

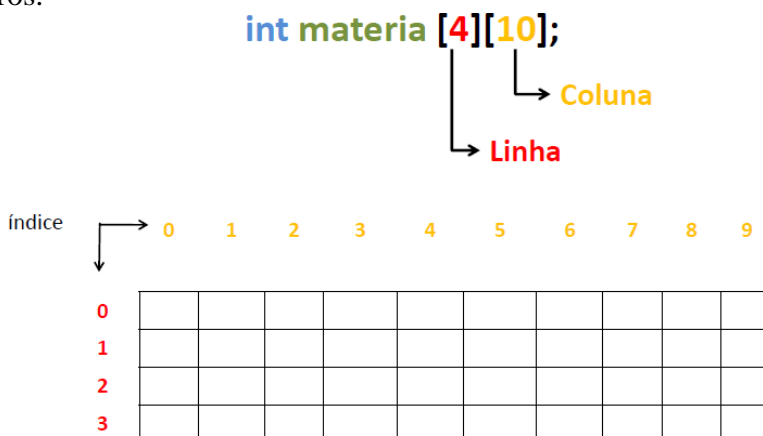
## Roteiro 6 - Matrizes

De forma semelhante aos Vetores, as matrizes também são **variáveis compostas homogêneas**, porém ao invés de **unidimensionais**, elas são multidimensionais. Também são capazes de armazenar vários valores de um mesmo tipo primitivo em um mesmo momento.

Cada um dos valores que compõem uma matriz são identificados pelo mesmo nome, sendo diferenciados apenas por meio de suas posições (índices) dentro da estrutura.

Em JAVA é possível a declaração de: Matrizes unidimensionais (vetores), bidimensionais, tridimensionais, etc. O limite de dimensões fica por conta da quantidade de recurso disponível pelo compilador (as matrizes mais utilizadas são de duas dimensões em que os índices representam as linhas e colunas).

Da mesma maneira como nos vetores, os índices começam por **0 (zero)**. O exemplo abaixo apresenta a declaração de uma variável chamada *materia* contendo 4 linhas (0 a 3) e 10 colunas (0 a 9), capazes de armazenar números inteiros.



### Declaração de uma matriz

A declaração de uma matriz é semelhante ao procedimento realizado para variáveis simples e vetores.

**tipo [ ][ ] nome = [tamanho\_da\_dim1] [tamanho\_da\_dim2];**

**tipo:** indica o tipo de cada elemento da matriz

**nome:** é o identificador da variável

**tamanho\_do\_dim1:** indica o tamanho da primeira dimensão

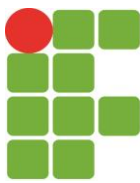
**tamanho\_do\_dim2:** indica o tamanho da segunda dimensão

### Exemplos:

```
int [ ][ ] nota = new int [10][5];
```

```
float [ ][ ] peso = new float [4][30];
```

No exemplos acima foram criados 2 matrizes: o primeiro com identificador **nota** de 10 linhas e 5 colunas que é capaz de armazenar valores inteiros; o segundo com nome **peso** de 4 linhas e 30 colunas que é capaz de armazenar valores reais.



### Preenchendo valor da matriz na declaração

```
int [][]idade = {{40, 50}, {80, 65}, {10, 15}, {50, 42}};
```

Declara-se uma matriz chamada idade com 4 linhas e 2 colunas.

### Atribuindo valores à uma matriz

Pode-se atribuir valores à uma matriz informando as posições na estrutura utilizando o comando direto de atribuição.

#### Exemplo:

```
peso[0][0] = 60;  
peso[0][1] = 55;  
.....  
peso[0][29] = 78;  
.....  
peso[3][29] = 80;
```

Porém, não é praticável preencher uma matriz dessa forma. Imagine preencher um vetor de 1000 x 1000 posições dessa forma?!

### Preenchendo uma matriz

Preencher uma matriz significa atribuir valores a todas suas posições. Assim, para preenchimento de uma matriz de forma mais prática utiliza-se estrutura de repetição.

