

# PRÁCTICA

## ARREGLOS UNIDIMENSIONALES

---

### Objetivos

*El alumno conocerá y aplicará el concepto de arreglos unidimensionales para resolver problemas que requieren algoritmos de tipo numérico.*

*Al final de esta práctica el alumno podrá:*

1. Manejar arreglos unidimensionales.
2. Realizar exitosamente programas que hagan uso de arreglos.

### Antecedentes

1. Conocer y manejar ciclos de repetición.
2. Conocer la utilización de la sentencia if-else.

### Introducción

Un *arreglo* es una secuencia de datos del mismo tipo que ocupan un lugar contiguo en memoria. Las posiciones consecutivas que ocupa el arreglo se denominan *elementos* del arreglo y se numeran sucesivamente 0, 1, 2, 3, etc. El tipo de información que se almacena en un arreglo puede ser cualquiera de los tipos de dato básicos de C, es decir, char, int, float o double. Así por ejemplo, un arreglo puede contener, la edad de los alumnos de una clase, las temperaturas de cada día del mes en una ciudad determinada, o el número de personas que residen en cada una de las delegaciones del Distrito Federal.

El arreglo contiene una serie de valores del mismo tipo dispuestos en memoria de forma consecutiva y a los que se accede a través de un índice que representa la posición relativa de los datos en la memoria. En un arreglo de tamaño  $N$ , La primer posición corresponde al índice 0 y la última al índice  $N-1$ . Si el identificador del arreglo es  $b$ , entonces  $b[0]$  es el nombre del elemento que está en la posición 0,  $b[1]$  es el nombre del elemento que está en la posición 1, etc. En general, el elemento  $i$ -ésimo está en la posición  $i-1$ . De modo que si el arreglo  $b$  tiene  $N$  elementos, sus nombres son  $b[0]$ ,  $b[1]$ , ...,  $b[N-1]$ .

### Declaración de un arreglo

Al igual que con las variables, se debe declarar el arreglo antes de utilizarlo. Un arreglo se declara de modo similar a otros tipos de datos, excepto que se debe indicar al compilador el tamaño o longitud – número de elementos - del arreglo. Para indicar al compilador la longitud del arreglo, se debe hacer seguir al nombre, el tamaño encerrado entre corchetes. La sintaxis para declarar un arreglo de una dimensión es:

tipo nombre\_arreglo[número\_de\_elementos];

---

Elaborada por:

Ing. Alejandra Vargas E. De los M. [alejandrav@correo.unam.mx](mailto:alejandrav@correo.unam.mx)

M.C. Vicente Fuentes Gea [vfg@servidor.unam.mx](mailto:vfg@servidor.unam.mx)

M.C. Cintia Quezada Reyes [cintia@fi-b.unam.mx](mailto:cintia@fi-b.unam.mx)

## PRÁCTICA ARREGLOS UNIDIMENSIONALES

---

Por ejemplo, para crear un arreglo de diez elementos enteros, se escribe:

```
int num[10];
```

Esta declaración hace que el compilador reserve espacio suficiente para contener diez valores enteros.

Las definiciones de arreglos pueden incluir, si se desea, la asignación de valores iniciales. Los valores iniciales deben aparecer en el orden en que serán asignados a los elementos individuales del arreglo, encerrados entre llaves y separados por comas. La forma general es:

```
tipo nombre_arreglo[número_de_elementos] = {valor1,valor2,...,valorN};
```

La presencia del número de elementos del arreglo, es opcional cuando los valores iniciales están presentes. El método para inicializar arreglos mediante valores constantes después de su definición, es adecuado cuando el número de elementos es pequeño. Por ejemplo, para inicializar un arreglo de diez enteros con los valores 5, 1, 3, 8, 9, 3, 0, 1, 5, 7, se escribe:

```
int num[] = {5, 1, 3, 8, 9, 3, 0, 1, 5, 7};
```

### Procesamiento de los datos de un arreglo

Una vez que un arreglo ha sido declarado se puede acceder a sus elementos colocando nuevamente corchetes después del identificador del arreglo y encerrando una expresión entera entre ellos. Por ejemplo para asignar 100 valores enteros a un arreglo de tamaño 100, se plantea así:

```
int i, a[100]; /*declaramos una variable índice y un arreglo de 100 elementos enteros*/
```

```
/* asignación de los 100 valores*/
```

```
for(i = 0; i < 100; i++)  
    a[i] = i + 1;
```

