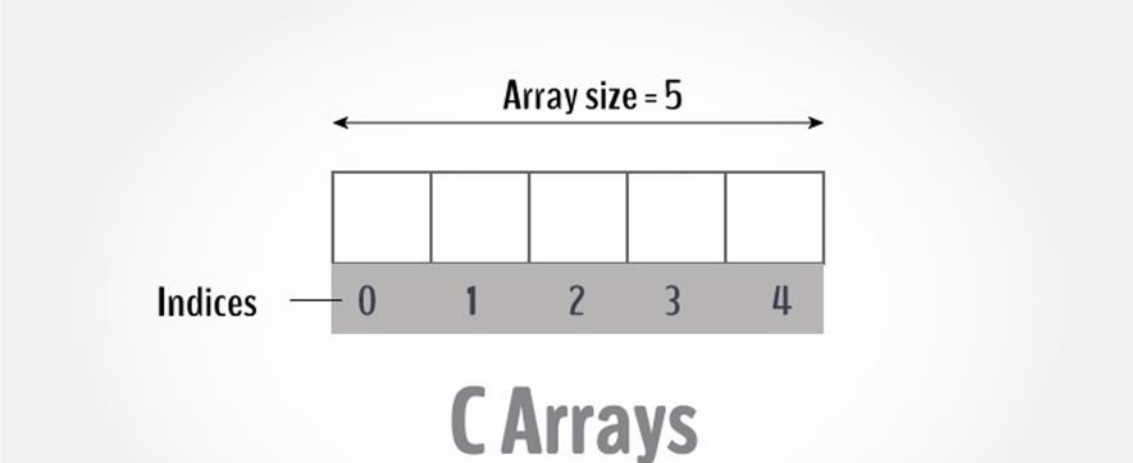
El lenguaje de programación C
- Arrays-

Array

Un array en C o C ++ es una colección de elementos almacenados en ubicaciones de memoria contiguas y se puede acceder a los elementos de forma aleatoria utilizando índices de un array . Se utilizan para almacenar tipos de elementos similares, ya que el tipo de datos debe ser el mismo para todos los elementos



Por qué usarlos?

Podemos usar variables normales (v1, v2, v3, ..) cuando tenemos una pequeña cantidad de objetos, pero si queremos almacenar una gran cantidad de instancias, se vuelve difícil administrarlas con variables normales. La idea de un array es representar muchas instancias en una variable.

Cómo declararlos

```
tipo_de_dato nombre_del_vector[tamanio];
```

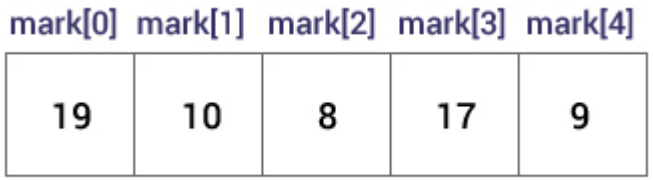
```
int my_vector1[10];
float my_vector2[25];
string my_vector3[500];
bool my_vector4[1000];
char my_vector5[2];
```

Declaración e inicialización

```
char vector[5] = {"5", "h", "2", "8", "a"};

int vector2[] = {1,2,3,4,10,9,80,70,19};

float vector3[5] = {10.5};

int vector2[3]; vector2[3] = {1,5,10};

int vector2[3]; vector2[0] = 1; vector2[1] = 3;
vector2[2] = 10;
```

Acceso a los Datos de un Array

```
int mark[5] = {19, 10, 8, 17, 9}
```




```
mark[2] = 7;//Modificar la posición 2
mark[4] = 0;//Modificar la posición 4
scanf("%d", &mark[2]); // Cargar desde el teclado
printf("%d", mark[0]); // Imprimir valores
```

Recorrido de Arrays

```
int main() {
  int valores[5];

  printf(“Ingrese 5 enteros: ");

  // carga de valores
  for(int i = 0; i < 5; ++i) {
    scanf("%d", &values[i]);
  }


  // Muestra de valores
  for(int i = 0; i < 5; ++i) {
    printf("%d\n", values[i]);
  }
  return 0;
}
```

Recorrido de Arrays(2)

```
int main() {
  int valores[5];

  printf(“Ingrese 5 enteros: ");

  // carga de valores
 int i=0;
  while(i < 5; ) {
     scanf("%d", &values[i]);
    ++i;
  }


  // Muestra de valores
  while(i < 5; ) {
     printf("%d\n", values[i]);
     ++i;
  }
  return 0;
}
```

Recorrido de Arrays(3)

Cúando usar FOR y cúando WHILE ?