#### Exemplo: matriz de duas dimensões (linhas e colunas)

```
int [][]matéria=new int[4][10];  
for (int i =0; i <matéria.length; i++) { //linhas  
    for (int j =0; j < matéria[i].length; j++) { //colunas  
        sout(" Digite o valor da linha"+ i+" coluna"+j);  
        matéria[i][j] = scan.nextInt( );  
    }  
}
```

### Imprimindo uma matriz

De forma semelhante ao procedimento de atribuição, a matriz pode ser impressa posição por posição diretamente.

#### Exemplo: imprimindo matriz peso do exemplo acima.

```
sout(peso[0][0]);
```

```
sout(peso[0][1]);  
.....  
sout(peso[0][29]);
```

Porém, essa não é uma tarefa praticável. Imagine imprimir uma matriz de 1000 x 1000 posições dessa forma?!

Portanto, para impressão de uma matriz de forma mais prática utiliza-se estrutura de repetição.

**Exemplo:** matriz de duas dimensões (linhas e colunas)

```
for (int i =0; i <4; i++) { //linhas  
    for (int j =0; i < 10; j++) { //colunas  
        sout(" Valor da linha"+ i+ " coluna"+j+ " é:  
"+materia[i][j]);  
    }  
}
```

**Exemplo:** impressão em outro formato de uma matriz de duas dimensões (linhas e colunas)

```
for (int i =0; i <4; i++) { //linhas  
    for (int j =0; i < 10; j++) { //colunas  
        sout(" "+materia[i][j]); //sem quebra de linha  
    }  
    sout(); //com quebra de linha  
}
```

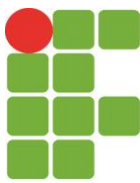
## Exemplos

**Exemplo 1:** Digite e compile o código abaixo. Nesse exemplo, faz-se o preenchimento e impressão de uma matriz de 5 notas e 5 disciplinas.

```
public class Random111 {  
    public static void main(String[] args) {  
        float [][] notas = new float[5][5];  
        Scanner scan = new Scanner(System.in);  
        for(int i=0; i<notas.length; i++){  
            for(int j=0; j<notas[i].length; j++){  
                System.out.println("Digite a nota do aluno "  
                    +(i+1)+" na dsiciplina "+(j+1)+":");  
                notas[i][j] = scan.nextFloat();  
            }  
        }  
        System.out.println("NOTAS DIGITADAS");  
        for(int i=0; i<notas.length; i++){  
            for(int j=0; j<notas[i].length; j++){  
                System.out.print(" "+notas[i][j]);  
            }  
            System.out.println("");  
        }  
    }  
}
```

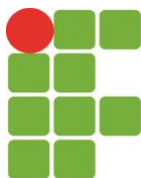
**Exemplo 2:** Digite e compile o exemplo abaixo. Nesse exemplo, faz-se o preenchimento de uma matriz de 3 notas de 5 alunos, imprimindo a soma e média das notas de cada aluno. Além disso, imprime a maior nota de cada aluno.

```
public class Random111 {  
    public static void main(String[] args) {  
        float[][] notas = new float[5][3];  
        float[] soma = new float[5]; float[] media=new float[5];  
        float[] maior=new float[5];  
        Scanner scan = new Scanner(System.in);  
        for(int i=0; i<notas.length; i++){  
            soma[i] = 0; //inicializa soma de cada aluno com zero  
            for(int j=0; j<notas[i].length; j++){  
                System.out.println("Digite a nota do aluno "  
                    +(i+1)+" na dsiciplina "+(j+1)+":");  
                notas[i][j] = scan.nextFloat();  
                soma[i] += notas[i][j];  
                if(j==0 || notas[i][j]>maior[i]){  
                    maior[i] = notas[i][j];  
                }  
            }  
            media[i] = soma[i]/3;  
        }  
        System.out.println("NOTAS DIGITADAS");  
        for(int i=0; i<notas.length; i++){  
            for(int j=0; j<notas[i].length; j++){  
                System.out.print(" "+notas[i][j]);  
            }  
            System.out.println("");  
        }  
        for(int i=0; i<notas.length; i++){  
            System.out.println("=====");  
            System.out.println("Soma do aluno "+(i+1)+" = "+soma[i]);  
            System.out.println("Média do aluno "+(i+1)+" = "+media[i]);  
            System.out.println("Maior nota do aluno "+(i+1)+" = "+maior[i]);  
            System.out.println("=====");  
        }  
    }  
}
```



