

(***) 1. (Adaptado da OBI 2008)

A maioria das universidades brasileiras usa o vestibular para selecionar seus alunos. O vestibular consiste de uma ou mais provas sobre as matérias do Ensino Médio, visando avaliar os conhecimentos dos candidatos. Um formato popular de prova de vestibular é a prova objetiva. Neste formato de prova, cada candidato deve escolher uma das cinco alternativas apresentadas pela questão como sendo a correta. Durante a correção dos cartões, cada questão onde a alternativa escolhida pelo candidato é a mesma do gabarito, ele ganha um ponto. Alguns dos vestibulares mais concorridos do Brasil são disputados por dezenas de milhares de candidatos, e, por isso, geralmente usa-se uma folha de leitura óptica e um programa de computador para corrigir as provas de todos os candidatos e gerar a lista com suas pontuações. Você trabalha no comitê responsável pelo vestibular em uma faculdade e deve escrever um programa que, dado o gabarito e as respostas de um dos candidatos, determina o número de acertos daquele candidato.

Entrada:

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão. O primeiro valor da entrada é um inteiro N ($1 \leq N \leq 80$), que indica o número de questões da prova. A segunda entrada é uma cadeia de N caracteres, que indica o gabarito da prova. A terceira entrada contém outra cadeia de N caracteres, que indica as opções marcadas pelo candidato. Ambas as cadeias contêm apenas os caracteres A, B, C, D e E (sempre em letra maiúscula).

Saída:

Seu programa deve imprimir na saída padrão uma única linha contendo um único inteiro, indicando o número de acertos do candidato.

Exemplos:

Entrada	Saída
7 AEDBCCE ADDCCBE	4
5 ABCDE ABCDE	5
10 ABCDEABCDE BCDEABCDEA	0

(**) 2. Escreva um programa que gera e mostra um vetor contendo N elementos aleatórios. O usuário deve informar uma posição do vetor. O programa deve então remover o elemento na posição informada, “puxando” todos os elementos das posições posteriores uma posição para “trás”. Por exemplo, se o vetor contém os números 1, 3, 6, 2 e 10 e o usuário digita 2, o programa deve remover o elemento na posição número 2 do vetor – o 6 (lembre-se que a contagem inicia em 0) – ficando com os números 1, 3, 2 e 10. Após a remoção, o programa deve mostrar como o vetor ficou e pedir uma nova entrada ao usuário. O programa termina quando o vetor ficar vazio ou se o usuário selecionar uma posição que estaria fora do vetor. Note que, após uma ou mais remoções, o vetor terá posições inutilizadas, o conteúdo destas posições é indeterminado.

(***) 3. Cebolinha é um personagem de história em quadrinhos que quando fala, troca o R pelo L. Escreva uma função `stling`, que recebe uma string `s` e um buffer (um vetor capaz de armazenar outra string com o mesmo tamanho de `s`) e armazene no buffer uma versão de `s` com todos os R's trocados por L's, exceto quando o R é a última letra de uma palavra. Dois R's seguidos devem ser substituídos por um único L. Lembre-se de manter maiúsculas e minúsculas como no texto original.



(***) 4. (adaptado de <https://neps.academy/br/exercise/118>)

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
jaissjksjksjjskjakij
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 “riajkdhhhhjak” 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 uma função que determine, para uma risada digital, se ela é das mais engraçadas ou não.

Entrada:

A função deve receber como entrada uma string 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:

A função deve retornar 1 caso a risada seja das mais engraçadas, ou 0 caso contrário.

Exemplos:

Entrada	Saída
hahaha	1
riajkdhhihhjak	0
a	1
huaauhahhuahau	1