

Funções

1) Faça um programa em Python para calcular as quatro operações, uma calculadora. Crie uma função que retorne a soma de dois números passados por parâmetro, outra para subtração, multiplicação e divisão respectivamente. Crie uma função chamada "interface_calculadora", onde o programa deverá pedir dois números ao usuário e a operação desejada, se a operação for soma deve ser chamado a função soma, e assim para as outras opções.

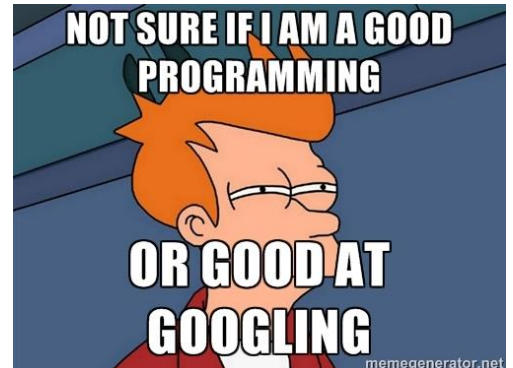
```
def soma(a,b):  
    return a + b  
  
def subtracao(a,b):  
    return a - b  
  
def multiplicacao(a,b):  
    return a * b  
  
def divisao(a,b):  
    return a / b  
  
def interface_calculadora():  
  
    print "Menu:"  
    print "1 Soma"  
    print "2 Subtracao"  
    print "3 Multiplicacao"  
    print "4 Divisao"  
    opc = input()  
  
    inputA = input("Valor:")  
    inputB = input("Valor:")  
  
    if(opc == 1):  
        print "O valor da soma: ", soma(inputA, inputB)  
  
    if(opc == 2):  
        print "O valor da subtracao: ", subtracao(inputA, inputB)  
  
    if(opc == 3):  
        print "O valor da multiplicacao: ", multiplicacao(inputA, inputB)  
  
    if(opc == 4):  
        print "O valor da divisao: ", divisao(inputA, inputB)  
  
interface_calculadora()
```



2) Escreva uma função em Python que receba uma lista de n números inteiros e retorne, para o usuário, o comprimento da maior sequência crescente. Ex: na lista a = [6, 11, 4, 3, 5, 8, 10, 9, 6], o comprimento da maior sequência crescente é 4 (pois 3, 5, 8 e 10 é a maior sequência crescente). Já nesta lista b = [11, 9, 6, 4, 3], o comprimento da maior sequência é 1.

```
def crescentSegmentLength(l):
    seq = 1
    aux = 0
    for i in range(1, len(l)):
        if l[i-1] < l[i]:
            seq = seq + 1
        else:
            seq = 1
        if seq > aux:
            aux = seq

    return aux
```



3) Faça um algoritmo que solicite ao usuário números e os armazene em um vetor de 20 posições. Crie uma função que recebe o vetor preenchido e substitua todas as ocorrências de valores negativos por zero, as ocorrências de valores menores do que 10 por 1 e as demais ocorrências por 2.

```
def altera(vet, tam):
    for i in range(tam):
        if vet[i] < 0:
            vet[i] = 0
        else:
            if vet[i] < 10:
                vet[i] = 1
            else:
                vet[i] = 2
    return vet

tam = 39

v = [0]*tam
for i in range(tam):
    v[i] = input('Digite um valor: ')
altera(v, tam)
print v
```



4) Crie uma função que retorne o valor da expressão: $\frac{2}{3} + \frac{3}{5} + \frac{4}{7} + \frac{5}{9} + \dots + \frac{n}{m}$, para um valor de n e m definido pelo usuário. No programa, verifique se o valor de n definido pelo usuário é positivo (antes de chamar a função) e, caso não seja, solicite outro valor até ser fornecido um valor positivo.

```
def sequ(n):
    aux1 = 2
    aux2 = 3.0
    soma = 0
```

```

while aux1 <= num:
    print aux1, aux2
    soma = soma + aux1/aux2
    aux1 = aux1 + 1
    aux2 = aux2 + 2

return soma

num = input('Digite um valor: ')
while num < 0:
    num = input('Digite um valor positivo: ')

res = sequ(num)
print res

```

5) Escreva uma função que recebe uma lista B com n elementos (sem repetições) e um índice k (onde $0 \leq k < n$) e tem como saída o índice do elemento mínimo entre $B[k]$, $B[k+1]$, ..., $B[n-1]$.
Ex: B[6,2,9,4,6,11,2,3] e k=3 → índice 7

```

def mini(l, k):
    smallerIndex = k
    for i in range(k+1, len(l)):
        if(l[i] < l[smallerIndex]):
            smallerIndex = i
    return smallerIndex

```