

Lista de Exercícios - Funções, Funções Recursivas e de Ordem Superior
Prof. Igor Machado Coelho - 2025.1

Parte I

1) Faça um programa em 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 a operação for soma deve ser chamado a função soma, e assim para as outras opções.

2) Escreva uma função 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.

3) Faça um programa 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.

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 repetidamente até ser fornecido um valor positivo.

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 6 (que equivale ao elemento 2)

6) Crie uma função para calcular e imprimir os n primeiros números de Tribonacci. A série de Tribonacci consiste em: 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504,... Para calcula-la o primeiro elemento vale 1, o segundo elemento vale 1, o terceiro elemento vale 2, e daí por diante.

Assim, o i-ésimo elemento vale o (i-1)-ésimo elemento somado ao (i-2)-ésimo elemento somado ao (i-3)-ésimo elemento. Exemplo, $13=7+4+2$. Observe que n deve ser positivo.

Parte II = Funções Recursivas e de Ordem Superior

7) Implemente uma função recursiva "eh_palindromo" que recebe um string e retorna verdadeiro, caso seja palindromo, ou falso, caso contrário.

Exemplos:

eh_palindromo("ana") retorna True

eh_palindromo("radar") retorna True

eh_palindromo("python") retorna False

Dica I: Considere que um string vazio é palindromo.

Dica II: Considere parametros auxiliares na função recursiva, como o inicio e fim do string: eh_palindromo(s, inicio, fim), tal que inicio=0 e fim=len(s)-1

Dica III: Teste a função para palavras de tamanho par e de tamanho ímpar.

8) Faça uma função recursiva soma_bignum(s1, s2) que efetua a soma de dois números gigantes s1 e s2 representados como strings de decimais como *little endian*, ou seja, começando dos números menores ao invés dos maiores (o contrário do que estamos acostumados).

Exemplos:

soma_bignum("321", "490") retorna "3601"

pois $123 + 940 = 1063$, em little endian: "3601"

soma_bignum("9", "1") retorna "01"

pois $9 + 1 = 10$, em little endian: "01"

soma_bignum("5", "7"), retorna "21"

soma_bignum("321", "49") retorna "712"

Dica I: considere que os números nunca são vazios

Dica II: considere parametros auxiliares na função recursiva, como um índice i (inicialmente zero), e um parâmetro "vaium" (inicialmente zero)

Dica III: comece por strings que tenham um mesmo tamanho, depois expanda a solução para strings com tamanhos distintos

9) Considere a função de ordem superior **mapear**, definida no material de aula.

9.a) Explique o que acontece na seguinte expressão:

```
dados = mapear(int, input("digite uma lista de entrada:").split())
```

9.b) Escreva uma função media(a,b,c), que calcula a média entre 3 elementos

9.c) Escreva uma função de ordem superior chamada suavizar, que recebe uma função de entrada e uma lista, aplicando a função na lista a cada três elementos consecutivos (exceto o primeiro e o último)

Exemplos:

```
print(suavizar(media, [1, 3, 7, 9, 4]))
```

```
# [1, 3.666, 6.333, 6.666, 4]
```

```
# Pois,  $(1+3+7)/3 = 3.666$ ,  $(3+7+9)/3=6.333$ ,  $(7+9+4)/3=6.666$ 
```

```
print(suavizar(media, [10, 20, 30, 40, 50]))
```

```
# [10, 20.0, 30.0, 40.0, 50]
```