

Minutos  
Restantes:  
22302

Usuário:  
Gustavo Ribeiro  
da Fonseca  
Santos

Notas:  
Q1: ?  
Q2: ?  
Q3: ?  
Q4: ?  
Q5: ?  
Total: 0

Exercícios de Strings

Prova Aberta Até: 26/10/2019 05:00:00

Número Máximo de Tentativas: 6

Atenuação da Nota por Tentativa: 0%

Instruções para a prova:

Questão 1: Vetores - String - Palavras Justificadas

Construa um programa capaz de justificar um conjunto de palavras.

O usuário deverá inicialmente definir o número de palavras a serem justificadas. Em seguida o programa deverá ler a quantidade de palavras definida e no fim do processo apresentar as palavras na mesma ordem de leitura alinhadas a direita. Para o alinhamento deverá ser utilizado o caractere \* (asterisco). Assuma que as palavras não contêm espaços.

Obs.: Para programas em C, recomenda-se não utilizar o comando getline.

Para programas em Python, os vetores são implementados como listas cujos elementos são todos homogêneos.

Entrada:

- Número de palavras,
- Palavras.

Saída:

- Palavras digitadas alinhadas a direita utilizando-se como referência a maior palavra. A ordem das palavras deverá ser a mesma da entrada.

Exemplo de entrada:

```
4
PROVA
ALGORITMOS
UFLA
DCC
```

Exemplo de saída:

```
*****PROVA
*****ALGORITMOS
*****UFLA
*****DCC
```

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Questão 2: Vetores - String - Contar vogais

Faça um programa que dada uma string, somente com letras minúsculas, sem espaços, conta quantas vogais apareceram.

Exemplo de entrada:  
paralelogramo

Exemplo de saída:  
6

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Questão 3: Strings - Metades iguais

Faça um programa que leia diversas strings e as separe em duas categorias:

1. Strings que se divididas ao meio possuem as mesmas letras e mesma frequência em cada lado, exemplo é **xyzxyz** onde após a divisão o primeiro lado **xyz** e o segundo **xyz** possuem as mesmas letras que se repetem na mesma frequência (uma única vez). Caso a palavra possua número ímpar de letras então deve-se desconsiderar a letra do meio, por exemplo **aab**cd**rad**bca****, quando essa string é dividida obtemos os dois lados **aabcd** e **adbca** e a letra do meio à desconsiderar **r**.

2. Palavras que se divididas ao meio não tenham as propriedades da categoria 1.

Entradas:

- Número inteiro indicando a quantidade de strings que serão lidas.
- Todas as strings.

Saídas:

- Strings que respeitam a categoria 1 em ordem crescente por número de letras seguida pelas strings da categoria 2 também em ordem crescente (cada string em uma linha).

Exemplo de Entrada:

```
5
abcabc
xyyx
zistzip
abdba
abc
```

Exemplo de Saída:

```
xyyx
abdba
abcabc
abc
zistzip
```

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Questão 4: Strings - Letra não repetida

Faça um programa que dada um string de entrada, informa qual o primeiro caractere que não se repete.

**Obs:** Considere que sempre existirá pelo menos uma letra que não se repete.

Entradas:

- Uma string.

Saídas:

- Primeira letra que não se repete na string.

Exemplo de Entrada:

```
aabcade
```

Exemplo de Saída:

```
b
```

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Questão 5: Vetores e Strings - Dick Vigarista e Muttley - Fazendo hash de um vetor de palavras

Dick Vigarista enviou uma mensagem para Muttley, mas por medo da mensagem ser alterada no meio do caminho por inimigos (incluindo o terrível pombo), eles vão utilizar um **cálculo de hash** para verificar se a mensagem chegou corretamente, sem qualquer modificação. Um hash é um algoritmo que mapeia dados de comprimento variável para dados de comprimento fixo, sendo muito utilizado para fazer verificações (checksums ou hashes). No caso, ele enviará um **vetor de palavras**, informando antes quantas palavras farão parte da mensagem. Após o envio da última palavra, será enviado um único número para indicar o hash da mensagem.

O código a seguir faz o hash de uma palavra qualquer, utilizando uma chave baseada em um número primo e um valor inteiro M:

```
const int UMPRIMO = 13;
const int M = 10;
long hash = 0;

for (unsigned j = 0; j < palavra.size(); j++) {
    hash = (UMPRIMO * hash + palavra[j]) % M;
}
```

Nesse caso, perceba que o hash vai ser o resto da divisão por 10 a partir de um cálculo com os caracteres da palavra. Ou seja, cada palavra terá um hash entre 0 e 9. O hash da mensagem inteira será o resultado da soma dos hashes individuais. Por exemplo, caso a mensagem enviada seja *"muttley faca alguma coisa"*, cada palavra terá o seguinte hash:

```
muttley 6
faca 1
alguma 7
coisa 3
```

E nesse caso o hash da mensagem inteira será 17 (soma dos hashes individuais).

Para auxiliá-los, seu objetivo é **implementar um algoritmo que recebe um vetor de palavras, armazenando-o, e imprime o hash individual de cada palavra, bem como o hash da mensagem inteira, utilizando como base o código fornecido acima.**

Entradas:

- Tamanho do vetor de palavras
- Vetor de palavras separadas por espaço

Saídas:

- Hash individual de cada palavra
- Hash de toda a mensagem

Exemplo de Entrada:

```
4
muttley faca alguma coisa
```

Exemplo de Saída:

```
6 1 7 3 17
```

Exemplo de Entrada:

```
5
pegue aquele maldito pombo agora
```

Exemplo de Saída:

```
8 7 4 5 6 30
```

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta