

Parte-02-Matrizes

Prof. Dr. Gustavo Teodoro Laureano
Prof. Dr. Thierson Rosa Couto

Sumário

1	Determinante 2x2 (+)	2
2	Diagonal Principal (+)	3
3	Diagonal Secundária	4
4	Quadrado de matriz 2x2	6
5	Ampulheta	7
6	Desenha bordas (++)	9
7	Desenha losango (+++)	10
8	Matriz de permutação (+++)	12
9	Operações matriciais (++)	14
10	Cadê o Wally?	15
11	Elementos únicos da Matriz (+++)	17
12	Frequencia do Maior e Menor	18
13	Matriz bissimétrica (+++)	19
14	Matriz Ordenada	20
15	Separador de palavras (+++)	21
16	Turismo (+++)	23

1 Determinante 2x2 (+)



(+)

Escreva um programa em C que leia uma matrix de dimensão 2×2 e calcule o seu determinante. Seja a matriz $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ o determinante de A é $|A| = ad - bc$.

Entrada

Uma sequência de 4 números reais.

Saída

A saída deve conter 1 linha, contendo o determinante da matriz com 2 casas decimais.

Exemplo

Entrada	Saída
0 5 11 2	-55

2 Diagonal Principal (+)



(+)

Faça um programa que receba uma matriz quadrada e imprima os elementos de sua diagonal principal.

Entrada

A entrada contém apenas em caso de teste. A primeira linha ha um inteiro N , $1 < N \leq 1000$, representando a dimensão das matrizes. A seguir haverá N linhas com N inteiros em cada linha separados por um espaço em branco cada, representando os elementos da matriz A .

Saída

A saída consiste de N linhas com um inteiro em cada linha. Na primeira linha haverá o elemento da diagonal principal da primeira linha da matriz, na segunda linha haverá o elemento da diagonal principal da segunda linha da matriz e assim por diante até o n -ésimo elemento da diagonal principal da n -ésima linha da matriz. Após o último elemento impresso quebre uma linha.

Exemplo

Entrada	Saída
3 1 2 3 4 5 6 7 8 9	1 5 9

Entrada	Saída
2 34 23 56 98	34 98

3 Diagonal Secundária



(+)

Crie um programa que receba uma matriz quadrada e imprima os elementos de sua diagonal secundária.

Entrada

Na primeira linha há um inteiro n , $1 < n \leq 1000$, representando a ordem da matriz quadrada. A seguir haverá N linhas com N inteiros em cada linha separados por um espaço em branco cada, representando os elementos da matriz quadrada.

Saída

A saída consiste de n linhas com um inteiro em cada linha. Na primeira linha haverá o elemento da diagonal secundária da primeira linha da matriz, na segunda linha haverá o elemento da diagonal secundária da segunda linha da matriz e assim por diante até o n -ésimo elemento da diagonal secundária da n -ésima linha da matriz. Após o último elemento impresso quebre uma linha.

Exemplo

Entrada	Saída
3 1 2 3 4 5 6 7 8 9	3 5 7

Entrada	Saída
5 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1	0 0 1 0 0

Entrada	Saída
20 59 18 45 66 59 34 96 26 30 24 41 0 63 94 32 63 0 50 55 76 49 50 66 45	66 26 0 63 76

Entrada		Saída	
1		100	
100			

4 Quadrado de matriz 2x2



(++)

Faça um programa que leia uma matriz 2x2 e imprima o seu quadrado. O quadrado de uma matriz **A** é dado pela operação: $A^2 = AA$.

Entrada

O programa deve ler 4 números reais.

Saída

O programa deve imprimir a matriz resultante com precisão de 3 casas decimais.

Exemplo

Entrada	Saída
1 0 0 1	1.000 0.000 0.000 1.000

Entrada	Saída
5 8 7 1	81.000 48.000 42.000 57.000

5 Ampulheta



(++)

O objetivo desse exercício é identificar um conjunto de elementos cuja somatória seja o maior valor entre todos os conjuntos definidos por um padrão na forma de ampulheta, em uma matriz de inteiros, quadrada, e de ordem 6. Cada elemento da matriz está no intervalo $[-9, 9]$. Uma “ampulheta” é formada pelos valores

que estão posicionados de acordo com a seguinte configuração:

$$\begin{bmatrix} a & b & c \\ & d & \\ e & f & g \end{bmatrix}.$$

O valor de uma ampulheta é a soma de todos os valores presentes nas respectivas posições. Seu programa deve retornar o maior valor entre todos os valores de ampulheta possíveis na matriz.

