

# Matrizes

Prof. Karl Apaza Agüero

# Matrizes

- ▶ Matrizes nada mais são do que vetores com múltiplas dimensões.
- ▶ Exemplo de uma matriz 2D de tamanho 3x6:

	0	1	2	3	4	5
0	a	b	c	d	e	f
1	g	h	i	j	k	l
2	m	n	o	p	q	r

- ▶ O primeiro tamanho (3) corresponde à quantidade de linhas, e o segundo (6) à quantidade de colunas.
- ▶ Os índices verticais identificam a posição em relação às linhas na matriz.
- ▶ Os índices horizontais identificam a posição em relação às colunas na matriz.

# Matrizes

## ► Declaração de uma matriz 2D:

```
<tipo de dado> <nome da variável> [tamanho1][tamanho2];
```

```
int m[5][5];
```

```
char tabela[1000][1000];
```

```
float notas[8][3];
```

## ► Manipulação da matriz:

```
m[1][3] = 3;  
scanf("%d", &m[1][3]);  
printf("%d", m[1][3]);
```

# Matrizes multidimensionais

- ▶ Como declarar matrizes multidimensionais:

```
tipo nome[tamanho1][tamanho2][tamanho3]....[tamanhoN];
```

- ▶ Exemplo de inicialização de uma matriz 3D:

```
int i, j, k, altura=10; largura=20; comprimento=30;  
int mat3D[altura][largura][comprimento]
```

```
for(i=0; i < altura; i++)  
    for(j=0; j < largura; j++)  
        for(k=0; k < comprimento; k++)  
            mat3D[ i ][ j ][ k ] = 0;
```

# Exemplo: Ler e imprimir uma matriz 2D

```
#include <iostream>
using namespace std;
int main() {

    int N, M, i, j;
    cin>>N>>M;
    int A[N][M];

    // Lê a matriz
    for (i=0; i < N; i++){
        for (j=0; j < M; j++){
            cin>>A[i][j];
        }
    }

    // Imprime a matriz em uma linha
    for (i=0; i < N; i++){
        for (j=0; j < M; j++){
            cout<<A[i][j]<<" ";
        }
    }

    return 0;
}
```

# Matrizes

## ► Problema

Fazer um programa para ingressar duas matrizes de tamanho  $n \times m$  e imprimir o resultado da soma destas matrizes.

Nota: apresentar os resultados como uma matriz  $n \times m$ .

Entrada:

2 3

1 2 3 4 5 6

7 8 9 10 11 12

Saída:

8 10 12

14 16 18

# Matrizes

```
#include <iostream>
using namespace std;
int main() {

    int N, M, i, j;
    cin>>N>>M;
    int A[N][M], B[N][M];

    // Lê a matriz A
    for (i=0; i < N; i++){
        for (j=0; j < M; j++){
            cin>>A[ i ][ j ];
        }
    }
```

```
// Lê a matriz B e soma A
    for (i=0; i < N; i++){
        for (j=0; j < M; j++){
            cin>>B[ i ][ j ];
            B[ i ][ j ] += A[ i ][ j ];
        }
    }

    // Imprime a soma em formato de
    // matriz
    for (i=0; i < N; i++) {
        for (j=0; j < M-1; j++){
            cout<<B[ i ][ j ]<<" ";
        }
        cout<<B[ i ][ j ]<<endl;
    }

    return 0;
}
```

# Matrizes

## ► Problema

Vidal e Bira estavam jogando batalha naval. Porém, nessa variação do jogo, Vidal tinha um campo de tamanho  $N \times M$  com  $K$  navios e Bira tinha apenas um canhão capaz de atirar  $C$  vezes. Bira vence o jogo caso afunde metade ou mais da metade dos navios de Vidal. Dada a disposição dos navios de Vidal no campo e as posições onde Bira atirou, diga quem venceu o jogo. É garantido que o número de navios no campo de Vidal no início do jogo é par.

### Entrada

A primeira linha da entrada possui quatro inteiros  $N$ ,  $M$  e  $C$ , os dois primeiros indicando o tamanho do campo de Vidal e o terceiro indicando a quantidade de tiros de Bira. Nas próximas  $N$  linhas haverá  $M$  inteiros que podem ser 0 ou 1. 1 indica que existe um navio na posição dada, enquanto 0 indica que aquela posição é água. Nas próximas  $C$  linhas haverá dois inteiros  $X$  e  $Y$ , indicando as coordenadas de um tiro dado por Bira.

### Saída

A saída deve ter apenas uma palavra: “Bira”, caso Bira tenha ganhado o jogo, ou “Vidal” caso contrário.

### Entrada

```
3 3 3
0 0 1
1 0 0
0 1 1
0 2
1 0
2 2
```

Saída  
Bira

### Entrada

```
4 3 5
1 0 1
0 1 1
1 0 1
0 0 0
0 0
3 0
0 2
3 1
1 0
```

Saída  
Vidal

### Entrada

```
3 3 3
0 0 1
1 0 0
0 1 1
0 2
0 2
0 2
```

Saída  
Vidal



# Matrizes

```
#include <iostream>
using namespace std;
int main(){
    int n,m,c;
    cin>>n>>m>>c;
    int campo[n][m], i, j, navios=0;

    for(i=0; i<n; i++){
        for(j=0; j<m; j++){
            cin>>campo[i][j];
            if(campo[i][j]==1){
                navios++;
            }
        }
    }
}
```

```
int navios_afundados=0;
for(int t=0; t<c; t++){
    cin>>i>>j;
    if(campo[i][j]==1){
        navios_afundados++;
        campo[i][j]=0;
    }
}

if(navios_afundados>=navios/2){
    cout<<"Bira"<<endl;
}
else{
    cout<<"Vidal"<<endl;
}

return 0;
}
```

# Matrizes

## ► Problema

Fazer um programa para ingressar duas matrizes de tamanho 3x3 e imprimir o resultado da multiplicação destas matrizes.

Nota: apresentar os resultados como uma matriz 3x3.

Entrada:

1 0 1

2 1 1

1 2 1

1 2 2

2 1 0

0 0 1

Saída:

1 2 3

4 5 5

5 4 3

# Matrizes

```
#include <iostream>
using namespace std;
#define N 3
int main() {

    int A[N][N], B[N][N], C[N][N];
    int i,j,k;
    for(i=0; i<N; i++){
        for(j=0; j<N; j++){
            cin>>A[i][j];
        }
    }

    for(i=0; i<N; i++){
        for(j=0; j<N; j++){
            cin>>B[i][j];
        }
    }
}
```

```
for(i=0; i<N; i++){
    for(j=0; j<N; j++){
        C[i][j]=0;
        for(k=0; k<N; k++){
            C[i][j]+=(A[i][k]*B[k][j]);
        }
    }
}

for(i=0; i<N; i++){
    for(j=0; j<N-1; j++){
        cout<<C[i][j]<<" ";
    }
    cout<<C[i][j]<<endl;
}

return 0;
}
```