**Exemplo 3:** Digite e compile o exemplo abaixo. Nesse exemplo, faz-se o preenchimento de uma matriz com três dimensões de tamanhos 4, 3 e 2, sendo que o valor de cada posição é a soma dos índices da mesma.

```
public class Random111 {  
    public static void main(String[] args) {  
        float[][][] matriz = new float[4][3][2];  
        for(int i=0; i<matriz.length; i++){  
            for(int j=0; j<matriz[i].length; j++){  
                for(int z=0; z<matriz[i][j].length; z++){  
                    matriz[i][j][z] = i + j + z;  
                }  
            }  
        }  
        for(int i=0; i<matriz.length; i++){  
            System.out.println("DIMENSÃO "+i);  
            for(int j=0; j<matriz[i].length; j++){  
                for(int z=0; z<matriz[i][j].length; z++){  
                    System.out.print(" "+matriz[i][j][z]);  
                }  
            }  
            System.out.println("");  
        }  
    }  
}
```



## Exercícios

1. Faça um programa que preencha uma matriz  $2 \times 2$ , calcule e mostre uma matriz resultante que será a matriz digitada multiplicada pelo maior elemento da mesma.
2. Faça um programa que preencha uma matriz  $3 \times 5$  com números reais. Some os valores de cada linha e armazene em um vetor. A seguir, mostre qual a linha que tem maior valor somado.
3. Faça um programa que preencha uma matriz  $A[3][4]$ , e outra matriz  $B[\text{linhas}][\text{colunas}]$  em que as dimensões são digitadas pelo usuário. Após isso, verifique se a operação de multiplicação de matrizes ( $A \times B$ ) pode ser realizada, caso seja possível, realize a operação e imprima a matriz resultante.
4. A produção diária de uma fábrica é armazenada, semana a semana, em uma tabela com o seguinte formato.

	Dia1	Dia2	Dia3	Dia4	Dia5	Dia6	Dia7
Semana 1							
Semana 2							
Semana 3							
....							
Semana N							

Faça um programa que a partir dos dados contidos na tabela gere o seguinte relatório:

### RELATÓRIO DE PRODUÇÃO RELATIVO À N SEMANAS

Produção média = xxxx.x

Número de dias com produção acima da média = xx

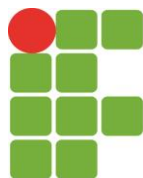
### INDICAÇÃO DOS DIAS DE MÍNIMA PRODUÇÃO:

Semana 1 ..... DIA x

Semana 2 ..... DIA x

Semana 3 ..... DIA x

5. Faça um programa que:
  - Receba as notas de 10 alunos em cinco provas diferentes e armazene-as em uma matriz  $10 \times 5$ .
  - Receba o nome dos 10 alunos e armazene-os em um vetor de string com 10 posições
  - Calcule e exiba o nome do aluno, a média das notas do mesmo e a situação:
    - **Aprovado:** média maior ou igual a 7
    - **Reprovado:** média menor que 4
    - **Exame Especial:** média entre 4 e 7



6. O desvio padrão de uma amostra de dados calcula o quanto de variação existe da amostra em relação à média. Valores baixos indicam que os dados tendem a estar próximos à média, por outro lado, valores altos indicam maior dispersão dos dados. O gerente de produção da Refrigerator Tabajará está avaliando o processo de fabricação de seus refrigerantes em lata, especificamente a etapa de preenchimento do conteúdo. As latas devem possuir 390 ml, com desvio padrão de + ou - 5 ml. Na análise foram coletadas 20 amostras aleatórias das 6 linhas de produção da fábrica, medindo-se a quantidade de produto das mesmas. Você é o programador da empresa, e ficou incumbido de implementar um programa que auxilie seu gerente, mostrando ao mesmo a média e o desvio padrão de cada linha de produção. Além disso, exiba, se for o caso, qual das linhas de produção precisa ser revisada.
7. Na teoria de sistemas define-se o elemento MINMAX de uma matriz como sendo o maior elemento da linha onde se encontra o menor elemento da matriz. Faça um programa que carregue uma matriz 4 x 7 com números reais, calcule e mostre o MINMAX e sua posição (linha e coluna).
8. Uma matriz quadrada de dimensão  $n$  é dita uma matriz permutação se em cada linha e em cada coluna houver  $n-1$  elementos nulos e um único elemento igual a 1. Dada uma matriz digitada pelo usuário, informe se a mesma é ou não uma matriz permutação.

