



UNISANTOS

Universidade Católica de Santos

Centro de Ciências Exatas, Arquitetura e Engenharia

Professor:	Ciro Cirne Trindade
Disciplina:	Introdução à Computação-II
Cursos:	Ciência da Computação/Sistemas de Informação

Lista de Exercícios – Strings

1. Escreva um programa que receba uma string do teclado e informe quantas letras maiúsculas ela possui.
2. Escreva um programa que receba uma linha de texto, conte as vogais e apresente o histograma da frequência de cada uma delas. Considere que a linha de texto não contém acentuação. Por exemplo:
Linha de texto: “As eleicoes 2018 acontecerao em Outubro”

Histograma:
a: *** (3)
e: ***** (6)
i: * (1)
o: ***** (5)
u: ** (2)
3. Escreva um programa que dada uma string, remove todos os espaços em branco dessa string e depois a imprime no vídeo.
4. Escreva um programa que receba uma string do teclado e informe se ela é palíndromo ou não. Uma string é palíndromo quando pode ser lida tanto de trás para frente quanto de frente para trás e possui exatamente a mesma sequência de caracteres. Por exemplo: “ASA”, “SUBI NO ONIBUS”. Desconsidere os espaços.
5. Escreva um programa que dadas duas strings (A e B), construa e exiba uma terceira string (C) formada pelos caracteres de A e B intercalados. Por exemplo: A = “Quarta” e B = “Segunda”, a resposta deve ser C = “QSueagrutnada”.
6. Escreva um programa que receba uma string de 0s e 1s, interprete essa string como um número binário e informe o valor desse número na base decimal.
7. Escreva um programa para ler 10 nomes em um vetor de strings *A*. Depois formar um vetor de strings *B* de tal forma que os elementos (nomes) estejam em ordem inversa a do vetor *A*.
8. Escreva um programa para ler 8 nomes em cada um dos vetores de strings *A* e *B*. Construir um vetor de strings *C*, que será formada por um elemento do vetor *A* e por um elemento do vetor *B*, intercaladamente.
9. Escreva um programa que dada um inteiro *n* e um caractere *ch*, solicite que o usuário informe *n* palavras que começam com a letra *ch*. Se o usuário informar alguma palavra que não comece com a letra *ch*, o programa deve exibir uma mensagem informando que a palavra não começa com a letra *ch*. No final o programa deve exibir as *n* palavras digitadas pelo usuário que começam com a letra *ch*.



UNISANTOS

Universidade Católica de Santos

10. Sentença Dançante¹. Uma sentença é chamada de dançante se sua primeira letra for maiúscula e cada letra subsequente for o oposto da letra anterior. Espaços devem ser ignorados ao determinar o case (minúsculo/maiúsculo) de uma letra. Por exemplo, "A b Cd" é uma sentença dançante porque a primeira letra ('A') é maiúscula, a próxima letra ('b') é minúscula, a próxima letra ('C') é maiúscula, e a próxima letra ('d') é minúscula.

Entrada

A entrada contém vários casos de teste. Cada caso de teste é composto por uma linha que contém uma sentença, que é uma string que contém entre 1 e 50 caracteres ('A'-'Z', 'a'-'z' ou espaço ' '), inclusive, ou no mínimo uma letra ('A'-'Z', 'a'-'z').

Saída

Transforme a sentença de entrada em uma sentença dançante (conforme o exemplo abaixo) trocando as letras para minúscula ou maiúscula onde for necessário. Todos os espaços da sentença original deverão ser preservados, ou seja, " sentence " deverá ser convertido para " SeNtEnCe ".

Exemplo

Entrada	Saída
This is a dancing sentence	ThIs Is A dAnCiNg SeNtEnCe
This is a dancing sentence	ThIs Is A dAnCiNg SeNtEnCe
aaaaaaaaaaaa	AaAaAaAaAaA
z	Z

11. Avance as Letras². É dado na entrada uma string A e outra B. Em uma operação você pode escolher uma letra da primeira string e avançar esta letra. Avançar uma letra significa transformá-la na próxima letra do alfabeto, veja que a próxima letra depois de z vem a letra a novamente!

Por exemplo, podemos transformar a string ab em bd em no mínimo 3 operações: $ab \rightarrow bb \rightarrow bc \rightarrow bd$. Podemos aplicar operações nas letras em qualquer ordem, outra possibilidade seria: $ab \rightarrow ac \rightarrow bc \rightarrow bd$.

Dadas as duas strings, calcule o mínimo número de operações necessárias para transformar a primeira na segunda.

Entrada

Na primeira linha terá um inteiro T ($T \leq 100$) indicando o número de casos de teste. Para cada caso, na única linha teremos as duas strings A ($1 \leq |A| \leq 100$ ou $1 \leq |A| \leq 10^4$ - sendo que |A| significa o tamanho da string A) e B ($|B| = |A|$ ou $|B| = |A|$) separadas por um espaço. Ambas as strings são compostas por letras do alfabeto minúsculas apenas e são do mesmo tamanho.

Saída

Para cada caso imprima o número mínimo de operações.

1 Extraído de URI Online Judge. Disponível em:
<https://www.urionlinejudge.com.br/judge/pt/problems/view/1234>

2 Extraído de URI Online Judge. Disponível em:
<https://www.urionlinejudge.com.br/judge/pt/problems/view/1607>



UNISANTOS

Universidade Católica de Santos

Exemplo

Entrada	Saída
3	3
ab bd	0
abc abc	173
abcdefghijklhiz aaaaaaaaaa	

13. Huaauhahhuahau³. Em chats, é muito comum entre jovens e adolescentes utilizar sequências de letras, que parecem muitas vezes aleatórias, para representar risadas. Alguns exemplos comuns são:

huaauhahhuahau
hehehehe
ahahahaha
jaisjjkasjksjjskjakijs
huehuehue

Cláudia é uma jovem programadora que ficou intrigada pela sonoridade das “risadas digitais”. Algumas delas ela nem mesmo consegue pronunciar! Mas ela percebeu que algumas delas parecem transmitir melhor o sentimento da risada que outras. A primeira coisa que ela percebeu é que as consoantes não interferem no quanto as risadas digitais influenciam na transmissão do sentimento. A segunda coisa que ela percebeu é que as risadas digitais mais engraçadas são aquelas em que as sequências de vogais são iguais quando lidas na ordem natural (da esquerda para a direita) ou na ordem inversa (da direita para a esquerda), ignorando as consoantes. Por exemplo, “hahaha” e “huaauhahhuahau” estão entre as risadas mais engraçadas, enquanto “riaajkdhhhhjak” e “huehuehue” não estão entre as mais engraçadas.

Cláudia está muito atarefada com a análise estatística das risadas digitais e pediu sua ajuda para escrever um programa que determine, para uma risada digital, se ela é das mais engraçadas ou não.

Entrada

A entrada é composta por uma linha, contendo uma sequência de no máximo 50 caracteres, formada apenas por letras minúsculas sem acentuação. As vogais são as letras ‘a’, ‘e’, ‘i’, ‘o’, ‘u’. A sequência contém pelo menos uma vogal.

Saída

Seu programa deve produzir uma linha contendo um caractere, “S” caso a risada seja das mais engraçadas, ou “N” caso contrário.

Exemplos

Entrada	Saída
hahaha	S

Entrada	Saída
riaajkdhhhhjak	N

3 Maratona de Programação da SBC 2016. Disponível em:
<https://www.urionlinejudge.com.br/judge/pt/problems/view/2242>



UNISANTOS

Universidade Católica de Santos

Entrada	Saída
a	S

Entrada	Saída
huaauhahhuahau	S