Entrada

Uma matriz quadrada de ordem 6.

Saída

Um único inteiro que corresponde à maior soma de todos os valores de ampulheta. Após imprimir o valor quebre a linha.

Exemplo

Entrada	Saída
<pre>1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>	7

Entrada	Saída
<pre>1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 2 4 4 0 0 0 0 2 0 0 0 0 1 2 4 0</pre>	19

Entrada	Saída
<pre>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 1</pre>	0

Entrada	Saída
5 5 -1 -4 1 6 -2 -3 3 -2 3 -7 9 -4 -5 3 8 -9 -2 -2 7 3 5 -9 -8 -7 6 -3 -2 8 1 9 7 0 3 -2	30

Entrada	Saída
-9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9	-63

Entrada	Saída
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	63

6 Desenha bordas (++)



(++)

Faça um programa que gere uma matriz de zeros, de tamanho definido pelo usuário, de no máximo 100×100 , com uma borda de largura k de valor x .

Entrada

O programa deve ler quatro números inteiros, os dois primeiros relacionados à largura e altura da matriz, o terceiro a largura da borda e o por último o valor da borda.

Saída

O programa deve apresentar a matriz como uma imagem PGM, ou seja, seguindo a sequência:

```
P2
largura altura
255
<dados da matriz>
```

Os dados da matriz devem ser impressos sempre com um espaço à direita e seguido de quebra de linha ao final de cada linha da matriz.

Observações

Para testar seu código, você pode redirecionar a saída padrão do seu programa para um arquivo com extensão ".pgm", usando o comando `./programa > img.pgm`.

Exemplo

Entrada	Saída
5 5 1 2	P2 5 5 255 2 2 2 2 2 2 0 0 0 2 2 0 0 0 2 2 0 0 0 2 2 2 2 2 2

7 Desenha losango (+++)



(+++)

Faça um programa que gere uma matriz quadrada de caracteres, de tamanho definido pelo usuário, de no máximo 99×99 , com um losango desenhado em seu interior, equivalente ao maior quadrado rotacionado 45 graus. O programa deve ler a dimensão da matrix, o caractere a ser usado para representar a borda do losango e o caractere a ser usado como interior do losango. O restante da matriz deve ser preenchido com o caractere espaço.

Entrada

O programa deve um número inteiro n , ímpar, que representa a dimensão da matriz. Em seguida, deve-se ler dois caracteres, o primeiro representando o caractere de borda e o segundo o caractere de fundo. Para realizar a leitura use a expressão "%d %c %c".

Saída

Caso o valor informado para a dimensão da matriz seja inválido, o programa deve imprimir a mensagem "Dimensao invalida!" e encerrar. Caso contrário, o programa deve imprimir a matriz $n \times n$ gerada, com cada elemento separado por um espaço à direita, (usando "%c ") e cada linha separada por uma quebra de linha '\n'.

Exemplo

Entrada 1

```
3 o y
```

Saída 1

```
  o
o y o
  o
```

Entrada 2

```
11 x o
```

Saída 2

```

      x
    x o x
  x o o o x
x o o o o o x
x o o o o o o x
x o o o o o o o x
x o o o o o o o x
  x o o o o x
    x o x
      x

```

8 Matriz de permutação (+++)



(+++)

Dizemos que uma matriz inteira $A_{n \times n}$ é uma matriz de permutação se em cada linha e em cada coluna houver $n - 1$ elementos nulos e um único elemento igual a 1.

A matriz A abaixo é de permutação:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz B abaixo não é de permutação.

$$B = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Dada uma matriz inteira $A_{n \times n}$, verificar se A é de permutação.

