

Prova-03

Prof. Msc. Elias Batista Ferreira
Prof. Dr. Gustavo Teodoro Laureano
Profa. Dra. Luciana Berretta
Prof. Dr. Thierson Rosa Couto

Sumário

1	Leitura de conjuntos (+++)	2
2	Matriz bissimétrica (+++)	4
3	Remove vogais (+++)	5

1 Leitura de conjuntos (+++)



(+++)

Em uma atividade de Teoria de Conjuntos do curso de Ciência da Computação, o aluno deve fazer operações básicas entre diversos conjuntos. Para tornar as atividades mais fáceis, um aluno de Ciência da Computação teve a ideia de automatizar as operações de Teoria de Conjuntos em um programa de computador usando vetores de números inteiros. Por definição, um conjunto válido não possui elementos repetidos. No entanto, é comum cometer o erro de informar elementos repetidos para compor um conjunto. O processo de leitura precisa de uma validação de cada novo elemento buscando garantir somente a inserção de elementos únicos. Você deve construir uma função que garanta a leitura de somente conjuntos válidos formados por números inteiros. Essa função deve ler a quantidade de números do conjunto e uma sequência de números inteiros. À medida que um número é lido, a função deve verificar se esse número já compõe o conjunto e montar um vetor formado apenas por elementos únicos. Essa função deve seguir o protótipo:

```
1  /**
2   * Essa função lê um vetor de n números inteiros que formam um conjunto válido.
3   * A função lê uma sequência de números até encontrar uma quantidade de
4   * n números únicos.
5   * @param vect vetor de números inteiros
6   * @param n quantidade de números únicos a serem lidos
7   * @return o vetor vect é preenchido com números não repetidos
8   */
9  void le_conjunto( int * vect, int n );
```

Faça um programa que leia uma sequência de números inteiros de modo a formar um conjunto válido. O conjunto deve ter no mínimo 1 e no máximo 100 elementos.

Entrada

Durante a entrada de dados, o programa deve ler um tamanho n válido para o conjunto. Caso o tamanho informado seja inválido, o programa encerrar. Uma vez definido o tamanho do conjunto, o programa deve ler uma sequência indeterminada de números inteiros até atingir a quantidade de n números únicos que preencham o vetor.

Saída

O programa deve apresentar na tela uma linha contendo os elementos do conjunto, entre parênteses e separados por vírgulas e sem espaços.

Exemplo

Entrada	Saída
3 1 2 3	(1, 2, 3)

Entrada	Saída
4 9 5 0 0 0 7	(9, 5, 0, 7)

Entrada												Saída											
5												(5, 9, 0, 7, 2)											
5 9 5 5 9 5 5 0 0 7 7 2																							

2 Matriz bissimétrica (+++)



(+++)

Uma matriz bissimétrica é uma matriz quadrada que é simétrica em ambas diagonais. Faça um programa que leia uma matriz quadrada (com números inteiros) de tamanho $n \times n$, onde $1 \leq n \leq 10$, e verifique se a matriz é bissimétrica ou não.

Entrada

O programa deve ler um número inteiro n válido que representa a dimensão da matriz e, em seguida, os $n * n$ elementos da matriz.

Saída

Caso o número n esteja fora do intervalo, o programa imprime a mensagem "dimensao invalida" e encerra. Caso contrário, o programa imprime a mensagem "bissimetrica" ou "nao bissimetrica".

Observações

Exemplo

Entrada	Saída
3 1 2 3 2 5 2 3 2 1	bissimetrica
Entrada	Saída
3 1 2 3 9 5 2 3 2 1	nao bissimetrica
Entrada	Saída
21	dimensao invalida

3 Remove vogais (+++)



(+++)

Escreva a função `remove_vogais` que remove todas as vogais de um texto e calcula a quantidade de vogais removidas. A função `remove_vogais` recebe como parâmetro uma *string* `str`, e um vetor de inteiros com 5 posições, correspondendo às vogais 'a', 'e', 'i', 'o' e 'u'. A função deve modificar a *string* passada como parâmetro e atualizar o vetor de ocorrências de vogais. Considere o tamanho máximo de 256 caracteres para a *string* de entrada.

```
1
2 /**
3  * Função que remove vogais e calcula a quantidade de vogais removidas
4  * @param str string de entrada
5  * @param vogais vetor de 5 posições que contabiliza a quantidade
6  *           de vogais removidas
7  * @return A função atualiza os vetores str e vogais.
8  */
9 void remove_vogais( char * str, int * vogais );
```

Entrada

Seu programa deve ler uma *string*.

Saída

Uma linha contendo a *string* modificada e outras 5 linhas contendo a quantidade das vogais 'a', 'e', 'i', 'o' e 'u' que foram removidas.

Exemplo

Entrada	Saída
Fulano de Tal da Silva	Fln d Tl d Slv a: 4 e: 1 i: 1 o: 1 u: 1

Entrada	Saída
Ciencia DA COMPUTACAO	Cnc D CMP TC a: 4 e: 1 i: 2 o: 2 u: 1