String Triangular

Mário Leite

...

O tratamento e computação de *strings* é a coisa mais difícil para um computador. Na verdade, a computação eletrônica, como todos sabem, trabalha com a "Base Binária" (0 e 1); assim, todas as vezes que mandamos o computador executar alguma tarefa e a entrada (*input*) é uma *string* ele (a CPU) trabalha muito para converter em *bits* numéricos, seja uma *string* pequena ou um texto. Quando um programa recebe uma *string* ele precisa converter essa sequência de caracteres em uma representação binária; e esse processo de conversão e manipulação pode consumir recursos significativos, especialmente se a string for longa ou se houver muitas operações de manipulação de desse tipo sendo realizadas. Algumas operações ficam muito "difíceis para o computador:

Alocação de Memória: As *strings* podem variar de tamanho, e o computador precisa alocar dinamicamente a memória para armazená-las; e gerenciar essa memória de forma eficiente é crucial para o desempenho do sistema.

Operações de Manipulação: Muitas operações comuns com *strings*, como concatenação, busca, substituição e divisão, podem ser computacionalmente intensivas. Por exemplo, concatenar duas strings pode envolver a criação de uma nova *string* e a cópia dos caracteres das strings originais para a nova string.

Codificação e Decodificação: A *strings* podem ser representadas em diferentes codificações (por exemplo, UTF-8, UTF-16). O computador precisa ser capaz de lidar com essas diferentes codificações, o que adiciona uma camada extra de complexidade.

Processamento de Texto: Analisar e processar texto, como em tarefas de processamento de linguagem natural (PLN), pode ser extremamente exigente em termos de poder computacional. Tarefas como análise sintática, reconhecimento de entidades e tradução automática envolvem algoritmos complexos que manipulam strings extensivamente.

Entretando, na prática diária os programas estão repletos de *inputs* e *outputs* de *strings*, já que o programa é ditado pelo usuário, que às vezes exige que as informações sejam legíveis e bem explicadas nas formas mais diversas. O programa **"StringTriangular"** é um exemplo-desafio para os programadores iniciantes; ele pede como entrada uma palavra e a reescreve na forma triangular: diretamente (começando com a primeira e finalizando com a palavra inteira), ou inversamente (começando com a palavra inteira e finalizando com a sua primeira letra. Embora seja bem simples, é um bom exercício para os iniciantes em programação.

