

Teste o seu código. Entregue todas as funções em um arquivo chamado `lab1.py`. Critérios de avaliação:

Principalmente (50%) o código funciona, os parâmetros de entrada e os valores de retorno são corretos, a função faz o que foi pedido.

Adicionalmente (25%) o código é legível, bem organizado, o mais simples possível.

Adicionalmente (25%) o código é eficaz – não faz operações desnecessárias.

1. (2 pontos) Crie a função chamada `concatena` que receba duas strings e dois números inteiros `m` e `n`. A função deve retornar a concatenação da primeira string sem os `m` primeiros caracteres, com a segunda sem os últimos `n` caracteres.
2. (2 ponto) Implemente o método chamado `sublista` com três parâmetros de entrada: uma lista de números reais e dois números reais `m` e `n`. O método deve retornar uma sublista formada por todos os números da lista que são maiores que `m` e menores que `n`.
3. (2 pontos) Escreva uma função chamada `fun` cujo parâmetro de entrada é uma string `palavra` e uma lista de strings. A função junta a string `palavra` com as strings da lista em uma frase, separando as strings com espaço. O valor de retorno é a frase nova.
4. O número de Euler e pode ser aproximado pela soma da seguinte série:

$$e \approx \sum_{k=0}^n \frac{1}{k!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

- (a) (1 ponto) Escreva a função chamada `numeroEuler` que calcule o valor de e por meio da série acima até o n -ésimo termo do somatório, onde o número n é passado como um parâmetro. Obs.: Pode usar a função `factorial` do módulo `math` que retorna o fatorial de um número.
- (b) (1 pontos) O módulo `math` possui uma aproximação para o valor de e (`math.e`). Faça a função chamada `precisaoEuler` que descubra quantos termos da série acima devem ser calculados para que o erro absoluto entre `math.e` e o valor de e da série seja inferior a um `erro` passado como parâmetro. Dica: Pode usar a função `fabs` do módulo `math` que retorna o valor absoluto de um número.
- (c) (2 pontos) Crie a função `main` com a mesma funcionalidade como a função `precisaoEuler` acima, mas que peça ao usuário informar a precisão desejada (o `erro` máximo a ser tolerado). Não esqueça de chamar a função `main`, ou seja, incluir o seguinte comando no arquivo:

```
if __name__ == "__main__":
    main()
```