

## Unidad 4: Array de array - Matrices

<http://bit.ly/2HRBWgq>

Profesores:

Ernesto Cuadros- Vargas, PhD.

[ecuadros@utec.edu.pe](mailto:ecuadros@utec.edu.pe)

María Hilda Bermejo, M. Sc.

[mbermejo@utec.edu.pe](mailto:mbermejo@utec.edu.pe)

# Telegram:

**1. Configurar tu cuenta**

**2. <http://bit.ly/2TJnwBq>**

# Logro de la sesión:

**Al finalizar la sesión, los alumnos desarrollan sus programas utilizando arrays de arrays.**

# **Arrays de arrays (Matrices)**

---

# Lenguas en el mundo

Busca en la sopa de letras 16 lenguas del mundo

Es una columna

C	A	S	T	E	L	L	A	N	O	O	N	K	S	D	C
G	X	J	O	Y	P	K	E	O	C	Y	W	I	R	W	A
V	Y	I	E	E	X	Y	R	U	H	O	G	W	N	I	T
R	K	B	Y	U	I	E	E	V	I	F	A	E	M	Y	A
I	I	Y	V	P	O	L	O	B	N	D	L	W	C	W	L
T	O	E	X	J	A	S	I	F	O	Z	L	O	T	H	Á
A	O	Y	E	U	I	W	L	H	S	V	E	G	C	I	N
L	O	M	L	A	J	S	T	E	B	O	G	K	U	N	E
I	N	G	L	É	S	V	J	A	O	I	O	P	Y	D	U
A	A	T	B	E	B	K	I	R	O	Y	J	O	Y	I	S
N	C	N	I	T	A	L	I	A	N	O	A	X	Y	I	K
O	A	R	U	M	A	N	O	B	K	N	P	T	Y	O	E
O	X	I	E	W	W	Y	B	E	U	Q	O	U	C	O	R
I	R	P	O	R	T	U	G	U	É	S	N	I	E	N	A
C	O	G	F	R	A	N	C	É	S	R	É	U	F	R	P
A	L	E	M	Á	N	X	A	I	R	U	S	O	V	A	Z

Es una fila

Francés  
Euskera  
Gallego  
Ruso  
Italiano

Castellano  
Japonés  
Hindi  
Rumano  
Catalán

Arabe  
Inglés  
Chino  
Portugués  
Italiano

Alemán



`char Pupiletras[16][16];`

Lo que solía denominarse arrays multidimensionales son actualmente arrays de arrays.

Definiciones:

**int ia[3][4];** // array of size 3, each element is an arrays of ints of size 4

**int arr[10][20][30] = {0};**

**// array of size 10; each element is a 20-element arrays whose element are arrays of 30 ints.**

**// initialize all elements to 0**

## Inicialización de los elementos de un array:

```
int ia[3][4]={      //three elements, each element is an array of size 4  
    {0, 1, 2, 3},    // initializers for the row indexed by 0  
    {4, 5, 6, 7},    // initializers for the row indexed by 1  
    {8,9,10,11}     // initializers for the row indexed by 2  
};
```

		0	1	2	3
ia	0	0	1	2	3
	1	4	5	6	7
	2	8	9	10	11

ia	0	1	2	3	4	5	6	7	8	9	10	11
----	---	---	---	---	---	---	---	---	---	---	----	----



```
int ia[3][4] = { { 0 }, { 4 }, { 8 } };  
// explicitly initialize only element 0 in each row
```

		0	1	2	3
ia	0	0			
	1	4			
	2	8			

```
int ix[3][4] = { 0, 3, 6, 9 };  
// explicitly initialize row 0; the remaining elements are value initialized.  
The remaining elements are initialized to 0.
```

		0	1	2	3
ia	0	0	3	6	9
	1	0	0	0	0
	2	0	0	0	0

```
constexpr size_t rowCnt = 3, colCnt = 4;
```

```
int ia[rowCnt][colCnt]; // 12 uninitialized elements
```

```
// for each row
```

```
for( size_t i=0; i<rowCnt; i++)
```

```
    //--- for each column within row
```

```
    for( size_t j=0; j<colCnt; j++)
```

```
        ia[i][j] = i*colCnt + j;
```

```
    // Assign the element's positional index as its value
```