En general para recorrer un array completamente se usa FOR, y para una recorrida incomplete del array se usa WHILE

Recorrido de Arrays(2)

```
int main()
{
    int marks[10], i, n, suma = 0, promedio;

    printf(“Ingrese la Cantidad de Números: ");
    scanf("%d", &n);

    for(i=0; i<n; ++i)
    {
        printf(“Ingrese un Número: ",i+1);
        scanf("%d", &marks[i]);

        // suma
        suma += marks[i];
    }

    promedio = suma/n;
    printf(“Promedio = %d", promedio);

    return 0;
}
```

Arrays con Funciones

```
void leo arreglo(int a[],int n)
{
      for (int i=0;i < n;i++){ printf(“Ingrese elemento :”);
                  scanf (“ %d”,&a[i]);
                   printf(“\n”); }
}

leo arreglo(array2,TAMANIO)

void imprimo arreglo(int a[],int n)
 {
   for (int i=0;i
    { printf(“Elemento numero %d = %d”, i+1, a[i]);
                  printf(“\n”); }

}
```

Arrays con Funciones (2)

Cuando queremos pasar un único valor del array se indica el índice.

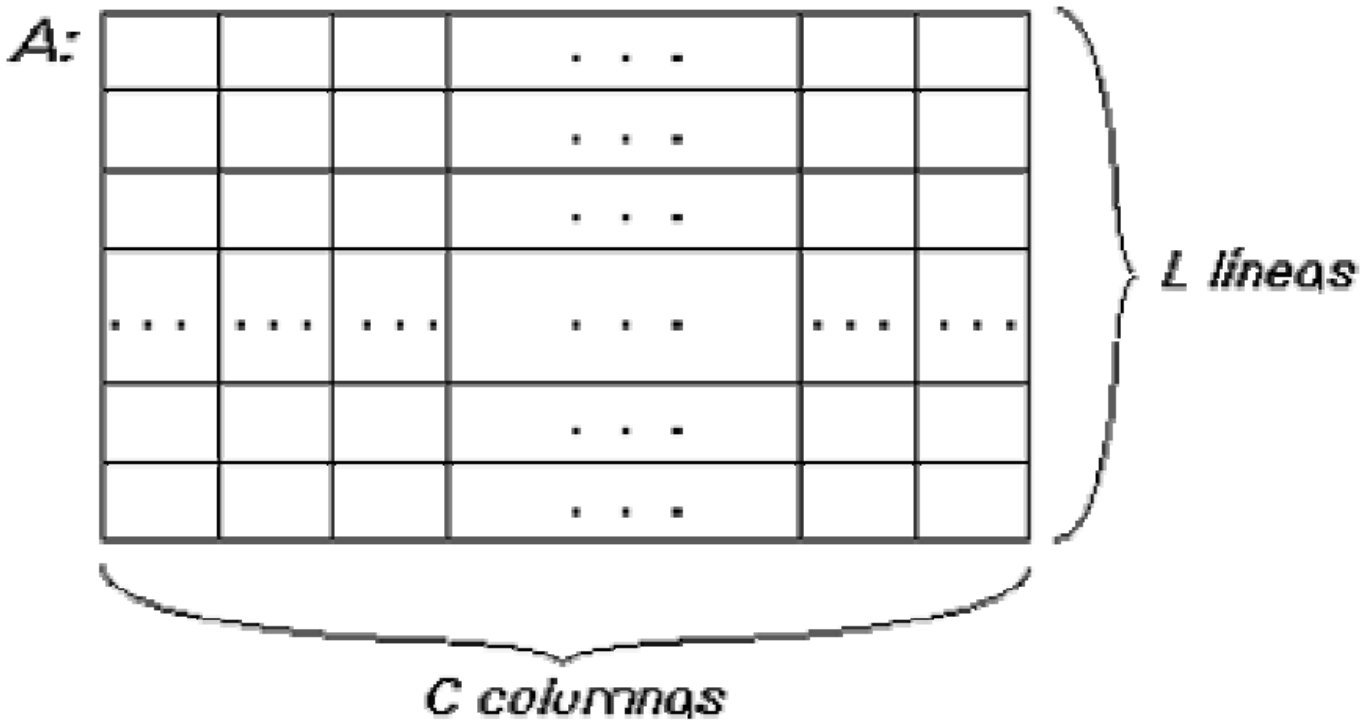
```
void leo arreglo(int a,int n)
{
                ……….
}
```

leo arreglo(array2[5],TAMANIO)