Ex.: Matriz permutação: 
$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

9. Faça um programa que carregue uma matriz 8 x 8 com números reais e mostre uma mensagem dizendo se a matriz é ou não simétrica. Uma matriz é simétrica se para todas as posições da mesma, tem-se a relação  $A[i][j] = A[j][i]$ .
10. Os elementos  $a_{ij}$  de uma matriz  $A_{n \times n}$  representam os custos de transporte da cidade  $i$  para a cidade  $j$ . O caminhão de uma empresa de logística deve passar exatamente uma vez por cada uma das  $n$  cidades. Sabendo que, uma rota é representada por um vetor de  $n$  posições, em que cada posição representa a cidade visitada, você deve implementar um algoritmo que calcule o custo da rota de um caminhão da empresa. Note que, a rota e a matriz de custo devem ser fornecidas pelo usuário.

**Exemplo:** Considere 4 cidades em que os custos de transporte entre as mesmas são representados pela matriz de custo  $M$  abaixo. Considere ainda uma rota denotada pelo vetor  $= [3 \ 1 \ 0 \ 2]$  significando que o caminhão parte da cidade 3 com destino à cidade 1; sai de 1 e vai para 0; e por fim parte da cidade 0 para a cidade 2. Portanto, o custo de transporte dessa rota é:  $M[3][1] + M[1][0] + M[0][2] = 19 + 15 + 16 = 50$  unidades.

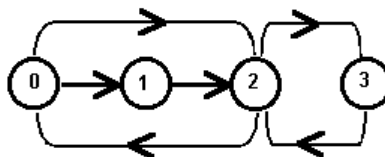
Matriz de custos  $M = \begin{bmatrix} 0 & 10 & 16 & 20 \\ 15 & 0 & 25 & 18 \\ 12 & 14 & 0 & 20 \\ 20 & 19 & 35 & 0 \end{bmatrix}$

11. Considere  $n$  cidades numeradas de 0 a  $n-1$  que estão interligadas por uma série de estradas de mão única. As ligações entre as cidades são representadas pelos elementos de uma matriz quadrada  $L_{n \times n}$ , cujos elementos  $l_{ij}$  assumem o valor 1 ou 0, conforme exista ou não estrada direta que saia da cidade  $i$  e chegue à cidade  $j$ . Assim, os elementos da linha  $i$  indicam as estradas que saem da cidade  $i$ , e os elementos da



coluna  $j$  indicam as estradas que chegam à cidade  $j$ . Por convenção  $l_{ii} = 1$ . A figura abaixo um exemplo para  $n = 4$ .

$$L = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$



- Dada uma cidade  $k$ , determine quantas estradas saem de  $k$  e quantas cidades chegam à  $k$ .
- Qual das cidades possuem maior número de estradas chegando?
- Encontre, se existirem, as cidades isoladas. Ou seja, aquelas em que não chegam nem saem estradas.
- Dada uma determinada rota de  $m$  cidades, representada por um vetor de inteiros cujos valores estão entre 0 e  $n-1$ , tal como exercício anterior, verifique se é possível realizar tal rota. No exemplo anterior, a rota ( $m = 5$ ) 2 3 2 1 0 é impossível; já a rota ( $m = 6$ ) 0 1 2 3 2 0 é possível.

12. Faça um programa que receba o estoque atual de 3 produtos que estão armazenados em 4 armazéns e coloque esses dados em uma matriz  $5 \times 3$ . Sendo que a última linha da matriz contém o custo de estocagem de uma unidade de cada produto, calcule e mostre:

- Quantidade de itens armazenados em cada armazém;
- Qual armazém possui maior estoque do produto 2;
- Qual armazém possui menor estoque;
- Qual armazém possui maior custo de estocagem.