De la misma forma, para imprimir en la consola los números previamente almacenados en el arreglo, se procede como sigue:

```
for(i = 0; i < 100; i++)  
    printf("%d\n", a[i]);
```

## PRÁCTICA ARREGLOS UNIDIMENSIONALES

---

Es importante señalar que cualquier elemento del arreglo puede ser manipulado en la misma manera que una variable. Por ejemplo, se puede incrementar en una unidad el contenido del tercer elemento de tres formas diferentes:

```
i = 2;           /* accederemos al tercer elemento*/
a[i] = a[i] + 1; /* a[2] contiene el número 4*/
a[i] += 1;      /* a[2] contiene el número 5*/
a[i]++;        /* a[2] contiene el número 6*/
```

O llevar a cabo operaciones más complejas, como elevar al cubo el contenido del tercer elemento así:

```
a[i] *= a[i] * a[i]; /* a[2] contiene el número 216*/
```

así como signar al arreglo valores proporcionados desde el teclado, como se muestra a continuación:

```
printf("Dame diez números enteros, los que tu quieras") ;
for(i = 0; i < 10; i++)
{
    printf("Número %i:\n", i + 1) ;
    scanf("%i", &a[i]) ;
}
```

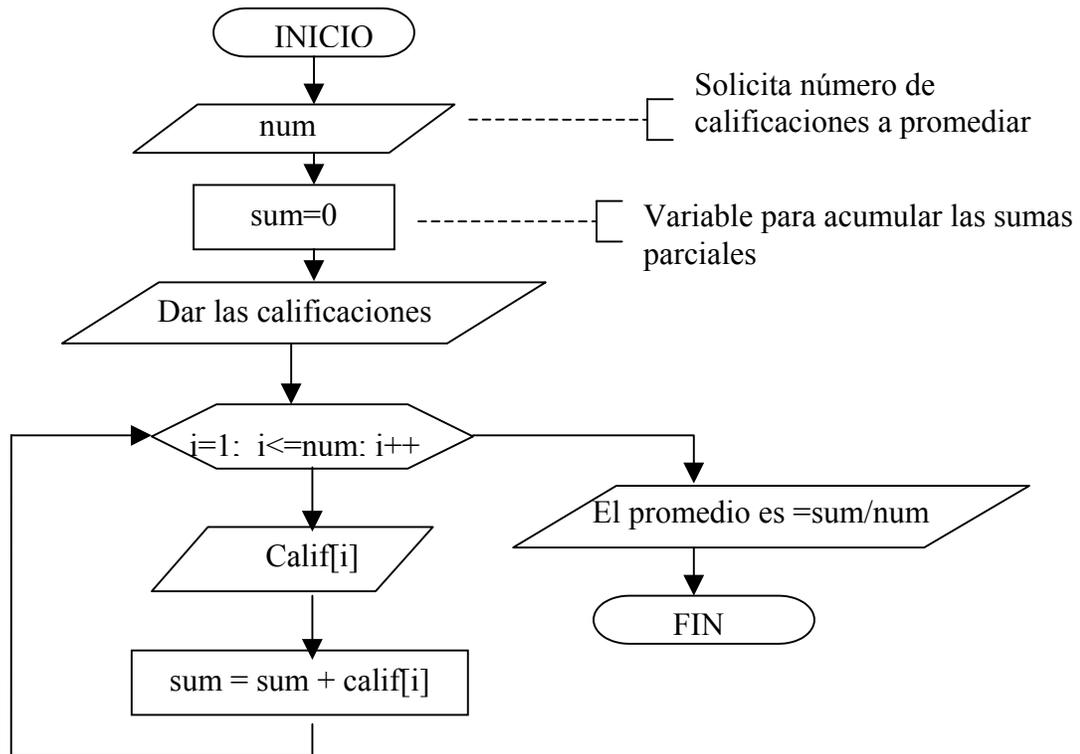
Conviene aclarar que la manipulación del arreglo debe realizarse elemento por elemento, como se mostró en los ejemplos anteriores, de modo que, tratándose de arreglos relativamente grandes, casi siempre es necesario emplear procesos cíclicos que utilizan *do-while*, *while* o *for*.

### Ejemplos

El siguiente algoritmo obtiene el promedio de una serie de calificaciones guardadas en un arreglo, utilizando la proposición *for*, además solicita al usuario el número de calificaciones a promediar.

