# Arreglos en C/C++

Mag. Ing. Nancy López

# Arreglos (arrays) unidimensionales

- **Estructuras estáticas** que sirven para representar vectores.
- Se declaran de un tamaño y lo conservan a los largo de todo el programa.
- Se almacenan en la memoria de forma contigua.
- Almacenan **datos del mismo tipo**.
- Cada dato se puede referenciar individualmente mediante la utilización de un índice.

# Arreglos unidimensionales

**Declaración:**

- TipoDeDato  nombre [tamaño];
- Ejemplos:
  - float sueldos[100];
  - int edades[50];
  - char letras[22];

# Arreglos unidimensionales

- int edades [10];
- Es un vector llamado edades que va a almacenar 10 datos de tipo entero.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Valores** | **1** | **30** | **15** | **60** | **12** | **34** | **56** | **6** | **24** | **13** |
| **Índice** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |

- Edad en el elemento 1: edad[0].
- Para mostrarla: cout<<edad[0].

# Arreglos unidimensionales

**Inicializar un vector:**

- Con un solo valor:
  - int edades[10]={0};
- Con valores distintos
  - int edades[10]={1,30,15,60,12,34,56,6,24,13};

- Si no se inicializa, toma lo que haya en la memoria en ese momento.

# Arreglos unidimensionales

**Cargar un vector**

- Se puede cargar en forma aleatoria:
  - edades[5]=20;

- O secuencial:

```
for (i=0; i<10; i++)
  {
     cout<<“Ingrese una edad ”;
     cin>>edades[i];
}
```

# Arreglos unidimensionales

**Mostrar un vector:**

- Se puede mostrar un elemento:
  - cout<<edades[5];

- O todos:

```
for (i=0; i<10; i++)
  {
     cout<<edades[i]<<endl;
}
```

# Arreglos unidimensionales

**Pasaje de arreglos a funciones**

- Se considera siempre que es ***por referencia***, o sea que la función puede modificar los valores del vector.
- En este caso, no devuelve ningún valor.
- El siguiente es un ejemplo de funciones con arreglos:

```
#include…
float promEdad(int[]);
int main()
{int edades[10], i;
 for (i=0; i<10; i++)
  {cout<<"Ingrese edad: ";
   cin>>edades[i];
  }
  cout<<"El promedio de las edades ingresadas es: "<<promEdad(edades);
return 0;
 }

float promEdad(int edades[10])
{
 int j=0;
 float prom=0;
 for(j=0; j<10; j++)
   { prom=prom+edades[j];
   }
 return (prom/10);
}
```

# Arreglos Multidimensionales

- En este caso los arreglos (estáticos) tienen más de una dimensión, en este curso usaremos los arreglos bidimensionales, pero se puede utilizar más de dos dimensiones.

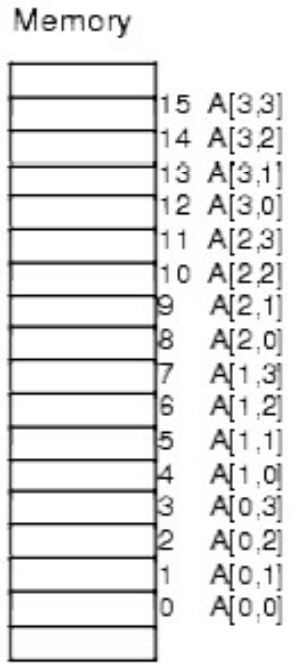
# Arreglos Multidimensionales

- Arreglo bidimensional

int arreglo[4][4]