0	1	2	3
4	5	6	7
8	9	10	11

Ejemplo:

```
int Dosmil48[ 4 ][ 4 ];
```

filas      columns

	0	1	2	3
0	4			
1			4	8
2		4	32	64
3	4	2	4	4

Orden de la matriz 4x4

**Pasando un array multidimensional a funciones:**

```
void funcion1( int matrix[][10], int rowSize, int colSize);
```

**El parámetro es un puntero a un array de 10 enteros.**

**El array nunca se pasa por valor, lo que envía es el puntero al primer elemento**

## Cómo se asigna un dato a un casillero de la matriz?

```
int Dosmil48[ 4 ] [ 4 ];
```

	0	1	2	3
0	4			
1			4	8
2		4	32	64
3	4	2	4	4

`Dosmil48[2][1] = 4;`

`cout << Dosmil48[2][3];`

## Recorrido fila por fila:

	0	1	2	3
0	4			
1			4	8
2		4	32	64
3	4	2	4	4

**Recorrido columna por columna:**

	0	1	2	3
0	4			
1			4	8
2		4	32	64
3	4	2	4	4

Por ejemplo cuando se quiere leer datos desde el teclado y almacenarlos en la matriz.

```
int Dosmil48[ 4 ] [ 4 ];
```

Se recorre la matriz fila por fila

	0	1	2	3
0	4			
1			4	8
2		4	32	64
3	4	2	4	4

```
for(int f=0; f<4; f++)  
{  
    for(int c=0; c<4; c++)  
    { cout << "Dosmil48[" << f << "][" << c << "]=";  
      cin >> Dosmil48[f][c];  
    }  
}
```



## Ejemplo: 1

Desarrolle un programa que permita generar aleatoriamente números enteros (entre 0 y 99), los almacene en un array de arrays, cuyas características se muestra en la figura:

	M							
	0	1	2	3	4	5	6	7
0	22	45	67	12	34	21	3	56
1	21	45	65	76	77	89	99	88
2	5	65	34	52	21	33	45	23
3	21	56	78	80	32	45	22	27
4	45	67	56	23	45	4	43	87
5	55	23	23	4	12	3	23	65
6	78	45	45	12	34	23	22	43
7	12	23	28	33	65	12	12	44

continúa

**Luego realice lo siguiente:**

- 1.Imprima el array de arrays.**
- 2.Hallar el dato más pequeño del array de arrays.**
- 3.Imprimir la suma de los elementos de la diagonal**
- 4.Hallar la suma de los elementos que están por encima de la diagonal.**

**Ejemplo de la Salida del programa:**

65	29	75	29	68	98	70
8	38	63	13	18	71	42
94	69	46	38	4	27	57
97	93	59	45	97	9	79
23	72	79	42	36	72	53
14	65	27	60	92	93	7
94	28	99	61	37	79	24

El menor elemento almacenado en el array es 4

La suma de la diagonal es igual a 347

Suma de los elementos por encima de la diagonal : 1019

**El código está distribuido en estos  
archivos:**

**main.cpp**

**Arrays.h**

**Arrays.cpp**

## Arrays.h

```
#ifndef EJEMPLO1_RECORRIDOS_ARRAYS_H
#define EJEMPLO1_RECORRIDOS_ARRAYS_H

#include <iostream>
#include <cstdint> //-- para usar size_t
using namespace std;

typedef long int TipoEntero;
constexpr size_t nFILAS = 8, nCOLUMNAS = 8;

void LlenarArray(TipoEntero M[][nCOLUMNAS], size_t filas, size_t columnas);
void ImprimirArray(TipoEntero M[][nCOLUMNAS], size_t filas, size_t columnas);
TipoEntero ElMenor(TipoEntero M[][nCOLUMNAS], size_t filas, size_t columnas);
TipoEntero SumaDiagonal(TipoEntero M[][nCOLUMNAS], size_t filas, size_t columnas);
TipoEntero SumaPorEncimadelaDiagonal(TipoEntero M[][nCOLUMNAS], size_t filas, size_t
columnas);

#endif //EJEMPLO1_RECORRIDOS_ARRAYS_H
```

## main.cpp

```
#include <iostream>
#include <ctime>
#include "Arrays.h"
using namespace std;

int main()
{
    TipoEntero M[nFILAS][nCOLUMNAS];

    srand(time(nullptr));
    cout<<"\n";
    LlenarArray(M,nFILAS, nCOLUMNAS);
    ImprimirArray(M,nFILAS, nCOLUMNAS);
    cout<<"\n";
    cout<<"El menor elemento almacenado en el array es " << EIMenor(M,nFILAS,nCOLUMNAS);
    cout<<"\n";
    cout<<"La suma de la diagonal es igual a " << SumadeDiagonal(M,nFILAS,nCOLUMNAS);
    cout<<"\n";
    cout<<"Suma de los elementos por encima de la diagonal : "
    << SumaPorEncimadelaDiagonal(M,nFILAS,nCOLUMNAS);
    cout<<"\n";
    return 0;
}
```

