

Para o Desafio do Módulo 2, você deve criar um programa que informa uma série de opções para o usuário. De acordo com o que o usuário digitar, uma ação diferente será tomada.

Você deve criar um menu de opções, que vai de 1 até 7. O menu na tela deve ser assim:

```
Olá Fulano.  
Digite a opção desejada:  
1) Verificar triângulo  
2) Calcular equação do segundo grau  
3) Conferir data  
4) Verificar tamanho do texto  
5) Analisar CPF  
6) Contar caracteres  
7) Sair
```

Dependendo da opção informada, seu programa deve fazer o que se pede, conforme as descrições abaixo. Se o usuário digitar um valor que não esteja entre 1 e 7, imprima uma mensagem de erro e finalize o programa.

1) Verificar triângulo

O usuário deve informar os 3 lados de um triângulo (inteiros) e você deve imprimir na tela se o triângulo é isósceles, escaleno, equilátero ou se não é um triângulo. Para saber se 3 valores formam ou não um triângulo, devemos obedecer à Condição de Existência de um Triângulo:

a medida de qualquer um dos lados seja menor que a soma das medidas dos outros dois e maior que o valor absoluto da diferença entre essas medidas.

Ou seja:

$$\begin{array}{l} |b - c| < a < b + c \\ |a - c| < b < a + c \\ |a - b| < c < a + b \end{array}$$

Para calcular o valor absoluto (ou seja, o que está dentro dos `| |`, de um valor em Python, basta utilizar o comando `abs(número)`. Por exemplo: `abs(-4) = 4`

2) Calcular equação do segundo grau

O usuário deve informar os termos a, b e c de uma equação do segundo grau, e o resultado deve ser mostrado na tela. Para calcular uma equação do segundo grau, entretanto, devemos tomar alguns cuidados:

- (i) a não pode ser zero: se isto ocorrer, informe que não se trata de uma equação do segundo grau e não calcule;
- (ii) o delta calculado não pode ser negativo (parte dentro da raiz quadrada): se isto ocorrer, a equação não possui raízes reais. Informe ao usuário e não calcule
- (iii) o delta calculado é 0: se isto ocorrer, a equação possui apenas uma raiz real. Informe com uma mensagem e mostre a única raiz;
- (iv) se delta positivo, mostre as duas raízes.

3) Conferir data

O usuário digita um dia, um mês e um ano, e o programa informa se a data está correta ou não (conferindo os dias em relação aos meses. Não é necessário verificar se o ano é bissexto ou não).

4) Verificar tamanho do texto

O comando `len(texto)` do Python retorna a quantidade de caracteres de uma string. Por exemplo:

```
time = "Grêmio!"  
x = len(time)  
print(x)
```

Neste caso, x terá o valor 7. Teste para ver como funciona.

Sabendo disto, peça para o usuário digitar um texto qualquer. Caso ele digite um texto com menos de 5 caracteres, informe que o texto é muito pequeno. Caso o texto tenha entre 5 (inclusive) e 15 caracteres, informe que o texto é de tamanho médio. Caso o texto tenha entre 15 (inclusive) e 20 caracteres, informe que o texto é grande. Caso contrário, informe que o texto é inválido.

5) Analisar CPF

O comando `isdigit()` pode ser aplicado a uma determinada string, e ele informa se existem apenas números nela. Por exemplo:

```
texto = "1234a"  
y = texto.isdigit()
```

Neste caso, `y` receberá o valor `False`, pois não existem apenas números na string.

Sabendo disto, solicite que o usuário informe seu CPF apenas com números e informe a mensagem CPF inválido ou CPF válido. Não esqueça que um CPF correto deve 11 números.

6) Contar caracteres

O comando `count` conta a quantidade de vezes que um determinado texto aparece em uma string. Por exemplo:

```
texto = "Hoje é sexta-feira"  
vezes = texto.count("H")
```

Neste caso, a variável `vezes` terá valor 1, pois existe apenas um "H" em texto. Neste outro exemplo:

```
vezes = texto.count("sexta")
```

a variável `vezes` terá o valor 1, pois o texto "sexta" aparece 1 vez. Já neste exemplo:

```
vezes = texto.count("segunda")
```

a variável `vezes` terá o valor 0, pois o texto "segunda" não está presente na string texto.

Sabendo disto, peça para o usuário digitar um texto qualquer e informe ao usuário: a quantidade de vogais no texto, a quantidade de espaços e a quantidade de outros caracteres (consoantes ou caracteres especiais).

7) Sair

Apenas imprime a mensagem: "Obrigado por utilizar nosso sistema."

O que devo entregar?

- Você deve entregar um arquivo com extensão .py com o seu código.

Atenção!

- É muito importante: não deixe de entregar o que você fez, mesmo que incompleto!
- O que foi entregue precisa estar com código sem erros de sintaxe para que seja avaliado.
- Trabalhos copiados da internet ou de outros colegas levarão nota zero automaticamente (todos os envolvidos).
- Caso tenhas dúvidas contate o professor nas webconferências, mas não deixe para a última hora!

Boa sorte!