13. Faça um programa que leia o nome completo e o apelido de uma pessoa. Imprima: a) quantidade de caracteres no nome completo da pessoa; b) Concatene o apelido ao nome do usuário; c) Verifique se o nome do usuário é igual ao seu apelido. Funções da biblioteca string: strcmp, strcat, strlen.

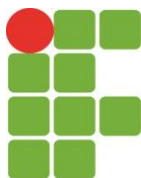
14. Uma empresa de possui ônibus com 48 lugares (24 nas janelas e 24 no corredor). Faça um programa que utilize uma matriz para controlar as poltronas ocupadas no corredor e janela. Considere que 0 representa poltrona desocupada e 1, poltrona ocupada. Inicialmente, todas as poltronas estarão livres. Depois disso, o programa deverá apresentar as seguintes opções:

```
=====
                        MARIAS UNIDAS
=====
a. Vender passagem
b. Mostrar mapa de ocupação do ônibus
c. Encerrar
=====
```

- Quando a opção escolhida for a letra 'a', deve-se verificar se o ônibus ainda possui poltronas a serem vendidas. Caso haja poltrona vazia, deve ser perguntado ao usuário qual poltrona o mesmo deseja ocupar. O programa deverá, então dar uma das seguintes mensagens: 1) Venda efetivada – se a poltrona solicitada estiver livre, marcando-a como ocupada; 2) Poltrona ocupada – se a poltrona solicitada não estiver disponível para venda.

- Quando a opção escolhida for letra 'b', deverá ser mostrada uma listagem separando as poltronas da janela e corredor, o número da poltrona e se está livre ou ocupada.

- Quando for escolhida a opção letra 'c', a execução do programa deverá ser finalizada.



15. O gerente salão de beleza **Maria Saliente** tem uma tabela em que registra os serviços realizados por suas 5 manicures. Os serviços realizados pelas mesmas são: “unha dos pés”, “unhas das mãos” e serviço de podologia. Como nenhuma das 5 manicures possuem carteira assinada, elas ganham 50% do que faturou ao mês. Crie um algoritmo que faça o controle, calcule e imprima quanto cada uma vai receber por cada tipo de serviço prestado. Os valores pagos para cada serviço são, respectivamente, R\$ 10,00; R\$ 15,00 e R\$ 30,00.
16. Uma floricultura conhecedora de sua clientela gostaria de fazer um algoritmo que pudesse controlar sempre um estoque mínimo de determinadas plantas, pois todos os dias, pela manhã, o dono faz novas aquisições. Crie um algoritmo que possua as seguintes opções:

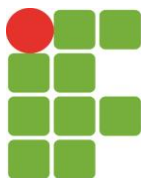
```
=====
                        FLORICULTURA MARIASFLOR
=====
1. CADASTRAR NOVA PLANTA
2. RETIRAR PLANTA
3. INSERIR PLANTA
4. IMPRIMIR RELATÓRIO
5. SAIR
=====
```

O algoritmo deve permitir o cadastro de 50 tipos de plantas. Na opção de cadastro, para cada planta, o algoritmo deve cadastrar um código (inteiro), o nome (string), o estoque ideal e a quantidade atual em estoque. Na opção retirar planta, o sistema deve permitir a retirada do estoque atual de uma planta, dado o código digitado pelo usuário, lembre-se de verificar se o estoque é suficiente para atender o pedido. Para a opção de inserção, seu algoritmo deve atualizar o estoque atual de uma certa planta, dadas as informações de quantidade comprada e do código da planta que são digitadas pelo usuário. Na opção de imprimir relatório, seu algoritmo deve imprimir os nomes, os estoques atuais e as quantidades a serem compradas das plantas que possuem estoque abaixo do ideal.