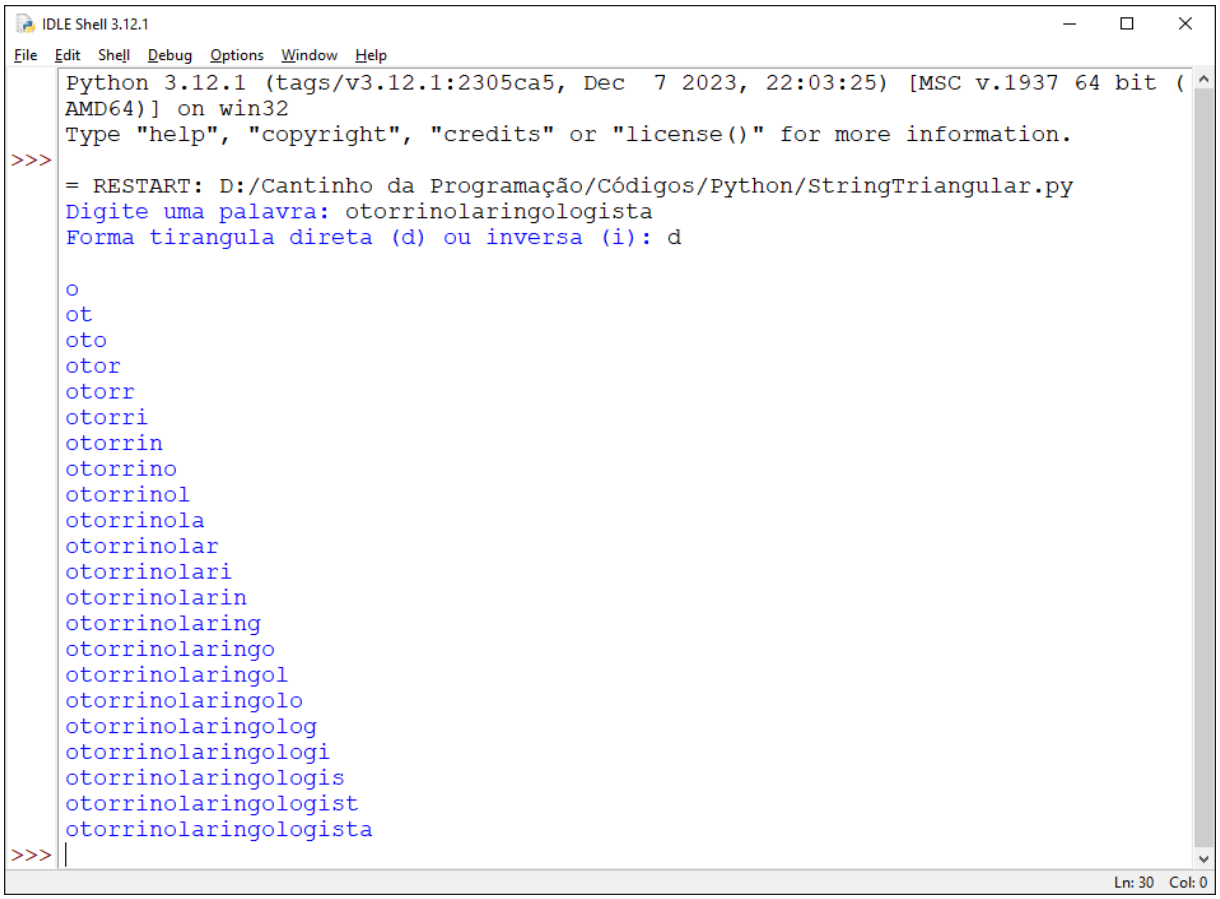

Figura 1a - string direta

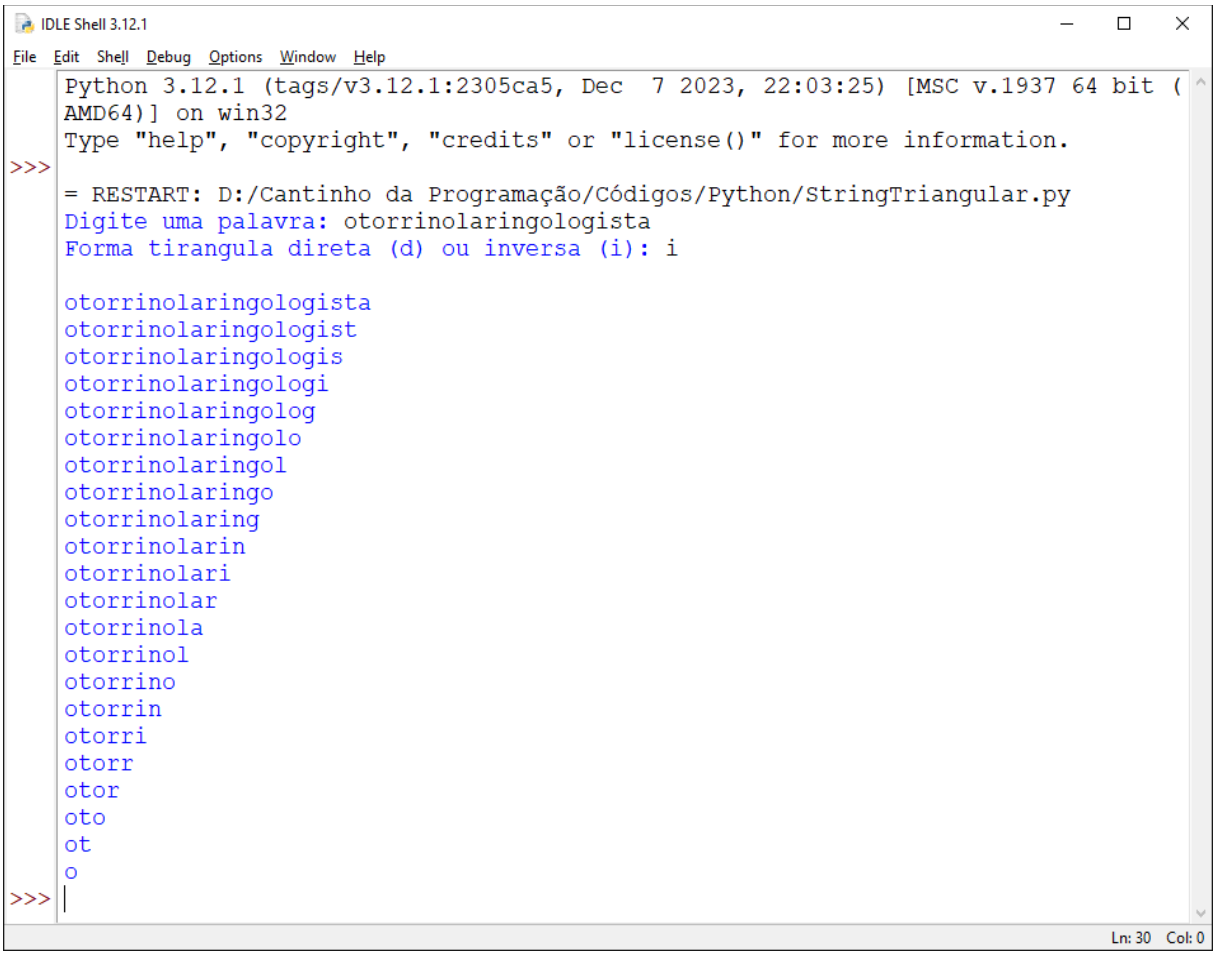

Figura 1a - string inversa

```
'''
StringTriangular.py
-------------------------------------------------------------------------
Lê uma palavra e a exibe na forma triangular "direta" ou "inversa".
-------------------------------------------------------------------------
'''

def MostrarDireta():
    #Exibe a palavra na forma triangular "direta"
    print()
    for j in range(tamStr):
        print(""*(tamStr -1) + strOrig[:j+1])
#-------------------------------------------------------------------------
def MostrarInversa():
    #Exibe a palavra na forma triangular "inversa".
    print()
    for j in range((tamStr-1),-1,-1):
        print(""*(tamStr -j -1) + strOrig[:j+1])
#-------------------------------------------------------------------------
def main():
    global tamStr, strOrig
    strOrig = input("Digite uma palavra: ")
    tamStr = len(strOrig)
    resp = input("Forma triangular direta (d) ou inversa (i): ")
    resp = resp.upper()
    if(resp=="D"):
        MostrarDireta()
    elif(resp=="I"):
        MostrarInversa()
    else:
        print()
        print("Opção inválida!")
#=========================================================================
#Programa principal
if(__name__ == "__main__"):
    main()
#Fim do programa "StringTriangular" -------------------------------
```