

Lista 04 - Matrizes e Alocacao Dinamica

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

Sumário

1	Determinante 2x2 (+)	2
2	Diagonal Secundária	3
3	Frações Equivalentes	5
4	Ler e imprimir	6
5	Tradutor do Papai Noel	7
6	Ampulheta	9
7	Desenha losango (+++)	11
8	Matriz de permutação (+++)	13
9	Frequencia do Maior e Menor	15
10	Imprime matriz em espiral (+++)	16
11	Soma e Subtração de Polinômios	18
12	Separador de palavras (+++)	20
13	Turismo (+++)	22

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 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		

3 Frações Equivalentes



(+)

Tia Zuleika está ensinando seus alunos sobre frações equivalentes. Ela vai gerar uma lista de frações e vai pedir aos seus alunos para indicarem quais pares de frações são equivalentes entre si, percorrendo a lista da esquerda para direita. Ela pretende gerar uma lista grande, para que todos os alunos da turma tenham oportunidade de responder. Tia Zuleika está muito ocupada e sabendo que você é “fera” em programação, está pedindo sua ajuda para que faça um programa capaz de ler sequências de frações e para cada sequência indicar quais pares são equivalentes entre si. O programa deve comparar a primeira fração com todas as outras seguintes, depois deve comparar a segunda com todas as outras seguintes e assim por diante, de modo que a posição na lista da primeira fração de um par de frações seja sempre menor que a posição da segunda fração do par.

Entrada

A primeira linha da entrada contém o número $n \leq 30$ de casos de teste. Para cada caso de teste há duas linhas na entrada. A primeira linha, contém o tamanho $m (m \leq 50)$ da sequência de frações. A segunda linha contém a sequência de m frações separadas entre si por um espaço. Cada fração tem o seguinte formato: x/y .

Saída

Para cada caso de teste a saída é formada por uma frase no formato “Caso de Teste t ”, onde t corresponde ao número de um caso de teste. Em seguida, o programa deve imprimir tantas linhas no formato “ x/y equivalente a w/u ” quantos forem os pares de frações equivalentes na sequência. Se não houver pares equivalentes o programa deve emitir uma linha com a mensagem: “Nao ha fracoes equivalentes na sequencia”.

Exemplo

Entrada
2
5
1/2 1/3 1/4 1/5 1/6
5
1/2 2/4 3/5 10/20 3/6
Saída
Caso de teste 1
Nao ha fracoes equivalentes na sequencia
Caso de teste 2
1/2 equivalente a 2/4
1/2 equivalente a 10/20
1/2 equivalente a 3/6
2/4 equivalente a 10/20
2/4 equivalente a 3/6
10/20 equivalente a 3/6

4 Ler e imprimir



(+)

Escreva um programa em C que armazene elementos inseridos pelo usuário em uma matriz A de dimensão máxima 10x10, e em seguida os imprima na forma matricial. O programa deve ler dois números inteiros válidos referentes às quantidades de linhas e colunas da matriz.

Entrada

Uma sequência de números inteiros com dois números válidos, m -linhas e n -colunas, para a definição da dimensão da matriz. Em seguida o programa deve ler $m \times n$ números inteiros.

Saída

A saída deve conter m linhas, cada linha iniciando com o texto "linha i :", onde i é o número da linha, e uma sequência de n elementos, separados por ',', correspondendo aos elementos (i, j) , onde $j = 0, 1, \dots, n - 1$.

Exemplo

Entrada	Saída
0	linha 1: 1,2
5	linha 2: 3,4
11	linha 3: 5,6
2	linha 4: 7,8
1 2 3 4 5 6 7 8 9 10	linha 5: 9,10

5 Tradutor do Papai Noel



(++)¹

Nicolau já está bastante cansado e sua memória não é mais a mesma. Você, como navegador, deverá auxiliar o Papai Noel a gritar a frase “Feliz Natal” no idioma correto de cada país de que o trenó está sobrevoando.

Como você é um elfo muito esperto, você já criou um pequeno app no seu celular (sim, elfos tem celular) que irá lhe informar a frase no idioma correto dado o nome do país. Como o trenó é moderno (foi atualizado no ano 2000) ele exibe no painel de navegação o nome do país atual.

Os dados inseridos no seu app foram:

brasil	Feliz Natal!
alemanha	Froehliche Weihnachten!
austria	Frohe Weihnacht!
coreia	Chuk Sung Tan!
espanha	Feliz Navidad!
greCIA	Kala Christougena!
estados-unidos	Merry Christmas!
inglaterra	Merry Christmas!
australia	Merry Christmas!
portugal	Feliz Natal!
suecia	God Jul!
turquia	Mutlu Noeller
argentina	Feliz Navidad!
chile	Feliz Navidad!
mexico	Feliz Navidad!
antardida	Merry Christmas!
canada	Merry Christmas!
irlanda	Nollaig Shona Dhuit!
belgica	Zalig Kerstfeest!
italia	Buon Natale!
libia	Buon Natale!
siria	Milad Mubarak!
marrocos	Milad Mubarak!
japao	Merii Kurisumasu!