17. Uma rede social de amigos pode ser representada por uma matriz quadrada (mesmo número de linhas e colunas) de dimensão número de pessoas da rede social. A amizade entre as pessoas presentes nessa rede se dá da seguinte forma: suponha que  $n_0$ ,  $n_1$ ,  $n_2$ ... representa uma pessoa e, caso duas pessoas sejam amigas, então a posição da matriz será 1, caso contrário será 0 (zero). Considere a matriz de exemplo abaixo:

id	$n_0$	$n_1$	$n_2$	$n_3$	$n_4$
$n_0$	0	1	1	0	1
$n_1$	1	0	0	1	0
$n_2$	1	0	0	0	0
$n_3$	0	1	0	0	1
$n_4$	1	0	0	1	0

Essa matriz representa uma rede social entre 5 pessoas:  $n_0$ ,  $n_1$ ,  $n_2$ ,  $n_3$  e  $n_4$ . Além disso, quando a posição  $(i, j)$  da matriz é 1, então as pessoas  $n_i$  e  $n_j$  são amigas entre si. Caso a posição  $(i, j)$  da matriz é 0, então  $n_i$  e  $n_j$  não são amigas. Observe que a pessoa  $n_0$  é amiga das pessoas  $n_1$ ,  $n_2$  e  $n_4$ , mas não é amiga da pessoa  $n_3$ . Importante: a relação de amizade é simétrica: se  $n_i$  é amigo de  $n_j$ , então  $n_j$  é,

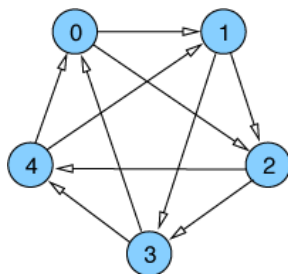


necessariamente, amigo de si. Além disso, em redes sociais de amizade, não existe amizade da pessoa com ela mesma. Sabendo disso, implemente um algoritmo que:

- preencha a matriz de amizades de uma rede social contendo 5 pessoas;
- calcule e imprima a quantidade de amigos que cada pessoa possui.

## EXERCÍCIOS ESTILO OLIMPÍADAS

18. A brincadeira da Pedra, Papel e Tesoura, muita gente conhece. Mas dá para fazer uma mais legal com cinco opções e não só três! Dois jogadores, dario e xerxes, jogam uma partida com N rodadas. Em cada rodada os jogadores escolhem uma "mão" entre cinco opções, que vamos representar aqui com os números 0, 1, 2, 3 e 4. A figura define exatamente quem ganha a rodada. Por exemplo, se dario escolheu 0 e xerxes escolheu 3, então xerxes ganha a rodada, pois existe uma seta na figura indo de 3 para 0.



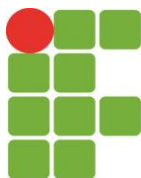
Depois de N rodadas, o vencedor da partida é o jogador que ganhou mais rodadas. O número N será sempre ímpar, para não haver empate na partida. Vamos também considerar que os jogadores nunca escolhem a mesma mão numa rodada, para não haver empate na rodada. Você deve escrever um programa que determine quem venceu a partida, se foi dario ou xerxes.

**ENTRADA:** primeira linha da entrada contém um inteiro N, o número de rodadas na partida. Cada uma das N linhas seguintes contém dois inteiros D e X, representando a mão que os jogadores dario e xerxes, respectivamente, jogaram em uma rodada.

**SAÍDA:** Seu programa deve imprimir uma linha contendo o nome do jogador que venceu a partida: dario ou xerxes. Todas as letras devem ser minúsculas, sem nenhum acento!

### EXEMPLOS

<b>Entrada</b> 3 1 3 4 2 0 2	<b>Saída</b> dario
<b>Entrada</b> 1 3 1	<b>Saída</b> xerxes



