP(x1,x2,x3,1,1,1)
5 def medH(x1,x2,x3):
6     return 3/(1/x1 +1/x2 +1/x3)
7
8 x1,x2,x3 = input().split()
9 x1,x2,x3 = float(x1),float(x2),float(x3)
10 opcao = input()
11
12 if opcao == "A":
13     print("Aritmetica")
14     print(f"{medA(x1,x2,x3):.2f}")
15 elif opcao == "P":
16     a,b,c = input().split()
17     a,b,c = int(a),int(b),int(c)
18     print("Ponderada")
19     print(f"{medP(x1,x2,x3,a,b,c):.2f}")
20 elif opcao == "H":
21     print("Harmonica")
22     print(f"{medH(x1,x2,x3):.2f}")
23 else:
24     print("Operacao inexistente")
25
```



	Input	Expected	Got	
✓	1 2 3 P 5 6 7	Ponderada 2.11	Ponderada 2.11	✓
✓	1 4 3 G	Operacao inexistente	Operacao inexistente	✓
✓	5 3 8 H	Harmonica 4.56	Harmonica 4.56	✓
✓	5 6 7 A	Aritmetica 6.00	Aritmetica 6.00	✓
✓	10 15 20 C	Operacao inexistente	Operacao inexistente	✓
✓	80 38 54 P 4 7 2	Ponderada 53.38	Ponderada 53.38	✓

Passou em todos os teste! ✓

Para resolver esta questão implemente as condicionais e as equações que realizam os tipos de médias solicitados no enunciado.

Correto

Notas para este envio: 3,00/3,00.



Questão 5

Correto

Atingiu 3,00 de 3,00

Debugando estruturas de decisão

Inácio, dono de uma empresa de eletrodomésticos, estava preocupado com a quantidade de reclamações que seus produtos e serviços tinham na Internet. Por isso, resolveu criar um indicador para melhor acompanhar estas reclamações, o índice de insatisfação (ii) do cliente.

O índice de insatisfação, chamado de ii, é composto por vários outros índices, tais como, o índice de reclamação, o índice de indisponibilidade, além do número de cancelamentos, de acordo com a descrição a seguir.

Ele definiu que o índice de reclamações seria uma escala de 0 a 100. Se o cliente é atendido imediatamente após ligar ao call center zero é adicionado ao ii e 100 se esperam em média mais de 100 minutos para serem atendidos. Se 60% ou mais das reclamações são resolvidas na primeira ligação, o ii baixa 5 pontos. Caso contrário aumenta 15 pontos.

Sobre o ii, é feita a seguinte alteração:

a) se o número de cancelamentos dos serviços é maior ou igual a 10% do total dos clientes, o ii aumenta 80 pontos se o cancelamento foi por problemas nos serviços prestados ou diminui 30 pontos caso contrário.

b) se o número de cancelamentos dos serviços é menor do que 10% do total dos clientes, o ii aumenta 50 pontos se o cancelamento foi por problemas nos serviços prestados ou diminui 10 pontos caso contrário.

Depois disso, é computado o índice de indisponibilidade do serviço que varia de 0 a 100. Se os seus serviços ficaram fora do ar 10% ou mais do tempo em um mês, seu ii total será aumentado de 70 pontos, caso contrário, seu ii será rebaixado de 20 pontos.

Para isto, Inácio, que está aprendendo a [programação](#), escreveu o seguinte [programa](#) em [Python](#):