## Arrays.cpp

```
#include <cstdlib>
#include <iomanip>
#include "Arrays.h"
```

```
void LlenarArray(TipoEntero M[][nCOLUMNAS],size_t filas, size_t columnas)
{
    //-----
    for(size_t f=0; f<filas; f++)
        for(size_t c=0; c<columnas; c++)
            M[f][c] = rand()%100;
}
```

```
void ImprimirArray(TipoEntero M[][nCOLUMNAS],size_t filas, size_t columnas)
{
    //-----
    for(size_t f=0; f<filas; f++)
    {
        for (size_t c = 0; c < columnas; c++)
            cout << setw(5) << M[f][c];
        cout<<"\n";
    }
}
```

## Arrays.cpp

```
TipoEntero ElMenor(TipoEntero M[][nCOLUMNAS], size_t filas, size_t columnas)
```

```
{//-----
```

```
    TipoEntero Menor;
```

```
    Menor=M[0][0];
```

```
    for(size_t f=0; f<filas; f++)
```

```
        for (size_t c = 0; c < columnas; c++)
```

```
            if(M[f][c]<Menor)
```

```
                Menor =M[f][c] ;
```

```
    return Menor;
```

```
}
```

```
TipoEntero SumadeDiagonal(TipoEntero M[][nCOLUMNAS], size_t filas, size_t columnas)
```

```
{//-----
```

```
    TipoEntero Suma=0;
```

```
    for(size_t f=0; f<filas; f++)
```

```
        Suma+=M[f][f];
```

```
    return Suma;
```

```
}
```

# Arrays.cpp

```
TipoEntero SumaPorEncimadelaDiagonal(TipoEntero M[ ][nCOLUMNAS], size_t filas, size_t columnas)
{
    //-----
    TipoEntero Suma=0;

    for(size_t f=0; f<filas-1; f++)
        for (size_t c = f+1; c < columnas; c++)
            Suma+=M[f][c];
    return Suma;
}
```

	M							
	0	1	2	3	4	5	6	7
0	22	45	67	12	34	21	3	56
1	21	45	65	76	77	89	99	88
2	5	65	34	52	21	33	45	23
3	21	56	78	80	32	45	22	27
4	45	67	56	23	45	4	43	87
5	55	23	23	4	12	3	23	65
6	78	45	45	12	34	23	22	43
7	12	23	28	33	65	12	12	44



## Ejemplo: 2

**Desarrolle un programa que permita multiplicar dos matrices cuyo orden es de 5 por 5.**

**Los valores para la matriz 1 y la matriz 2, se generarán aleatoriamente con números entre el 0 y el 99.**

**El algoritmo para multiplicar matrices:**

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1(3) + 0(2) + 2(1) & 1(1) + 0(1) + 2(0) \\ -1(3) + 3(2) + 1(1) & -1(1) + 3(1) + 1(0) \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 4 & 2 \end{bmatrix}$$

continúa

### Primera Matriz

14	65	63	21	60
23	50	17	63	34
57	17	91	31	14
94	50	36	36	13
12	9	85	91	7

### Segunda Matriz

96	88	75	36	16
25	11	81	67	6
76	58	49	33	24
27	40	31	34	50
98	0	39	50	24

### Tercera Matriz

14204	6441	12393	10652	4616
9783	6080	9887	8581	5042
15022	11721	11618	7948	5084
15256	12350	14487	9796	4780
10980	9725	8888	7284	7004

**El código está distribuido en estos  
archivos:**

**main.cpp**

**Matrices.h**

**Matrices.cpp**

## Matrices.h

```
#ifndef EJEMPLO_2_MULTIPLICACION_DE_MATRICES_MATRICES_H
#define EJEMPLO_2_MULTIPLICACION_DE_MATRICES_MATRICES_H

#include <iostream>
#include <cstdint>
#include <stdlib.h>
#include <iomanip>

typedef int Tipo;
constexpr size_t nFILAS=5, nCOLUMNAS=5;

void GenerarMatriz(Tipo M[][nCOLUMNAS], size_t nfilas, size_t ncol);
void ImprimirMatriz(Tipo M[][nCOLUMNAS], size_t nfilas, size_t ncol);
void MultiplicarMatrices(Tipo M1[][nCOLUMNAS], size_t nfilasM1, size_t ncolM1,
                        Tipo M2[][nCOLUMNAS], size_t nfilasM2, size_t ncolM2,
                        Tipo M3[][nCOLUMNAS], size_t &nfilasM3, size_t &ncolM3);

#endif //EJEMPLO_2_MULTIPLICACION_DE_MATRICES_MATRICES_H
```