Representación gráfica de un arreglo de **dos dimensiones**

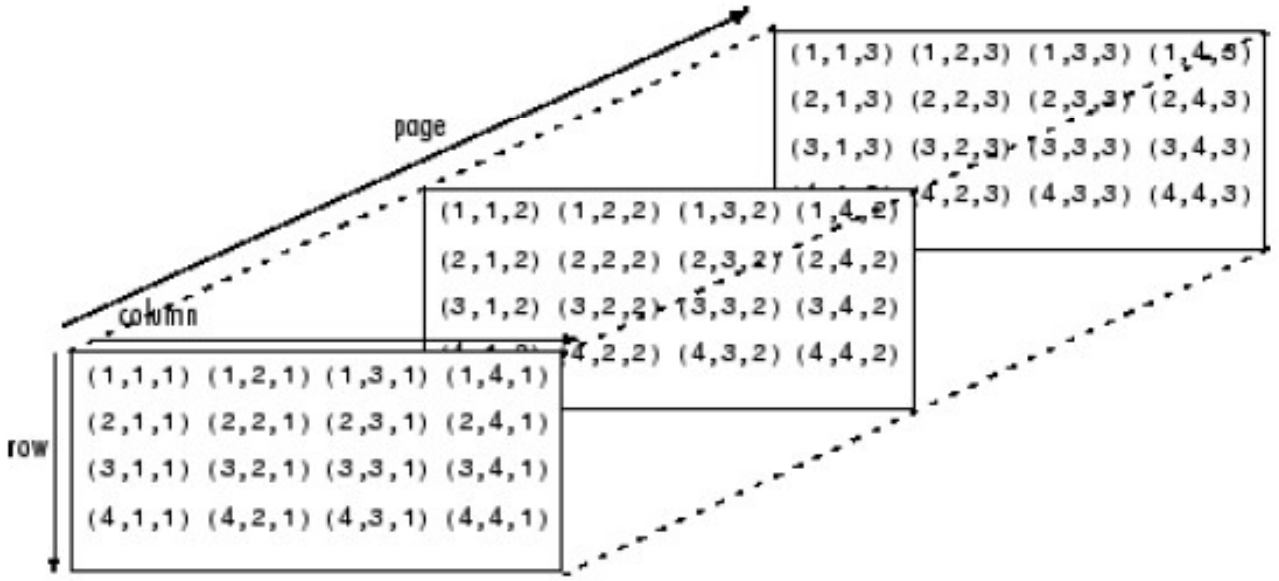
| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 |
| 1 | 4 | 5 | 6 | 7 |
| 2 | 8 | 9 | 10 | 11 |
| 3 | 12 | 13 | 14 | 15 |



# Arreglos Multidimensionales

- Arreglo tridimensional

int arreglo[4][4][3]
Representación gráfica de un arreglo de **tres dimensiones**



# Arreglos Multidimensionales

- Es una estructura de datos estática de un mismo tipo de datos y de longitud fija que almacena datos de forma matricial.
- Al igual que los arreglos unidimensionales, el almacenamiento de los datos en la memoria se realiza de forma secuencial y son accedidos mediante índices.
- Los arreglos multidimensionales son también conocidos como **matrices**.
- Se llama matriz de **orden** “m*n” a un conjunto rectangular de elementos dispuestos en filas “m” y en columnas “n”, siendo m y n números naturales.

# Arreglos Multidimensionales

- Ejemplo de matriz de 3x4

| | | ***columnas*** | | | |
|---|---|---|---|---|---|
| | | *c0* | *c1* | *c2* | *c3* |
| ***filas*** | *f0* | *m[f0,c0]* | *m[f0,c1]* | *m[f0,c2]* | *m[f0,c3]* |
| | *f1* | *m[f1,c0]* | *m[f1,c1]* | *m[f1,c2]* | *m[f1,c3]* |
| | *f2* | *m[f2,c0]* | *m[f2,c1]* | *m[f2,c2]* | *m[f2,c3]* |

# Arreglos Multidimensionales

**Declaración**

- tipoDato nombre [dim1][dim2]…[dimn]

- Ejemplo: matriz de enteros con 7 filas y 9 columnas.

- int datos[7][9];

# Arreglos Multidimensionales

Llenado de un arreglo

- Para llenar un arreglo, debemos recorrer tanto las filas como las columnas.
- Para eso utilizamos dos ciclos, uno dentro del otro.
- El ejemplo anterior sería:

```
#include…
int main()
{
 int datos[7][9], i, j;
 for (i=0; i<7; i++)            //recorre las filas de la matriz
 {
   for(j=0; j<9; j++)           //recorre las columnas de la matriz
   {
     cout<<"ingrese dato para la posición [" << i <<"]["<<j<<"]  ";
     cin>>datos[i][j];
   }
 }
return 0;
}
```

- Y para mostrar la matriz, la volvemos a recorrer:

```
for (i=0; i<7; i++)  //recorre las filas de la matriz
 {
   for(j=0; j<9; j++)  //recorre las columnas de la matriz
    {
      cout<<datos[i][j]<< "\t"; //muestra y hace una tabulación
    }                                          // entre columnas
   cout<<endl;  //hace un enter para pasar a la siguiente fila
 }
```

# Arreglos Multidimensionales

- En los ejemplos vistos estamos llenando y mostrando la matriz **por filas**, o sea que dejamos “fija” una fila (el primer for) y recorremos todas las columnas de esa fila (el segundo for) y luego pasamos a la segunda fila y repetimos el proceso.
- También se podría haber llenado y mostrado “por columna” según convenga a nuestros intereses.