Você deve implementar uma função que recebe a matriz e retorne 0 ou 1, sendo que o 0 (zero) indica que a matriz não é de permutação:

```
1 /**
2  * Função que verifica se a matriz é de permutação
3  * @param matriz Indica a matriz a ser verificada
4  * @param n indica a dimensão da matriz
5  * @param *soma parâmetro de saída, que armazenará a soma de todos os 'n' elementos
   da matriz.
6  * @return int
7  */
8 int ehPermutacao( int matriz[500][500], int n, int *soma );
```

Entrada

Na primeira linha há um inteiro n , $1 < n \leq 500$, representando a ordem da matriz quadrada. A seguir haverá n linhas com n inteiros em cada linha separados por um espaço em branco cada, representando os elementos da matriz quadrada.

Saída

Deverá imprimir 3 (três) linhas:

- A dimensão da matriz (n)
- A mensagem "PERMUTACAO" ou "NAO EH PERMUTACAO", que representa se esta **é** ou **não** uma matriz de permutação
- Soma de todos os elementos da matriz

Exemplo

Entrada	Saída
4 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1	4 PERMUTACAO 4

Entrada	Saída
3 2 -1 0 -1 2 0 0 0 1	3 NAO EH PERMUTACAO 3

9 Operações matriciais (++)



(++)

Faça um programa que, dada uma matriz quadrada A de dimensões N x N realize a seguinte operação:

$$\text{tr}(\mathbf{A}) \cdot \mathbf{A} + \mathbf{A}^T \quad (1)$$

Onde:

- \mathbf{A} é a matriz de entrada;
- \mathbf{A}^T é a matriz transposta de \mathbf{A} ;
- tr é o traço da matriz, ou seja, a soma dos elementos na diagonal principal;

Entrada

A entrada contém apenas um caso de teste. A primeira linha há um inteiro N, $1 < N \leq 1000$, representando a dimensão das matrizes. A seguir haverá N linhas com N inteiros em cada linha separados por um espaço em branco cada, representando os elementos da matriz A.

Saída

A saída consiste de N linhas com N inteiros em cada linha separados por um espaço em branco cada, representando o resultado da Equação 1.

Exemplo

Entrada	Saída
3 1 2 3 4 5 6 7 8 9	16 34 52 62 80 98 108 126 144

Entrada	Saída
2 34 23 56 98	4522 3092 7415 13034

10 Cadê o Wally?



(+++)

Wally costuma morar em um ambiente representado por uma matriz bidimensional de números inteiros de ordem $n \times m$ (n linhas e m colunas). Ele é único no ambiente e é representado na matriz por quatro números, distribuídos da seguinte forma:

	4	
0	1111	0
	8	

O número 1111 representa a camisa listrada em vermelho e branco, e o número 4 representa seu gorro das mesmas cores. Os números zero e oito representam suas extremidades. A matriz representa um ambiente bidimensional circular:

- Para o índice $i=0$, a célula à esquerda está no índice $m-1$.
- Para o índice $i=m-1$, a célula à direita está no índice 0.
- Para o índice $j=0$, a célula superior está no índice $n-1$.
- Para o índice $j=n-1$, a célula inferior está no índice 0.

Crie um programa que permita imprimir os índices i, j (da matriz) onde está a camisa do Wally. Caso o Wally não estiver na matriz mostre a seguinte mensagem: “WALLY NAO ESTA NA MATRIZ”. Na seguinte matriz de 7 linhas e 6 colunas, a camisa do Wally está nos índices $i=3, j=0$.

0	1111	0	1	0	0
0	0	0	0	1111	0
4	0	1	3	45	53
1111	0	89	211	87	0
8	4	56	4	55	98
0	222	0	11	0	5
0	8	23	8	66	3

Entrada

Dois valores inteiros, n e m , seguidos dos elementos inteiros da matriz de ordem $n \times m$, com $n \geq 3$ e $m \geq 3$.

Saída

Se o Wally estiver na matriz: o índice i, j da localização do Wally. Caso contrário “WALLY NAO ESTA NA MATRIZ” (sem acentos)

Exemplo

Entrada	Saída
3 7 4 7 7 7 7 7 7 1111 0 7 7 7 7 0 8 7 7 7 7 7 7	1 0

Entrada	Saída
5 4 5 5 8 5 5 5 5 5 5 5 5 5 5 5 4 5 5 0 1111 0	4 2

Entrada	Saída
3 3 1 4 1 0 1111 0 1 8 1	1 1

Entrada	Saída
3 3 0 0 0 0 0 0 0 0 0	WALLY NAO ESTA NA MATRIZ

11 Elementos únicos da Matriz (+++)



(+++)

Dada uma matriz $nl \times nc$, encontre todos os elementos que não se repetem. Considere a dimensão máxima da matriz igual a 10×10 .