## main.cpp

```
#include <iostream>
#include <ctime>
#include "Matrices.h"
using namespace std;

int main()
{
    Tipo M1[nFILAS][nCOLUMNAS], M2[nFILAS][nCOLUMNAS];
    Tipo M3[nFILAS][nCOLUMNAS];
    size_t nfilasM1, ncolM1, nfilasM2, ncolM2, nfilasM3, ncolM3;

    nfilasM1 = nFILAS;
    ncolM1 = nCOLUMNAS;
    nfilasM2 = nFILAS;
    ncolM2 = nCOLUMNAS;

    srand(time(nullptr));
    GenerarMatriz(M1,nfilasM1, ncolM1);
    GenerarMatriz(M2,nfilasM2, ncolM2);
    cout<<"\n";
    cout <<"Primera Matriz \n\n";
    ImprimirMatriz(M1,nfilasM1, ncolM1);
    cout<<"\n";
    cout <<"Segunda Matriz \n\n";
    ImprimirMatriz(M2,nfilasM2, ncolM2);
```

```
    if( ncolM1!=nfilasM2)
        cout <<"Imposible multiplicar las matrices ";
    else
    {
        MultiplicarMatrices(M1,nfilasM1, ncolM1,
                             M2,nfilasM2, ncolM2,
                             M3,nfilasM3, ncolM3);

        cout <<"\n";
        cout <<"Tercera Matriz\n\n";
        ImprimirMatriz(M3,nfilasM3, ncolM3);
    }
    return 0;
}
```

```
#include "Matrices.h"
using namespace std;
```

```
void GenerarMatriz(Tipo M[][nCOLUMNAS], size_t nfilas, size_t ncol)
{
    //-----
    for(size_t contFilas=0; contFilas<nfilas; contFilas++)
        for(size_t contColumnas=0; contColumnas<ncol; contColumnas++)
            M[contFilas][contColumnas] = rand()%100;
}
```

```
void ImprimirMatriz(Tipo M[][nCOLUMNAS], size_t nfilas, size_t ncol)
{
    //-----
    for(size_t contFilas=0; contFilas<nfilas; contFilas++)
    {
        for (size_t contColumnas = 0; contColumnas < ncol; contColumnas++)
            cout << setw(7) << M[contFilas][contColumnas];
        cout << "\n";
    }
}
```

```

void MultiplicarMatrices(Tipo M1[][nCOLUMNAS], size_t nfilasM1, size_t ncolM1,
                        Tipo M2[][nCOLUMNAS], size_t nfilasM2, size_t ncolM2,
                        Tipo M3[][nCOLUMNAS], size_t &nfilasM3, size_t &ncolM3)
{
    //-----
    // M3 = M1 * M2
    for (int f = 0; f < nfilasM1; ++f)
    {
        for (int c = 0; c < ncolM2; ++c)
        {
            M3[f][c] = 0;
            for (int k = 0; k < ncolM1; k++){
                M3[f][c] += M1[f][k]*M2[k][c];
            }
        }
    }
    nfilasM3=nfilasM1;
    ncolM3 = ncolM2;
}

```

$$\begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1(3) + 0(2) + 2(1) & 1(1) + 0(1) + 2(0) \\ -1(3) + 3(2) + 1(1) & -1(1) + 3(1) + 1(0) \end{bmatrix} = \begin{bmatrix} 5 & 1 \\ 4 & 2 \end{bmatrix}$$

### Ejemplo 3:

Librerías Crisol, tiene 7 tiendas en Lima y atiende de lunes a viernes. El gerente ha almacenado la cantidad de libros que se venden en cada una de sus tiendas, durante los 5 días que atiende. Para luego hacer análisis de sus ventas.

Los datos se almacenaron en un matriz, cuya vista lógica se muestra enseguida:

		Lunes	Martes	Miercoles	Jueves	Viernes
Número de Tienda		0	1	2	3	4
	0	43	45	50	12	55
	1	34	19	20	7	6
	2	15	33	21	8	20
	3	11	23	67	38	49
	4	45	32	109	30	27
	5	3	44	38	15	45
	6	78	56	77	14	24



**Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el día de la semana en que se vendió la menor cantidad de libros.**

**Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de tienda que vendió más cantidad de libros.**

**El código está distribuido en estos  
archivos:**

**main.cpp**

**UFunciones.h**

**UFunciones.cpp**

```
// Created by Maria Hilda Bermejo  
//
```

## UFunciones.h

```
#ifndef CRISOL_UFUNCIONES_H  
#define CRISOL_UFUNCIONES_H
```

```
#include <iostream>  
#include <cstdlib>  
#include <ctime>  
#include <iomanip>
```

```
using namespace std;
```

```
typedef unsigned int TipoEntero;  
const unsigned int MAXFILAS =7;  
const unsigned int MAXCOL =5;
```

```
void  LlenaMatriz(TipoEntero Libros[][MAXCOL], size_t filas, size_t col);  
void  ImprimirMatriz(TipoEntero Libros[][MAXCOL], size_t filas, size_t col);  
string DiaQueVendioMenos(TipoEntero Libros[][MAXCOL], size_t filas, size_t col);  
size_t NumDeTiendaQueVendioMas(TipoEntero Libros[][MAXCOL], size_t filas, size_t col);
```

```
#endif //CRISOL_UFUNCIONES_H
```

## Main.cpp

```
#include <iostream>
#include "UFunciones.h"
```

```
using namespace std;
```

```
int main()
{
    TipoEntero Libros[MAXFILAS][MAXCOL]={0};

    srand(time(nullptr));
    LlenaMatriz(Libros, MAXFILAS, MAXCOL);
    ImprimirMatriz(Libros, MAXFILAS, MAXCOL);
    cout << "\nSe vendió menos libros el día : " << DiaQueVendioMenos(Libros, MAXFILAS, MAXCOL);
    cout << "\nEl número de la tienda que vendió mas libros es : " <<
    NumDeTiendaQueVendioMas(Libros, MAXFILAS, MAXCOL);
    return 0;
}
```

## UFunciones.cpp

```
#include "UFunciones.h"
```

```
void LlenaMatriz(TipoEntero Libros[][MAXCOL], size_t filas, size_t col)
```

```
//-----
```

```
{  
    for(size_t f=0; f<filas; f++)  
        for(size_t c=0; c<col; c++)  
            Libros[f][c] = rand() %1000;  
}
```

```
void ImprimirMatriz(TipoEntero Libros[][MAXCOL], size_t filas, size_t col)
```

```
//-----
```

```
{  
    cout << "\n\n";  
    for(size_t f=0; f<filas; f++) {  
        for (size_t c = 0; c < col; c++)  
            cout << setw(7) << Libros[f][c];  
        cout << "\n";  
    }  
}
```

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el día de la semana en que se vendió la menor cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes
Número de Tienda		0	1	2	3	4
	0	43	45	50	12	55
	1	34	19	20	7	6
	2	15	33	21	8	20
	3	11	23	67	38	49
	4	45	32	109	30	27
	5	3	44	38	15	45
	6	78	56	77	14	24

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el día de la semana en que se vendió la menor cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes
Número de Tienda		0	1	2	3	4
	0	43	45	50	12	55
	1	34	19	20	7	6
	2	15	33	21	8	20
	3	11	23	67	38	49
	4	45	32	109	30	27
	5	3	44	38	15	45
	6	78	56	77	14	24

229

252

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el día de la semana en que se vendió la menor cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes
Número de Tienda		0	1	2	3	4
	0	43	45	50	12	55
	1	34	19	20	7	6
	2	15	33	21	8	20
	3	11	23	67	38	49
	4	45	32	109	30	27
	5	3	44	38	15	45
	6	78	56	77	14	24

229	252	382
-----	-----	-----



Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el día de la semana en que se vendió la menor cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes
Número de Tienda		0	1	2	3	4
	0	43	45	50	12	55
	1	34	19	20	7	6
	2	15	33	21	8	20
	3	11	23	67	38	49
	4	45	32	109	30	27
	5	3	44	38	15	45
	6	78	56	77	14	24

229	252	382	124
-----	-----	-----	-----

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el día de la semana en que se vendió la menor cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes	
Número de Tienda		0	1	2	3	4	
	0	43	45	50	12	55	
	1	34	19	20	7	6	
	2	15	33	21	8	20	
	3	11	23	67	38	49	
	4	45	32	109	30	27	
	5	3	44	38	15	45	
	6	78	56	77	14	24	
		229	252	382	124	226	

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el día de la semana en que se vendió la menor cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes
Número de Tienda		0	1	2	3	4
	0	43	45	50	12	55
	1	34	19	20	7	6
	2	15	33	21	8	20
	3	11	23	67	38	49
	4	45	32	109	30	27
	5	3	44	38	15	45
	6	78	56	77	14	24