# Arreglos Multidimensionales

- Vamos a ver un ejemplo más práctico:
- Se desea leer las edades y los sueldos de los 5 empleados de una empresa.
- En este caso necesitaremos 5 filas (una para cada empleado) y 2 columnas (una para la edad y otra para el sueldo).
- También podría hacerse al revés (2 filas y 5 columnas)
- Usaremos constantes para las filas y columnas.

```
#include…
#define FIL=5, COL=2;
void main()
{
 int datos[FIL][COL], i, j;
 for (i=0; i<FIL; i++)
 {  /*como son sólo 2 datos y son distintos, usamos un solo for y
    ponemos los índices de las columnas manualmente*/
    cout<<"ingrese edad del empleado "<< i+1 <<" ";
    cin>>datos[i][0];
    cout<<"Ingrese sueldo del empleado"<< i+1 <<" ";
    cin>>datos[i][1];
 }
 cout<<endl<<endl<<endl;
```

```
cout<<"No.        Edad     Sueldo"<<endl;
 for (i=0; i<FIL; i++)
 {
    cout<<i+1<<"\t";
    for(j=0; j<COL; j++)
     {
       cout<<datos[i][j]<< "\t";
     }
    cout<<endl;
 }

  return 0;
}
```

# Arreglos Multidimensionales

- Otros ejemplos:
- Sumar todos los elementos de una matriz:

```
for (i=0; i<FIL; i++)
 {
   for(j=0; j<COL; j++)
   {
      cout<<"ingrese valor en la pos.["<< (i)<<"]["<<j <<"] ";
      cin>>datos[i][j];
      suma=suma+datos[i][j];
   }
 }
 cout<<"La suma de todos los valores de la matriz es: <<suma;
```

# Arreglos Multidimensionales

- Sumar sólo las filas. En este caso, para evitar tener que definir tantas variables como filas tengo, defino un vector del mismo tamaño que las filas de la matriz.

- Lo mismo vale para la suma por columnas.

- Por filas:

```
int datos[FIL][COL], i, j, suma[FIL]={0};
 for (i=0; i<FIL; i++)
 {
   for(j=0; j<COL; j++)
   {
      cout<<"ingrese valor en la pos.["<< (i)<<"]["<<j <<"] ";
      cin>>datos[i][j];
      suma[i]=suma[i]+datos[i][j];
   }
 }
 cout<<"La suma de todos los valores de las filas es: "<<endl;
 for (i=0; i<FIL; i++)
    cout<<suma[i]<<"\t";
```

# Arreglos Multidimensionales

- Búsqueda en una matriz.
- Para encontrar un valor en una matriz, tenemos que **recorrerla** hasta encontrar el valor buscado.
- Tenemos que tener una variable que me indique si lo encontré o no:

```
int datos[FIL][COL], i, j, busc, encontrado=0;

//llenado de la matriz

cout<<"Ingrese valor a buscar ";
 cin>>busc;
 for (i=0; i<FIL; i++)
 {
   for(j=0; j<COL; j++)
   {
      if(datos[i][j]==busc)
        encontrado=1;
   }
 }
 //fuera del for
 if(encontrado==1)
   cout<<"Valor encontrado";
 else
   cout<<"Valor no encontrado";
```

# Arreglos Multidimensionales

- Si me interesa buscar en un lugar en particular:

```
cout>>”Ingrese fila”;
cin>>f;
cout<<“Ingrese columna”;
cin>>c;
cout<<“Ingrese valor a buscar”;
cin>>busc;
if(datos[f-1][c-1]==busc)
    cout<<“Valor encontrado”;
else
    cout<<“valor no encontrado”;
```

# Arreglos Multidimensionales

- Dos matrices se pueden sumar o restar.
- Para ello, ambas matrices tienen que ser de igual orden (misma cantidad de filas y columnas).
- Las operaciones se hacen elemento a elemento.
- Ej.
- matrizC[i][j]=matrizA[i][j]+matrizB[i][j];

# Arreglos Multidimensionales

- También se pueden multiplicar o dividir por un escalar.
- Ejemplo, hallar el duplo de los elementos de la matriz:

- matrizC[i][j]= matrizA[i][j] * 2;

# Arreglos Multidimensionales

- Producto de matrices:
- Sólo es posible si el número de columnas de la matriz izquierda es igual al número de filas de la matriz de la derecha:

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

# Arreglos Multidimensionales

- **Matrices cuadradas**

- Los índices de la diagonal principal son iguales.

- Los índices de la diagonal secundaria suman la cantidad de filas o columnas menos uno.