Array de 2 dimensiones (Matriz)

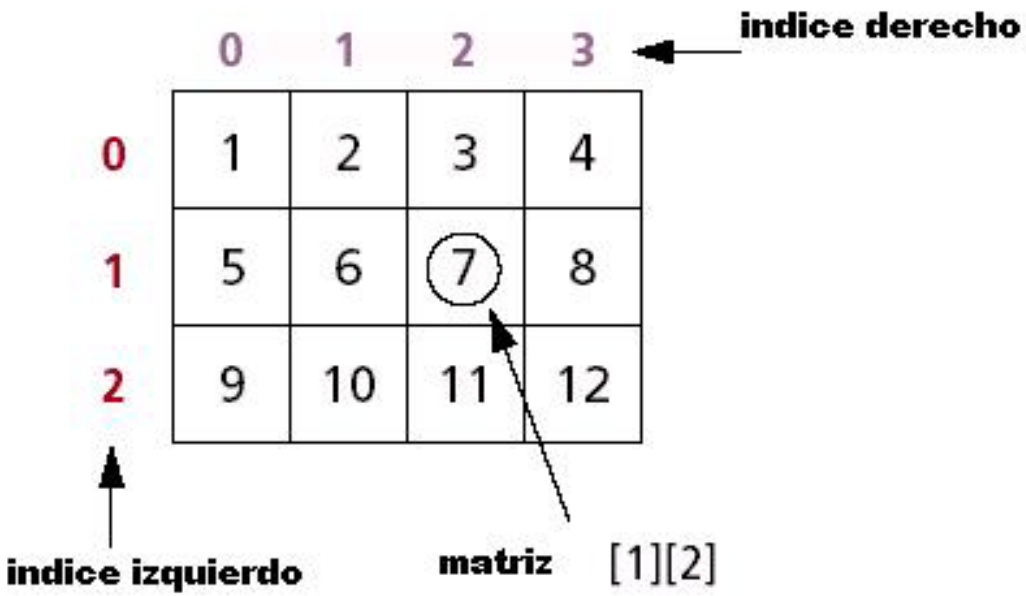
Un array en C puede tener una, dos o más dimensiones. Por ejmplo, un array de dos dimensiones también denominado matriz, es interpretado como un array (unidimensional) de dimensión "f" (número de filas), donde cada componente es un array (unidimensional) de dimensión "c" (número de columnas). Un array de dos dimensiones, contiene, pues, "f*c" componentes.

Array de 2 dimensiones (Matriz)



Array de 2 dimensiones (Matriz)

int matriz[3][4];



Declaración e Inicialización

```
char lista[5][5];
int estudiantes[3][25];
float ingresos[10][5];
int matriz[4][3];
//iniclialización
int ejemploArryaBidimensional [2][3] = {1, 2, 3, 7, 8, 9};
int ejemploArryaBidimensional [2][3] = { {1,2,3},{7,8,9} };
```

Acceso a los Datos

```
tabla[2][3] = 4.5;
estudiante[1][15] = 10;

ventas = año[1][27];
dia = semana[3][6];
notas = estudiante[1][15];
```

Recorrrido

```
int main(){
int fila=3,col=4,matriz[fila][col];

for(fila=0;fila<3;fila++)
        for(col=0;col<4;col++)
                matriz[fila][col]=fila*col;
return 0;
}
```

Recorrrido

```
int main () {

  /* array con 5 filas y 2 columnas*/
  int a[5][2] = { {0,0}, {1,2}, {2,4}, {3,6},{4,8}};
  int i, j;

  /* Impresion del contenido del array*/
  for ( i = 0; i < 5; i++ ) {

    for ( j = 0; j < 2; j++ ) {
      printf("a[%d][%d] = %d\n", i,j, a[i][j] );
    }
  }

  return 0;
}
```

Array Multidimensionales

C permite arreglos con mas de dos Dimensiones

La forma general de una declaración de arreglo es
tipo nombre_var[**tamaño1**] [**tamaño2**]… [**tamañoN**]