229

252

382

124

226

3

## UFunciones.cpp

```
string DiaQueVendioMenos(TipoEntero Libros[][MAXCOL], size_t filas, size_t col)
{
    //-----
    TipoEntero A[MAXCOL]={0};
    string Dias[5]= {"Lunes", "Martes", "Miercoles", "Jueves", "Viernes"};

    for (size_t c = 0; c < col; c++)
    {
        TipoEntero Suma=0;
        for(size_t f=0; f<filas; f++)
            Suma+= Libros[f][c];
        A[c]=Suma;
    }
    //--- hallamos la posicion del menor en array A
    TipoEntero menor = A[0];
    size_t posdelMenor=0;
    for(size_t i=1; i<MAXCOL;i++)
        if( A[i]<menor) {
            menor = A[i];
            posdelMenor = i;
        }
    return Dias[posdelMenor];
}
```

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de la tienda que vendió más cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes
Número de Tienda		0	1	2	3	4
	0	43	45	50	12	55
	1	34	19	20	7	6
	2	15	33	21	8	20
	3	11	23	67	38	49
	4	45	32	109	30	27
	5	3	44	38	15	45
	6	78	56	77	14	24

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de tienda que vendió más cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes
Número de Tienda		0	1	2	3	4
	0	43	45	50	12	55
	1	34	19	20	7	6
	2	15	33	21	8	20
	3	11	23	67	38	49
	4	45	32	109	30	27
	5	3	44	38	15	45
	6	78	56	77	14	24

205

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de tienda que vendió más cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes	
Número de Tienda		0	1	2	3	4	
	0	43	45	50	12	55	205
	1	34	19	20	7	6	86
	2	15	33	21	8	20	
	3	11	23	67	38	49	
	4	45	32	109	30	27	
	5	3	44	38	15	45	
	6	78	56	77	14	24	

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de tienda que vendió más cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes	
Número de Tienda		0	1	2	3	4	
	0	43	45	50	12	55	205
	1	34	19	20	7	6	86
	2	15	33	21	8	20	97
	3	11	23	67	38	49	
	4	45	32	109	30	27	
	5	3	44	38	15	45	
	6	78	56	77	14	24	



Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de tienda que vendió más cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes	
Número de Tienda		0	1	2	3	4	
	0	43	45	50	12	55	205
	1	34	19	20	7	6	86
	2	15	33	21	8	20	97
	3	11	23	67	38	49	188
	4	45	32	109	30	27	
	5	3	44	38	15	45	
	6	78	56	77	14	24	

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de tienda que vendió más cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes	
Número de Tienda		0	1	2	3	4	
	0	43	45	50	12	55	205
	1	34	19	20	7	6	86
	2	15	33	21	8	20	97
	3	11	23	67	38	49	188
	4	45	32	109	30	27	243
	5	3	44	38	15	45	
	6	78	56	77	14	24	

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de tienda que vendió más cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes	
Número de Tienda		0	1	2	3	4	
	0	43	45	50	12	55	205
	1	34	19	20	7	6	86
	2	15	33	21	8	20	97
	3	11	23	67	38	49	188
	4	45	32	109	30	27	243
	5	3	44	38	15	45	145
	6	78	56	77	14	24	

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de tienda que vendió más cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes	
Número de Tienda		0	1	2	3	4	
	0	43	45	50	12	55	205
	1	34	19	20	7	6	86
	2	15	33	21	8	20	97
	3	11	23	67	38	49	188
	4	45	32	109	30	27	243
	5	3	44	38	15	45	145
	6	78	56	77	14	24	249

Considerando las ventas de toda la semana y en todas las tiendas, se pide identificar cuál es el número de tienda que vendió más cantidad de libros.

		Lunes	Martes	Miercoles	Jueves	Viernes	
Número de Tienda		0	1	2	3	4	
	0	43	45	50	12	55	205
	1	34	19	20	7	6	86
	2	15	33	21	8	20	97
	3	11	23	67	38	49	188
	4	45	32	109	30	27	243
	5	3	44	38	15	45	145
	6	78	56	77	14	24	249

6

## UFunciones.cpp

```
size_t NumDeTiendaQueVendioMas(TipoEntero Libros[][MAXCOL], size_t filas, size_t col)
//-----
{
    TipoEntero A[MAXFILAS]={0};

    for(size_t f=0; f<filas; f++) {
        TipoEntero Suma = 0;
        for (size_t c = 0; c < col; c++)
            Suma += Libros[f][c];
        A[f] = Suma;
    }

    size_t posdelMayor=0;
    TipoEntero mayor = A[0];
    for(size_t i=1; i<MAXFILAS; i++)
        if( A[i]>mayor)
        {
            mayor = A[i];
            posdelMayor =i;
        }
    return posdelMayor;
}
```