**PRÁCTICA**  
**ARREGLOS UNIDIMENSIONALES**

---



/\* Ejemplo 1: Programa que obtiene el valor promedio de números contenidos en un arreglo \*/

```
#include <stdio.h>
#include <conio.h>
int main(void)
{
    int num, i;
    float x, calif[100], sum;

    printf("¿Cuántas calificaciones deseas introducir?: ");
    scanf("%d",&num);
    sum=0;
    for (i=0; i<num; i++)
    {
        printf("\nCalificación: ");
        scanf("%f",&calif[i]);
        sum=sum+calif[i];
    }
}
```

Elaborada por:

Ing. Alejandra Vargas E. De los M. [alejandrav@correo.unam.mx](mailto:alejandrav@correo.unam.mx)  
M.C. Vicente Fuentes Gea [vfg@servidor.unam.mx](mailto:vfg@servidor.unam.mx)  
M.C. Cintia Quezada Reyes [cintia@fi-b.unam.mx](mailto:cintia@fi-b.unam.mx)

**PRÁCTICA**  
**ARREGLOS UNIDIMENSIONALES**

---

```
    }

printf("El promedio es: %.2f",sum/num);
printf("\n Los datos que proporcionaste fueron: \n");

for (i=0;i<num;i++)
{
    printf("\n%d. %f",i+1,calif[i]);
}
return 0;
}

/* Ejemplo 2: Programa que almacena 5 números reales en un arreglo y los imprime*/

#include <stdio.h>
int main(void)
{
    float num[5];
    int i;

    for(i=0; i<5; i++)
    {
        printf("Introduce un número en la posición %d: ",i);
        scanf ("%f",&num[i]);
    }
    for(i=0; i<5; i++)
    {
        printf("\nEl número en la posición %d es %f",i,num[i]);
    }
    return 0 ;
}
```

/\*Ejemplo 3: Programa que pide n números, cada uno lo multiplica por 5 y el resultado lo guarda en otro arreglo \*/

```
#include <stdio.h>
int main(void)
{
    float ar1[20], ar2[20];
    int cont, n;

    printf("Dame un número entre 2 y 20: ");
    scanf("%d",&n);
    printf("A continuación se pedirán %d números",n);
```

---

Elaborada por:

Ing. Alejandra Vargas E. De los M. [alejandrav@correo.unam.mx](mailto:alejandrav@correo.unam.mx)  
M.C. Vicente Fuentes Gea [vfg@servidor.unam.mx](mailto:vfg@servidor.unam.mx)  
M.C. Cintia Quezada Reyes [cintia@fi-b.unam.mx](mailto:cintia@fi-b.unam.mx)

**PRÁCTICA**  
**ARREGLOS UNIDIMENSIONALES**

---

```
for(cont=0; cont<n; cont++)
{
    printf("Dame el valor del número %d: ",cont) ;
    scanf("%f",&ar1[cont]);
    ar2[cont]=ar1[cont]*5;
}
printf("Tras multiplicar cada número por cinco se tiene: ",n);
for(cont=0; cont<n; cont++)
{
    printf("Número %d es:%f",cont, ar2[cont]) ;
}
return 0 ;
}
```

**Errores comunes que se pueden cometer:**

1. Declarar un arreglo dándole dimensión mediante una variable.
2. Olvidar indicar el tamaño  $n$  del arreglo entre corchetes al momento de ser declarado cuando no se le asignen valores iniciales.
3. Olvidar que el primer elemento del arreglo tiene el subíndice cero.
4. Olvidar que el último elemento del arreglo tiene el subíndice  $n-1$ .