Entrada

O programa deve ler a quantidade de linhas (nl) e colunas (nc) da matriz e, em seguida, os $nl * nc$ elementos da matriz.

Saída

Caso os números nl ou nc estejam fora do intervalo, o programa imprime a mensagem "dimensao invalida" e encerra. Caso não exista nenhum elemento único, o programa deve imprimir a mensagem "sem elementos unicos". Caso contrário, o programa imprime uma linha contendo os elementos que não se repetem na matriz separados por vírgula e obedecendo a sequência em que aparecem na matriz (da esquerda para direita e de cima para baixo).

Exemplo

Entrada	Saída
3 3 1 9 3 2 5 4 3 2 1	9, 5, 4

Entrada	Saída
3 2 2 3 1 2 3 1	sem elementos unicos

Entrada	Saída
0 8	dimensao invalida

12 Frequencia do Maior e Menor



(+++)

Dada uma matriz A de dimensões N x M contendo apenas números inteiros positivos, verifique o maior e o menor valor da matriz e conte quantas vezes estes valores aparecem.

Entrada

A entrada contém apenas em caso de teste. A primeira linha há dois números inteiros N e M, tais que $1 < N, M \leq 1000$, representando as dimensões da matriz A. A seguir haverá N linhas com M inteiros em cada linha separados por um espaço em branco cada, representando os elementos da matriz A. Cada elemento de A é um número inteiro tal que $0 \leq a_{ij} \leq 1000$;

Saída

A saída consiste de duas linhas: a primeira deve conter o menor valor da matriz A, um espaço, a porcentagem correspondente à frequência dele na matriz, com duas casas após a vírgula e em seguida o símbolo de porcentagem. A segunda linha é semelhante, contendo o maior valor de A e a porcentagem correspondente à sua frequência na matriz. Após a impressão do último valor, quebre uma linha.

Exemplo

Entrada	Saída
2 3 2 5 9 2 6 2	2 50.00% 9 16.67%

Entrada	Saída
3 3 1 2 3 2 0 4 1 0 0	0 33.33% 4 11.11%

13 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

14 Matriz Ordenada



(+++)

Faça um programa que, dada uma matriz A de dimensões N x N, ordene de forma crescente as colunas da matriz.

Entrada

A entrada contém apenas um caso de teste. A primeira linha ha um inteiro N, $1 < N \leq 1000$, representando a dimensão das matrizes. A seguir haverão N linhas com N inteiros em cada linha separados por um espaço em branco cada, representando os elementos da matriz A.

Saída

A saída consiste de N linhas com N inteiros em cada linha separados por um espaço em branco cada, representando a matriz A após o processo de ordenação de suas colunas. Após a última linha da matriz quebre uma linha.

Exemplo

Entrada	Saída
3 1 2 11 5 10 6 7 8 9	1 2 6 5 8 9 7 10 11

Entrada	Saída
2 34 23 56 8	34 8 56 23

15 Separador de palavras (+++)



(+++)

Uma das etapas mais frequentes de um algoritmo de processamento de texto é a separação do texto em palavras. Essa tarefa pode ser uma tarefa difícil uma vez que a variabilidade com que os textos são escritos é muito grande. Por exemplo, o texto "Nossa!!! Que chuva.", possui somente 3 palavras mas pode acontecer com diversas configurações de pontuação. O objetivo desse exercício é que você desenvolva a primeira etapa de processamento de texto que é a separação de um texto, no formato de *string*, em palavras dado um conjunto de caracteres que são considerados como separadores. As palavras serão armazenadas em uma matriz de caracteres, de modo que cada linha seja uma *string*. Como restrições do problema, considera que cada texto tem no máximo 200 palavras e cada palavra no máximo 64 caracteres. Neste problema também não há a presença de caracteres acentuados, no entanto, os caracteres de pontuação são livres para ocorrer.