Para não correr o risco de informar o nome errado você decidiu testar o aplicativo mais algumas vezes.

Entrada

Você irá testar o seu aplicativo com diversos nomes de países, simulando os dados informados pelo painel de navegação do trenó. A entrada termina por fim de arquivo.

Saída

O seu aplicativo deverá mostrar na tela a frase no idioma correto. Caso ela não esteja cadastrada, você deverá exibir a mensagem -- NOT FOUND --"para que depois dos testes você possa completar o banco de dados.

¹Corresponde ao problema 1763 - Tradutor do Papai Noel do Uri Online Judge - <https://www.urionlinejudge.com.br/judge/pt/problems/view/1763>.

Exemplo

Entrada
uri-online-judge alemanha brasil austria
Saída
-- NOT FOUND -- Frohliche Weihnachten! Feliz Natal! Frohe Weihnacht!

6 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

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 o x
x o o o o o o o o x
x o o o o o o o x
  x 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
6  * da matriz.
7  * @return int
8  */
9 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 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%

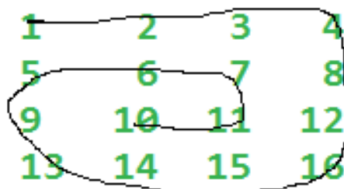
Entrada	Saída
3 3 1 2 3 2 0 4 1 0 0	0 33.33% 4 11.11%

10 Imprime matriz em espiral (+++)



(+++)

Faça um programa que leia uma matriz de inteiros de no máximo 10x10 elementos e a imprima de forma espiral. Por exemplo, dada a matriz abaixo, a ordem de impressão é expressa pela linha que forma uma espiral.



Entrada

A quantidade de linhas e colunas da matriz seguido dos elementos da matriz.

Saída

Uma linha contendo os elementos da matriz impressos na ordem de espiral. Caso a quantidade de linhas e colunas seja inválida, imprima a mensagem "Dimensao invalida\n"e finalize o programa.

Exemplo

Entrada	Saída
1 1 -2	Dimensao invalida

Entrada	Saída
1 5 1 2 3 4 5	1 2 3 4 5

Entrada	Saída
2 2 1 2 3 4	1 2 4 3

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

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

Entrada	Saída
4 4 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16	1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Entrada	Saída
6 3 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18	1 2 3 6 9 12 15 18 17 16 13 10 7 4 5 8 11 14

11 Soma e Subtração de Polinômios



(+++)

Faça um programa para somar ou subtrair polinômios

Entrada

A primeira linha da entrada corresponde ao número de casos de teste. Cada caso de teste é formado por várias linhas. A primeira linha contém o caractere '+' ou o caractere '-', indicando se a operação é, respectivamente de soma ou de subtração de polinômios. A próxima linha contém um número inteiro n_1 que indica o número de termos do primeiro polinômio. Em seguida, há n_1 ($n_1 \leq 50$) linhas cada uma contendo um par de números c e e separados entre si por um espaço. O número c é um valor float, com sinal e corresponde ao coeficiente de um termo do polinômio. O valor e é um número inteiro positivo e corresponde ao expoente do termo do polinômio. Após o par n_1 . Há um outro número inteiro n_2 ($n_2 \leq 50$) que corresponde ao número de termos do segundo polinômio, seguido, de $n - 2$ linhas contendo pares de coeficientes e expoentes, como explicando para o caso do primeiro polinômio. Os pares de cada polinômio estão ordenados em ordem decrescente de expoente.

Saída

Para cada caso de teste o programa deve imprimir uma linha contendo o polinômio resultante escrito no seguinte formato:

$$sc_1X \wedge e_n sc_2X \wedge e_{n-1} \cdot sc_nX \wedge e_1$$

onde $s \in \{'+', '-'\}$ é o sinal do coeficiente do polinômio, c_i é o valor do coeficiente e e_i é o valor do expoente do termo i . Se durante a operação se soma ou subtração de polinômios um dos termos do polinômio resultante ficar com coeficiente igual a zero, esse termo não deve aparecer no polinômio resultante.

Sugestão

Represente cada termo do polinômio como uma struct com dois campos: *coeficiente* e *expoente*. Represente cada polinômio de entrada e também o polinômio resultante como vetores de structs. Escreva uma função para cada uma das seguintes ações:

- Leitura de um polinômio
- Impressão de um polinômio
- Soma de dois polinômios
- Subtração de dois polinômios

Exemplo

Entrada
2 + 5 -0.5 7 +4.2 4 -3 2 +1 1 -4 0 4 +3 5 +2 4 -1 2 +3 0 + 2 -5 7 -3 0 6 +2 6 +2 5 +3 4 -2.5 3 +3 2 -1.2 1
Saída
$-0.50X^7 + 3.00X^5 + 6.20X^4 - 4.00X^2 + 1.00X^1 - 1.00$ $-5.00X^7 + 2.00X^6 + 2.00X^5 + 3.00X^4 - 2.50X^3 + 3.00X^2 - 1.20X^1 - 3.00$

12 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

13 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!