

Programação de Computadores I - Lista de Exercícios 2¹
Primeiro Semestre de 2023

1 Funções

1. Escreva uma função que dado um inteiro positivo n , devolve $n!$.
2. Escreva uma função que dados números inteiros positivos n e k , devolve o valor de $\binom{n}{k}$.
3. Escreva uma função que dado um número inteiro n , devolve uma lista com os divisores de n em ordem crescente.
4. Escreva uma função que dado um número inteiro n , devolve um Booleano dizendo se n é perfeito (veja lista 1 para a definição de perfeito).
5. Escreva uma função que dado um número inteiro n , devolve um Booleano dizendo se n é aritmético (veja lista 1 para a definição de aritmético).
6. Escreva uma função que dado um inteiro n , devolve a lista dos primeiros números perfeitos menores ou iguais a n .
7. Escreva uma função que dada uma lista de strings que representam números, devolve a lista de inteiros correspondentes.
8. Faça um programa que calcula a soma dos primeiros n números primos. Use diversas funções para resolver este problema.

2 Listas

1. Faça uma função que dada uma lista e um valor, insere o valor no final da lista apenas se ele não estiver presente na lista.
2. Faça uma função que dada uma lista, devolve uma nova lista sem repetição porém preservando a ordem original. Isto é, você deve manter a primeira cópia de cada elemento da lista original.

¹Lista elaborada a partir do material dos professores Rafael C. S. Schouery e André Backes

3. Faça uma função que dada uma lista, devolve uma nova lista com os elementos na ordem invertida.
4. Repita o exercício anterior para fazer a alteração *in-place*.
5. Um número k não-negativo pode ser escrito como $a + b + c$ onde a , b , c são números inteiros não-negativos de diversas formas. Por exemplo, $k = 6$ pode ser escrito como $6 = 3 + 3 + 0$ ou $6 = 2 + 2 + 2$. Escreva uma função que dado um número k , devolve uma lista com as possibilidades de escrita de k .
6. Repita o exercício anterior considerando apenas $a \leq b \leq c$. Ou seja, $6 = 3 + 3 + 0$ não será considerada pois é redundante com $6 = 0 + 3 + 3$.

3 Strings

1. Faça uma função que dada uma string, devolve uma lista, com repetição e em ordem, dos caracteres presentes nessa string.
2. Faça uma função que dada uma string, devolve uma lista, sem repetição mas em ordem, dos caracteres presentes nessa string.
3. Faça uma função que dada uma string e um inteiro k , devolve uma lista de todas as substrings de tamanho k presentes nessa string.
4. Faça uma função que dada uma string, devolve um Booleano representando se a string é um palíndromo.
5. Faça uma função que dada uma string, devolve um Booleano representando se a string é uma string quadrada. Uma string quadrada é a concatenação de uma string com ela mesma.
6. Faça uma função que procura um padrão em uma string, devolvendo o índice da primeira ocorrência do padrão (e dá um `ValueError` se o padrão não aparece na string). Nesse exercício um padrão é uma string que pode conter “*” para representar um caractere qualquer. Assim, “manda*o” está tanto na string “cassaram o mandato do prefeito”, quanto em “expediu um mandado de prisão”, mas não está em “pedaços de um caderno manchado de vinho”.

4 Matrizes

1. Faça uma função que dada uma matriz, devolve um Booleano representando se a matriz é simétrica.
2. Faça uma função que dada uma matriz, devolve um Booleano representando se a matriz é a identidade.
3. Faça uma função que dada uma matriz, devolve um Booleano representando se a matriz é triangular superior.
4. Faça uma função que dada uma matriz 3×3 , devolve o seu determinante.
5. Desafio: Faça uma função que dada uma matriz não-singular, inverte a matriz.

5 Arquivos

1. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas linhas esse arquivo possui.
2. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas letras são vogais e quantas são consoantes.
3. Faça um programa que receba do usuário um arquivo texto e um caractere. Mostre na tela quantas vezes aquele caractere ocorre dentro do arquivo.
4. Faça um programa que receba do usuário um arquivo texto e mostre na tela quantas vezes cada letra do alfabeto aparece dentro do arquivo.
5. Faça um programa que receba do usuário um arquivo texto. Crie outro arquivo texto contendo o texto do arquivo de entrada, mas com as vogais substituídas por “*”