## **Ejercicio 1:**

**Un fabricante de aspiradoras, desea construir una aspiradora que opere automáticamente y que sea capaz de detectar cuantos kilos de basura no podrá limpiar, porque el ambiente en donde se encuentra la basura está rodeada de paredes.**

**Para poder simular el ambiente, se ha optado por:**

**Una matriz cuyo tamaño es de 34 x 23.**

**En donde la cantidad de basura se ha representado por números enteros desde el 0 al 8.**

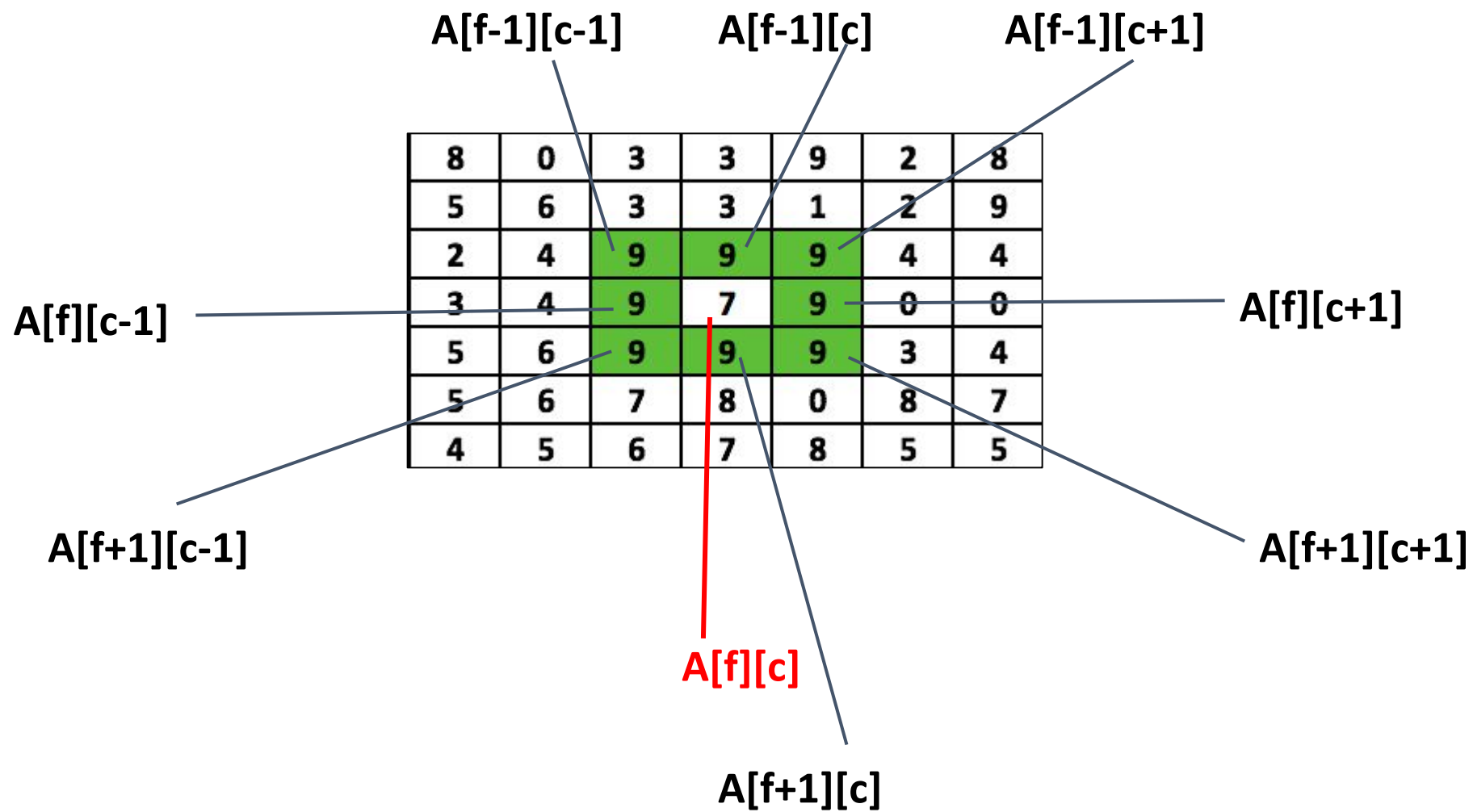
**Se ha reservado el número 9 para representar a una pared.**

**Elabore un programa que permita generar la matriz con valores de manera aleatoria y luego permita calcular la cantidad de kilos de basura que nunca podrán ser limpiados.**