19. Harry ganhou um mapa mágico no qual ele pode visualizar o trajeto realizado por seus amigos. Ele agora precisa de sua colaboração para, com a ajuda do mapa, determinar onde Hermione se encontra. O mapa tem L linhas e C colunas de caracteres, que podem ser '.' (ponto), a letra 'o' (minúscula) ou a letra 'H' (maiúscula). A posição inicial de Hermione no mapa é indicada pela letra 'o', que aparece exatamente uma vez no mapa. A letra 'H' indica uma posição em que Hermione pode ter passado -- o mapa é impreciso, e nem toda letra 'H' no mapa representa realmente uma posição pela qual Hermione passou. Mas todas as posições pelas quais Hermione passou são representadas pela letra 'H' no mapa. A partir da posição inicial de Hermione, Harry sabe determinar a posição atual de sua amiga, apesar da imprecisão do mapa, porque eles combinaram que Hermione somente se moveria de forma que seu movimento apareceria no mapa como estritamente horizontal ou estritamente vertical (nunca diagonal). Além disso, Hermione combinou que não se moveria de forma a deixar que Harry tivesse dúvidas sobre seu caminho (por exemplo, Hermione não passa duas vezes pela mesma posição). Considere o mapa abaixo, com 6 linhas e 7 colunas:

	1	2	3	4	5	6	7
1	.	.	.	<b>H</b>	<b>H</b>	<b>H</b>	.
2	H	H	H	.	.	.	<b>H</b>
3	H	.	H	H	H	.	.
4	H	.	.	.	H	H	.
5	H	.	o	.	.	.	.
6	H	H	H	.	.	<b>H</b>	<b>H</b>

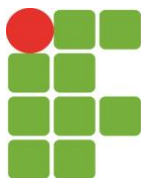
A posição inicial de Hermione no mapa é (5,3), e sua posição atual é (4,6). As posições marcadas em negrito ('H') são erros no mapa. Dado um mapa e a posição inicial de Herminone, você deve escrever um programa para determinar a posição atual de Herminone.

**ENTRADA** A primeira linha contém dois números inteiros L e C, indicando respectivamente o número de linhas e o número de colunas. Cada uma das seguintes L linhas contém C caracteres.

**SAÍDA** Seu programa deve produzir uma única linha na saída, contendo dois números inteiros: o número da linha e o número da coluna da posição atual de Hermione.

## EXEMPLOS

<b>Entrada</b> 3 4 HHHH H... o.HH	<b>Saída</b> 1 4
<b>Entrada</b> 6 7 ...HHH. HHH....	<b>Saída</b> 4 6



H.HHH.. H...HH. H.o.... HHH.HH.	
--	--

20. Joãozinho está aprendendo sobre matrizes. Hoje ele aprendeu como deixar matrizes na forma escada, e está exercitando. Para ajudá-lo, você deve escrever um programa que determine se o resultado dele realmente está no formato correto. Uma matriz está na forma escada quando, para cada linha, as condições a seguir forem satisfeitas:

- Se a linha só possuir zeros, então todas as linhas abaixo desta também só possuem zeros.
- Caso contrário, seja X o elemento diferente de zero mais à esquerda da linha; então, para todas as linhas abaixo da linha de X, todos os elementos nas colunas à esquerda de X e na coluna de X são iguais a zero.

**ENTRADA:** A primeira linha possui dois inteiros N e M, as dimensões da matriz. Cada uma das N linhas seguintes contém M inteiros não-negativos, os elementos da matriz.

**SAÍDA:** Seu programa deve produzir uma única linha, contendo o caractere 'S' caso a matriz esteja no formato escada, ou 'N', caso contrário.

## EXEMPLOS

Entrada	Saída
4 6 1 2 9 9 9 9 0 0 3 9 9 9 0 0 0 0 5 9 0 0 0 0 0 6	S

Entrada	Saída
5 8 0 5 1 0 3 2 2 0 0 0 0 0 4 0 1 2 0 0 0 0 0 0 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	S



**INSTITUTO FEDERAL DE MINAS GERAIS**  
**Campus Sabará**

Entrada	Saída
5 5 1 1 2 3 4 0 1 1 4 5 0 1 2 3 6 0 0 0 2 0 0 0 0 0 0	N

Entrada	Saída
5 5 1 1 2 3 4 0 1 1 4 5 0 1 2 3 6 0 0 0 2 0 0 0 0 0 0	N

Entrada	Saída
5 5 1 1 2 3 4 0 1 1 4 5 0 1 2 3 6 0 0 0 2 0 0 0 0 0 0	N