Para dar mais flexibilidade à solução desse problema, você deverá implementar uma função que receba a *string* original, uma matriz de caracteres e uma *string* contendo a lista de caracteres separadores. Considere as macros `MAX_WORDS` e `MAX_WORD_LEN` as definições dos limites máximos para a declaração da memória do programa. A função deve seguir o seguinte protótipo:

```
1
2 #define MAX_WORDS 200
3 #define MAX_WORD_LEN 64+1
4
5 /**
6  * @brief Função de separação de palavras de acordo com a uma lista de separadores.
7  *      Exemplo de chamada da função:
8  *      str_split("Ola mundo! 1,23", m, " , .!?");
9  *
10 * O resultado é a separação das strings "Ola", "mundo", "1", "23", cada uma
11 * ocupando uma linha da matriz m, com base nos caracteres de pontuação fornecidos.
12 *
13 * @param str ponteiro para o início da string original
14 * @param m matriz de caracteres, sendo cada linha uma palavra da string original
15 * @param sep string com a lista de caracteres separadores
16 * @return int quantidade de palavras detectadas
17 */
18 int str_split(char * str, char m[][MAX_WORD_LEN], char * sep);
```

Entrada

Seu programa deve ler duas *strings*, a primeira o texto a ser processado e a segunda, a lista de caracteres de separação.

Saída

O programa deve apresentar um conjunto de linhas, cada uma contendo uma palavra do texto original, precedida pela sua quantidade de caracteres entre parênteses. Ao final, o programa deve apresentar a quantidade de palavras que possui exatamente a mesma quantidade de caracteres que a maior palavra encontrada.

Observações

Lembre-se que, para fazer a leitura de espaços, você deve especificar qual o caractere terminador de *string* no `scanf`, exemplo: `scanf("%[^\n]", str);`. Tente decompor o problema em problemas menores e implemente funções para cada sub-problema.

Exemplo

Entrada	Saída
Fulando de Tal da Silva. , .;:?!	(7) Fulando (2) de (3) Tal (2) da (5) Silva 1

Entrada	Saída
Nossa!!! Que chuva forte. Voce tem capa de chuva? , .;:?!	(5) Nossa (3) Que (5) chuva (5) forte (4) Voce (3) tem (4) capa (2) de (5) chuva 4

16 Turismo (+++)



(+++)

Os acessos e distâncias entre 6 cidades são listadas pela Tabela 1. Cada célula da tabela mostra a distância, em quilômetros, entre a cidade de cada linha com as cidades de cada coluna. O caracter '-' indica que não há acesso entre as cidades, partindo da cidade da linha correspondente.

Tabela 1: Tabela de distâncias e acessos entre cidades.

	Cárceres	Bugres	Cuiabá	Várzea	Tangará	Lacerda
Cárceres	0	63	210	190	-	190
Bugres	63	0	160	150	95	10
Cuiabá	210	160	0	10	1	10
Várzea	190	150	10	0	10	20
Tangará	10	95	7	21	0	80
Lacerda	190	2	-	41	80	0

Tendo conhecimento dessa tabela, uma agencia de turismo gostaria de ter um programa que, dada uma rota, verifique se a rota é válida e que calcule e apresente a distância da rota fornecida.

As cidades Cárceres, Burgres, Várzea, Tangará e Lacerda são representadas pelos números 0, 1, 2, 3, 4, 5 respectivamente. Desse modo, uma rota pode ser representada por um vetor de inteiros que indica o traslado entre as cidades listadas.

Por exemplo, o vetor {1, 2, 3} indica que a rota válida que inicia pela cidade de Bugres, passa pela cidade de Cuiabá e termina em Várzea, totalizando 170 km. Uma rota é inválida se a sequência do vetor atinge um elemento da matriz com o caracter '-'.

Entrada

O programa deve ler um número inteiro N , correspondente ao tamanho da rota, sendo $0 < N \leq 100$, e um vetor de inteiros com N elementos.

Saída

O programa deve apresentar a distância total da rota percorrida ou a mensagem "rota invalida!" caso a rota seja inválida.

Exemplo

Entrada	Saída
3 1 2 3	170
Entrada	Saída
3 0 4 1	rota invalida!