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
0	1	2	3	4	5	6	7	8	9	0	1	1	3	4	4	5	6	1	1	1	3	1	7	
1	3	4	5	6	7	6	7	8	9	0	2	2	1	4	9	9	9	2	2	0	5	7	0	
2	2	9	9	9	2	1	2	1	4	3	3	3	2	3	9	7	9	3	3	0	7	7	2	
3	4	9	4	9	3	4	9	9	9	3	4	3	3	3	9	9	9	4	4	1	9	9	9	
4	5	9	9	9	4	5	9	3	9	3	5	3	4	3	3	3	3	5	5	2	9	2	9	
5	4	5	6	7	8	3	9	9	9	3	6	2	5	9	9	9	3	6	6	3	9	9	9	
6	3	4	5	6	7	8	8	9	8	3	7	3	6	9	4	9	3	7	2	3	3	4	6	
7	9	1	2	3	4	5	6	7	8	9	8	4	7	9	9	9	3	8	9	9	9	2	1	
8	7	2	4	5	6	7	8	0	3	3	9	2	8	2	3	4	2	9	9	1	9	3	9	
9	6	2	9	9	9	4	5	6	3	3	1	2	9	4	5	6	2	3	9	9	9	4	7	
10	5	2	9	4	9	1	2	4	9	9	9	4	4	5	6	7	8	5	7	7	6	5	6	
11	4	2	9	9	9	2	3	4	9	7	9	0	0	2	9	9	9	7	6	5	6	9	5	
12	3	1	1	2	2	4	5	6	9	9	9	3	4	7	9	9	9	3	3	9	9	9	4	
13	6	1	2	3	3	4	5	6	7	8	0	8	7	0	9	9	9	3	4	9	0	9	3	
14	1	1	1	2	1	2	4	5	6	7	8	5	5	2	3	5	6	6	3	9	9	9	6	
15	5	5	9	9	9	4	5	6	5	7	8	5	1	2	3	4	5	6	7	8	9	1	8	
16	5	5	9	8	9	2	4	6	6	8	1	2	3	4	5	6	7	8	9	3	5	2	1	
17	6	0	9	9	9	2	2	9	9	9	0	4	5	2	4	5	6	7	8	5	6	3	0	
18	4	8	6	7	8	2	1	9	5	9	0	9	9	9	6	7	4	5	9	9	9	4	2	
19	9	0	0	0	1	1	1	9	9	9	1	9	0	9	7	4	5	5	9	3	9	5	3	
20	2	3	4	2	2	2	1	2	2	3	2	9	9	9	7	4	4	5	9	9	9	6	4	
21	1	2	4	4	4	4	5	6	7	8	5	6	6	7	4	5	6	7	8	2	3	6	5	
22	4	5	6	7	8	5	5	5	6	4	5	6	7	8	5	4	4	5	6	7	8	6	7	
23	9	9	9	9	9	3	4	5	6	7	8	2	2	3	2	5	3	3	3	3	3	3	4	
24	3	4	9	4	9	4	5	6	9	9	9	3	4	5	9	9	9	7	4	5	6	7	8	
25	2	3	9	9	9	6	6	7	9	2	9	3	3	4	9	0	9	6	5	4	6	1	9	
26	2	3	4	5	6	7	7	8	9	9	9	3	5	3	9	9	9	5	6	4	5	2	7	
27	1	2	3	4	5	6	7	8	9	2	3	3	4	5	5	5	5	6	5	6	9	3	6	
28	5	5	4	5	6	7	8	5	1	2	3	4	5	6	7	8	9	5	9	9	9	4	5	
29	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	9	2	9	5	4	
30	5	5	5	4	6	9	9	9	5	5	6	7	8	3	4	5	6	1	9	9	9	6	3	
31	9	9	9	5	6	9	2	9	4	2	4	2	3	9	9	9	1	3	4	2	2	4	6	
32	9	3	9	5	6	9	9	9	3	2	2	3	4	9	9	9	1	2	3	4	5	5	7	
33	9	9	9	3	4	5	8	7	3	3	4	5	6	9	9	9	1	2	3	3	4	5	6	





**El Código de los programas lo pueden ubicar en este link:**

**<http://bit.ly/2H8i5Yo>**

## Unidad 4: Array de array - Matrices

<http://bit.ly/2HRBWgq>

Profesores:

Ernesto Cuadros- Vargas, PhD.

[ecuadros@utec.edu.pe](mailto:ecuadros@utec.edu.pe)

María Hilda Bermejo, M. Sc.

[mbermejo@utec.edu.pe](mailto:mbermejo@utec.edu.pe)