```
indiceReclamacao = int(input())
percentReclamResolPrim = int(input())
percentCliCancel = int(input())
indiceIndisponibilidade = int(input())
canceladoPorProblema = int(input())

if (percentReclamResolPrim > 60):
    indice = indiceReclamacao - 5
else:
    indice = indiceReclamacao + 15
print(f'{indice}')

if ( percentCliCancel <= 10):
    if (canceladoPorProblema==0):
        indice = indice + 80
    else:
        indice = indice - 30
else:
    if (canceladoPorProblema==0):
        indice = indice + 50
    else:
        indice = indice - 10
print(f'{index}');

if (indiceIndisponibilidade> 10):
    indice = indice - 70
else:
    indice = indice + 20
print(f'{indice}');
```

Corrija os erros do [programa](#) sabendo que:

A Entrada consiste de 5 valores inteiros, um por linha:

- o índice de reclamações;
- a porcentagem das reclamações resolvidas na primeira ligação;
- a porcentagem dos clientes que solicitaram o cancelamento;



- o índice de disponibilidade; e
- um número inteiro que pode ser 0 ou 1, conforme se segue:
 - 1 - cancelamento por problemas no serviço;
 - 0 - caso contrário.

A Saída deve apresenta três valores inteiros, um por linha:

- o índice de insatisfação da empresa levando-se em conta o índice de reclamações e a porcentagem de reclamações resolvidas na primeira ligação
- o ii final, depois de computado o índice de disponibilidade
- o valor da ii computando os cancelamentos e seus motivos.

Observações:

- Teste todos os intervalos possíveis para ver se o [programa](#) está certo, principalmente os casos que estão no limite dos intervalos.

Descrição dos Exemplos:

- Os casos de testes são autoexplicativos.

For example:

Input	Result
100	95
60	65
15	45
5	
0	
100	95
70	175
10	155
5	
1	
100	115
45	85
20	155
10	
0	

Answer: (penalty regime: 0, 0, 10, 20, ... %)

RESET ANSWER

```

1 indiceReclamacao = int(input())
2 percentReclamResolPrim = int(input())
3 percentCliCancel = int(input())
4 indiceIndisponibilidade = int(input())
5 canceladoPorProblema = int(input())
6
7 if (percentReclamResolPrim >= 60):
8     indice = indiceReclamacao - 5
9 else:
10     indice = indiceReclamacao + 15
11 print(f'{indice}')
12
13 if ( percentCliCancel >= 10):
14     if (canceladoPorProblema==1):
15         indice = indice + 80
16     else:
17         indice = indice - 30
18 else:
19     if (canceladoPorProblema==1):
20         indice = indice + 50
21     else:
22         indice = indice - 10
23 print(f'{indice}');
24
25 if (indiceIndisponibilidade >= 10):
26     indice = indice + 70

```



```

27 | else:
28 |     indice = indice - 20
29 | print(f'{indice}');

```

	Input	Expected	Got	
✓	100 60 15 5 0	95 65 45	95 65 45	✓
✓	100 70 10 5 1	95 175 155	95 175 155	✓
✓	100 45 20 10 0	115 85 155	115 85 155	✓
✓	100 80 30 10 1	95 175 245	95 175 245	✓
✓	50 60 20 30 0	45 15 85	45 15 85	✓
✓	70 70 20 15 1	65 145 215	65 145 215	✓

Passou em todos os teste! ✓

Para resolver esta questão copie o [código Python](#) no [IDE](#) e tente executar.

Ele vai dar um erro de [sintaxe](#). Leia o erro de [sintaxe](#) e corrija-o.

Faça isto até o [código](#) não ter mais erros de [sintaxe](#).

Não tendo mais erros de [sintaxe](#), o [código](#) será executado pelo [interpretador Python](#) até o fim e apresentará a saída.

Verifique se a saída está correta. Se a saída estiver errada, leia atentamente o [programa](#) procurando por erros semânticos. Para verificar o bom funcionamento do [programa](#), o aluno deve testar o [programa](#) com varias entradas diferentes, simulando todas as possibilidades que podem ocorrer. Em particular, verificar se o [programa](#) passa por todos os trechos de [códigos](#) que estão aninhados nas estruturas condicionais verificando se as condições para executar este [código](#) estão corretas.

Correto

Notas para este envio: 3,00/3,00.

