Unidad 2 - Arreglos

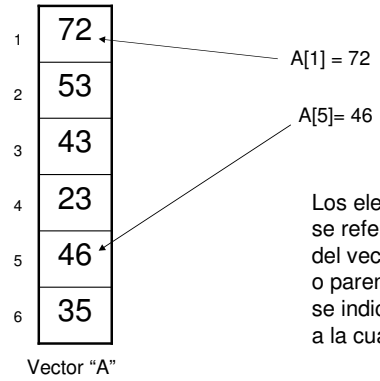
Un arreglo es un conjunto finito de posiciones de memoria consecutivas que tienen el mismo nombre y el mismo tipo de dato.

Los arreglos de una dimensión (Unidimensionales) se llaman vectores.

Los arreglos de dos dimensiones (bidimensionales) se llaman matrices.

1

Arreglos Unidimensionales (Vectores)

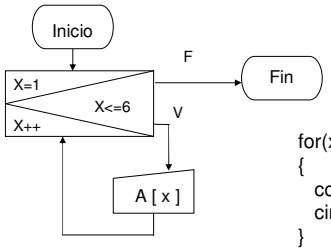


Los elementos de un vector se referencían por el nombre del vector seguido de corchetes o parentesis dentro de los cuales se indica la posición (indice) a la cual hacen referencia.

2

Arreglos Unidimensionales (vectores)

- Para recorrer, capturar o procesar secuencialmente todos los elementos de un vector, se puede utilizar un ciclo sencillo:




```
for(x=1; x<=6; x++)
{
   cout << "Dame el elemento" << x ;
   cin >> A[x];
}
```

3

Vectores en C++

int A [7];

"A" es un vector de 7 elementos que C++ reconoce de la posicion 0 a la 6.

Se puede "ignorar" la posición 0 y trabajar de la 1 a la 6.

Int A [7] = {0,72,53,43,23,46,35};

En este caso se inicializa el valor de los elementos al momento de declarar el vector. También puede hacerse desde el programa:

A[1] = 72;
A[2] = 53;
….etc.

4

Vectores de estructuras

- De la misma manera como se pueden realizar vectores de otros tipos de datos (entero, flotante, carácter, etc.), también son válidos los vectores de estructuras (también llamados arrays de registros).
- Los Vectores de estructuras tienen las mismas características de los vectores y las estructuras simples.

5

EJEMPLO 1
Los elementos del vector de estructuras se nombran de la siguiente manera:
NombreDeEstructura[indice].NombreDelCampo

Vector de Estructuras "e"

	Num (Capturar)	Sxh (Capturar)	Ht (Capturar)	Sb (Calcular)	Desc (Calcular)	Sn (Calcular)
0						
1						
2						
3						

e[1].Num
Numero de Identificación del empleado #1

e[3].Ht
Horas trabajadas para empleado #3

e[2].Desc
Descuento para empleado # 2

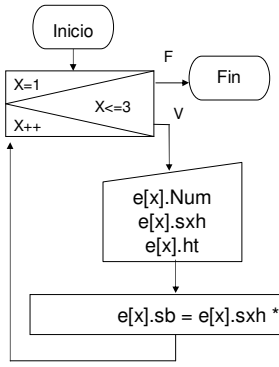
6

Declaración del vector de estructuras en C++ para Ejemplo 1

```
struct estructura
{
   int num;
   float sxh;
   float ht;
   float sb;
   float desc;
   float sn;
};                          //Declaracion del Tipo de la estructura


struct estructura e[4];     //Declaracion del Vector de
                            Estructuras llamado "e"
                            de 3 posiciones
```

