

Arreglos: Vectores y Matrices

$$A = \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \dots \\ \vec{v}_n \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 0 & 3 \\ 2 & 1 & -1 \\ 3 & 1 & 2 \end{pmatrix}$$

SESION 4



Arreglos

Los arreglos (arrays) son variables del mismo tipo de dato que tienen el mismo nombre y que se distinguen y referencían por un índice.

Sintaxis:

<tipo> <variable> [N]

Se declara un arreglo de nombre <variable> con N elementos de tipo <tipo>, (N es una constante).

Ejemplo: int a[10];



Características



Los arreglos se caracterizan por:

· Almacenan los elementos en posiciones contiguas de memoria

Tienen un mismo nombre de variable que representa a todos los elementos. Para hacer referencia a esos elementos es necesario utilizar un índice que especifica el lugar que ocupa cada elemento dentro del archivo.

Mónica E. García



Ejemplos:



float numero[10];	Arreglo de 10 números reales
int k[104];	Arreglo de 104 enteros
char ch[300];	Cadena de 299 caracteres
int m[23][3]	Matriz de 2 dimensiones, con 23 renglones y 3
	columnas.



Clasificación

Arreglos

Tipos de

Unidimensionales (Vectores)

•Bidimensionales (Matrices)

•De 3 dimensiones(Tridimensionales)

•De mas de 3 dimensiones (Multidimensionales)

Mónica E. García



Vectores

Es un arreglo de "N" elementos organizados en una dimensión donde "N" recibe el nombre de longitud o tamaño del vector.

Para hacer referencia a un elemento del vector se usa el nombre del mismo, seguido del índice (entre corchetes), el cual indica una posición en particular del vector.

Por ejemplo: V[x]

Donde:

V Nombre del arreglo

x Número de datos que constituyen el arreglo(tamaño)



Representación gráfica



V[0]	7
V[1]	8
V[2]	9
V[3]	10

Nota sobre los arreglos

El valor del índice debe de estar entre 0 y un LIMITE entero

Mónica E. García



Llenado de un vector

#include<stdio.h>
#include<conio.h>

```
void main()
{
  int renglon,vector[5];
  renglon=0;

do
  {
    printf("Dar valor: ");
    scanf("%d",&vector[renglon]);
  renglon++;
}while(renglon<5);</pre>
```



Salida de un vector

```
renglon=0;
  do
  {
    printf(" \nElemento: ",vector[renglon]);
    renglon++;
  }while(renglon<5);
  getch();
}</pre>
```

Mónica E. García



Asignaciones

```
int i, vector[5];
i=3;
vector[i]= 1;
vector[0]=8;
vector[5]=2;

i=vector[i];
vector[2] = vector[0]+vector[3];
vector[i+2]=20;
vector[vector[i]]=0;
```

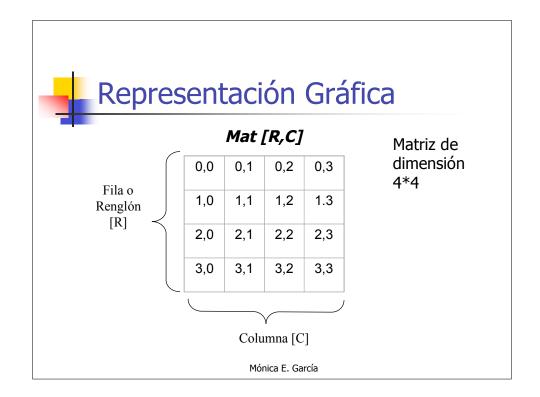


Es un arreglo de M * N elementos organizados en dos dimensiones donde ${\bf M}$ es el numero de filas o reglones y ${\bf N}$ el numero de columnas.

Para representar una matriz se necesita un nombre de matriz se necesita un nombre de matriz acompañado de dos índices.

Mat [R,C]

Donde R indica el renglón y C indica la columna, donde se encuentra almacenado el dato.





Llenado de una matriz



Salida de una matriz

```
printf("A continuación se muesta el contenido del vector\n");
printf("\t\t**************POR RENGLONES ******\n\n");
for (renglon=0;renglon<5;renglon++)
{
    for (columna=0;columna<3;columna++)
    {
        printf(" \nElemento : = %d",M[renglon][columna]);
    }
}
getch();
}</pre>
Mónica E. García
```



Asignaciones

```
int r,c,k, matriz[2][3];
k=3100;
j=2;
matriz[1][0]= 2;
matriz[0][c]=k;
matriz[r][c]=50;
matriz[c][k]=12;

k=matriz[r][c];
r=matriz[1][2];
matriz[1][r] = 5 *matriz[0][r];
c=matriz[5][c-1];
k=matriz[vector[r]][0];

Mónica E. García
```



Arreglos y cadena de caracteres

Una cadena de caracteres se representa por un arreglo unidimensional de caracteres.

Cada carácter de la cadena se almacena en un elemento del arreglo. En general el último elemento de la cadena es el carácter nulo ('\0').



Representación gráfica

Mat[renglon][columna]

Н	0	L	Α	
С	0	М	0	L
Н	0	Y		
L	Е	0		

Mat[0]
Mat[1]
Mat[2]
Mat[3]

vector de cadenas

Mónica E. García



Ejemplo: