

## Unidad 5: Arrays de arrays

**http:**

**Profesores:**

**Juan Flores Moroco**

# Telegram:

**1. Configurar tu cuenta**

**2. Link: <http://bit.ly/2OW5Ss9>**

# SISAP:



**SISAP 2018**  
October 7-9  
Lima, Perú

**Evento: SISAP 2018 – 11th International Conference on Similarity Search and Applications**

**Fechas: October 7-9 Lima, Perú**

**Resumen: <http://www.sisap.org/2018/>**

The 11th International Conference on Similarity Search and Applications (SISAP) is an annual forum for researchers and application developers in the area of similarity data management. It aims at the technological problems shared by numerous application domains, such as data mining, information retrieval, multimedia, computer vision, pattern recognition, computational biology, geography, biometrics, machine learning, and many others that make use of similarity search as a necessary supporting service.

**Inscripciones: [https://eventos.spc.org.pe/spire2018/registration\\_sisap.html](https://eventos.spc.org.pe/spire2018/registration_sisap.html)**

# SPIRE:

**SPIRE 2018: 25th International Symposium on String Processing and Information Retrieval**

**Fechas: October 9-11 Lima, Perú**

**Resumen: <https://eventos.spc.org.pe/spire2018/venue.html>**

**SPIRE 2018 is the 25th edition of the annual Symposium on String Processing and Information Retrieval. SPIRE has its origins in the South American Workshop on String Processing, which was first held in Belo Horizonte, Brazil, in 1993. Since 1998 the focus of the workshop has also included information retrieval, due to its increasing relevance to and inter-relationship with string processing.**

**SPIRE 2018 will be held in UTEC Lima, Peru.**

**Inscripciones: <https://eventos.spc.org.pe/spire2018/registration.html>**

# Logro de la sesión:

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

# Arrays de arrays (Matrices)

---

**A continuación se muestra el enunciado y el código del programa que se mostro en la clase del Auditorio.**

**Una vez que termine de revisar este código, cuyo archivos fuente se le proporciona, Ud. va a adicionar código para implementar las funciones nuevas que se le solicitarán mas adelante.**

## 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 mas 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.**

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

## Main.cpp

```
1  #include <iostream>
2  #include <ctime>    //Se incluye <ctime> para poder usar la función srand()
3  #include "Arrays.h"
4
5  using namespace std;
6
7
8  int main()
9  {
10     int M[nFILAS][nCOLUMNAS];
11
12     srand(time(nullptr));
13     cout<<"\n";
14     LlenarArray(M,nFILAS, nCOLUMNAS);
15     ImprimirArray(M,nFILAS, nCOLUMNAS);
16     cout<<"\n";
17     cout<<"El menor elemento almacenado en el array es " << ElMenor(M,nFILAS,nCOLUMNAS);
18     cout<<"\n";
19     cout<<"La suma de la diagonal es igual a " << SumadeDiagonal(M,nFILAS,nCOLUMNAS);
20     cout<<"\n";
21     cout<<"Suma de los elementos por encima de la diagonal : " << SumaPorEncimadeLaDiagonal(M,nFILAS,nCOLUMNAS);
22     cout<<"\n";
23     return 0;
24 }
```

## Array.h


```
1 //
2 // Created by Maria Hilda Bermejo on 9/17/18.
3 //
4
5 #ifndef EJEMPLO1_RECORRIDOS_ARRAYS_H
6 #define EJEMPLO1_RECORRIDOS_ARRAYS_H
7
8 #include <iostream>
9 #include <cstdint> //-- para usar size_t
10
11 using namespace std;
12
13 constexpr size_t nFILAS = 7, nCOLUMNAS = 7;
14
15 void LlenarArray(int M[][nCOLUMNAS], size_t filas, size_t columnas);
16 void ImprimirArray(int M[][nCOLUMNAS], size_t filas, size_t columnas);
17 int ElMenor(int M[][nCOLUMNAS], size_t filas, size_t columnas);
18 int SumadeDiagonal(int M[][nCOLUMNAS], size_t filas, size_t columnas);
19 int SumaPorEncimadeLaDiagonal(int M[][nCOLUMNAS], size_t filas, size_t columnas);
20
21 #endif //EJEMPLO1_RECORRIDOS_ARRAYS_H
22
```

## Array.cpp

```
1  //
2  // Created by Maria Hilda Bermejo on 9/17/18.
3  //
4
5  #include <cstdlib>
6  #include <iomanip>
7  #include "Arrays.h"
8
9
10 void LlenarArray(int M[][nCOLUMNAS], size_t filas, size_t columnas)
11 { //-----
12     for(size_t f=0; f<filas; f++)
13         for(size_t c=0; c<columnas; c++)
14             M[f][c] = rand()%100;
15 }
16
17 void ImprimirArray(int M[][nCOLUMNAS], size_t filas, size_t columnas)
18 { //-----
19     for(size_t f=0; f<filas; f++)
20     {
21         for (size_t c = 0; c < columnas; c++)
22             cout << setw(5) << M[f][c];
23         cout<<"\n";
24     }
25 }
26
```

```
27  ➡ int ElMenor(int M[][nCOLUMNAS], size_t filas, size_t columnas)
28  { //-----
29      int Menor;
30
31      Menor=M[0][0];
32      for(size_t f=0; f<filas; f++)
33          for (size_t c = 0; c < columnas; c++)
34              if(M[f][c]<Menor)
35                  Menor =M[f][c] ;
36      return Menor;
37  }
38
39
40  ➡ int SumaDiagonal(int M[][nCOLUMNAS], size_t filas, size_t columnas)
41  { //-----
42      int Suma=0;
43
44      for(size_t f=0; f<filas; f++)
45          Suma+=M[f][f];
46      return Suma;
47  }
48
```

Continúa .... Main.cpp

```
48
49  int SumaPorEncimadelaDiagonal(int M[][nCOLUMNAS], size_t filas, size_t columnas)
50 { //-----
51     int Suma=0;
52
53     for(size_t f=0; f<filas-1; f++)
54         for (size_t c = f+1; c < columnas; c++)
55             Suma+=M[f][c];
56     return Suma;
57 }
```





**El Código del proyecto se lo proporcionará su profesor.**

**Se pide adicionar funciones que permitan realizar lo siguiente:**

- 1. Imprimir la suma de los elementos de cada una de las filas.**
- 2. Imprimir el promedio de cualquiera de las columnas.**
- 3. Intercambiar el contenido de los datos de dos columnas se indiquen.**
- 4. Hallar el dato mayor de entre todos los elementos de la diagonal.**

**Cada profesor puede armar su Hacker Rank.**

**En la primera sesión, se sugiere utilizar programas como: el primero y el tercero de la imagen. Los pueden encontrar en el usuario compartido con los nombres que aparecen en la imagen.**

UTEC-POO1-MHBR-Matriz fila x fila de la última a la primera(array de arrays)+	matriz-2	mbermejo1
UTEC-MHxEC-CS1102- Intercalar Arrays++(OK)	intercalar-vectores	enriquecuri
UTEC-POO1-MHRB- Posición de dato en la matriz(arrays de arrays)+	poo-posicion-en-matriz	mbermejo1