7

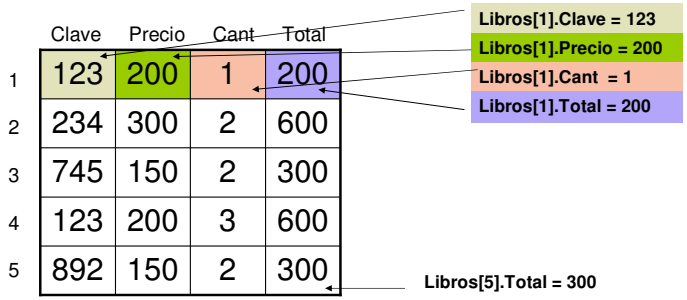
Diagrama de flujo y referencia en el programa para Ejemplo 1




```
for (x=1; x<=3; x++)
{
    cout<< "Numero de Empleado: ";
    cin >> e[x].num;

    cout<< "Sueldo por hora ";
    cin >> e[x].sxh;

    cout <<"Horas trabajadas";
    cin >> e[x].ht;

    e[x].sb = e[x].sxh  *  e[x].ht;
}
```

8

Vectores de estructuras (Ejemplo 2)

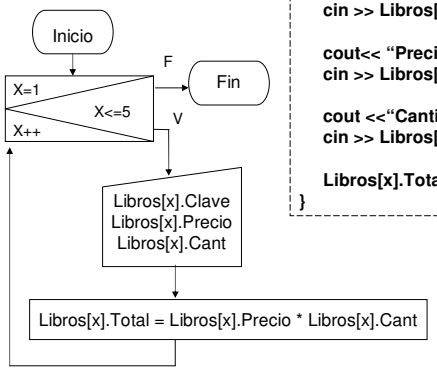


Vector de Estructuras "Libros"

Declaración del vector de estructuras en C++ para ejemplo 2

```
//Declaración del tipo "Estructura":
struct Estructura
{
  int clave;
  float precio;
  int cant;
  float total;
};

//Declaración del Vector de Estructuras de tipo "Estructura":
Struct Estuctura Libros[ 6 ];
```

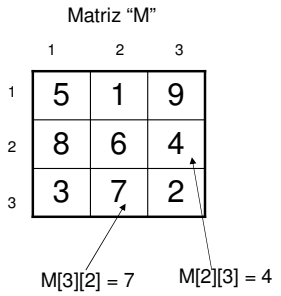

Diagrama de Flujo y Codificación para Ejemplo 2




```
for (x=1; x<=5; x++)
{
    cout<< "Clave: ";
    cin >> Libros[x].Clave;

    cout<< "Precio: ";
    cin >> Libros[x].Precio;

    cout <<"Cantidad:";
    cin >> Libros[x].Cant;

    Libros[x].Total = Libros[x].Precio * Libros[x].Cant;
}
```

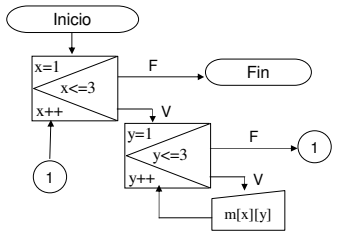

Arreglos Bidimensionales (Matrices)

- Una matriz tiene renglones y columnas.
- Sus elementos se referencían por el nombre de la matriz seguido de corchetes o parentesis indicando renglon y columna en donde se encuentra ese elemento.



Arreglos Bidimensionales (Matrices)

- Para recorrer, capturar o procesar secuencialmente todos los elementos de una matriz, se pueden utilizar dos ciclos anidados, uno para los renglones y otro para las columnas:




```
for(x=1; x<=3; x++)
{
  for (y=1; y<=3; y++)
  {
    cout << "Dame el elemento"<< x << y;
    cin >> m[x][y];
  }
}
```


Matrices en C++

int M [4] [4];

"M" es una matriz de 4 renglones y 4 columnas que C++ reconoce de la posicion 0 a la 3.

Se puede "ignorar" la posición 0 y trabajar de la 1 a la 3.

```
int M [ 4 ] [ 4 ] = {  0,0,0,0,
                       0,5,1,9,
                       0,8,6,4,
                       0,3,7,2  };
```

En este caso se inicializa el valor de los elementos al momento de declarar la matriz. También puede hacerse desde el programa:

```
A[1][1] = 5;
A[1][2] = 1;
  ….etc.
```