**Ejercicios propuestos**

**Nota: No olvide realizar el algoritmo *antes* de elaborar el programa**

1. Sean los vectores  $\vec{d} = (3,7,2)$  y  $\vec{e} = (10,4,11)$ , obtenga  $\vec{d} \bullet \vec{e}$ . (Donde  $\vec{a} = (a_1, a_2, a_3)$ ,  $\vec{b} = (b_1, b_2, b_3)$  y  $\vec{a} \bullet \vec{b} = (a_1b_1 + a_2b_2 + a_3b_3)$ )
2. Sean los vectores  $\vec{d} = (8,8,2)$  y  $\vec{e} = (10,15,17)$ , obtenga  $6\vec{d} \bullet 20\vec{e}$  y  $15\vec{e} \bullet 4\vec{d}$ . (Recuerde que  $2\vec{a} \bullet 5\vec{b} = (2a_1, 5b_1 + 2a_2, 5b_2 + 2a_3, 5b_3)$ ).
3. Determinar el módulo del siguiente vector:  $\vec{a} = (4,5)$ .  
(Use la fórmula:  $|\vec{a}| = \sqrt{a_1^2 + a_2^2}$ ).
4. Determinar el cuadrado del módulo del siguiente vector:  $\vec{k} = (9,5,2)$ .  
(Use la fórmula:  $|\vec{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2}$ ).
5. Determinar el del módulo del siguiente vector:  $\vec{a} = (9,15,6,7,13,4)$ .  
(Use la fórmula:  $|\vec{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2 + a_4^2 + a_5^2 + a_6^2}$ ).
6. Realice un programa que lea el arreglo  
14 7 3 45 78 12 56 90 123 567  
y lo imprima como:

---

Elaborada por:

Ing. Alejandra Vargas E. De los M. [alejandrav@correo.unam.mx](mailto:alejandrav@correo.unam.mx)  
M.C. Vicente Fuentes Gea [vfg@servidor.unam.mx](mailto:vfg@servidor.unam.mx)  
M.C. Cintia Quezada Reyes [cintia@fi-b.unam.mx](mailto:cintia@fi-b.unam.mx)

**PRÁCTICA**  
**ARREGLOS UNIDIMENSIONALES**

---

14  
7  
3  
45  
78  
12  
56  
90  
123  
567

7. Calcule la resistencia equivalente que pueda sustituir 20 resistencias conectadas en serie. (Use la fórmula:  $R_{eq} = R_1 + R_2 + \dots$ )
8. Escribir un programa que lea una lista de números reales, los cuente y a continuación imprima su varianza. (Haga uso de la fórmula:  $Varianza = \frac{1}{n} \sum_{i=1}^n (x_i - x)^2$ , donde  $x$  es el promedio de todos los números  $x_i$ ).
9. Calcular el promedio de los elementos que se encuentren en las posiciones pares de un arreglo de  $n$  números.
10. Utilice un vector con 10 elementos y devuelva la suma de los mismos.
11. Lea las matrículas y las calificaciones de los alumnos de un grupo almacénelas en dos arreglos unidimensionales. Después debe imprimir el contenido de ambos arreglos.
12. ¿Cuál es la resistencia equivalente en un circuito que tiene 5 resistencias en paralelo con los siguientes valores: 10, 5, 14, 3, 8?. (Use la fórmula:  $\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \dots$ )
13. Hacer un programa que solicite al usuario 20 números y los ordene de mayor a menor.
14. Realizar una variante del programa anterior ordenado de menor a mayor.
15. Programar un algoritmo que obtenga la moda de un cierto número de valores dados por el usuario. Moda es el número que aparece con más frecuencia.
16. Cargar un arreglo de  $m$  números enteros indicar cuál es el mayor y donde está.
17. Cargar un arreglo de  $n$  números y calcular la media de los que estén en posiciones pares.
18. Cargar un arreglo de  $n$  números y calcular la media de los que estén en posiciones impares.
19. Se pide realizar un programa que calcule el centro de masas de un sistema de  $n$  masas puntuales. Para ello el programa pedirá conjuntos de tres números reales: la coordenada

---

Elaborada por:

Ing. Alejandra Vargas E. De los M. [alejandrav@correo.unam.mx](mailto:alejandrav@correo.unam.mx)

M.C. Vicente Fuentes Gea [vfg@servidor.unam.mx](mailto:vfg@servidor.unam.mx)

M.C. Cintia Quezada Reyes [cintia@fi-b.unam.mx](mailto:cintia@fi-b.unam.mx)

**PRÁCTICA**  
**ARREGLOS UNIDIMENSIONALES**

---

$x$  del punto, la coordenada  $y$  y su masa  $m$ . El número  $n$  de masas será establecido por el usuario. El programa imprimirá por pantalla como resultado las coordenadas  $x_g$  y  $y_g$  del centro de masas del conjunto de puntos que se han introducido. Las fórmulas que dan la posición del centro de gravedad son las siguientes:

$$x_g = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i} \qquad y_g = \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i}$$

20. Genere un vector de tamaño 20 con números entre -10 y 10. Deje en un arreglo  $Z$  todos los números negativos y en un arreglo  $X$  todos los positivos o iguales a cero.
21. Genere un vector de tamaño 10 con números reales leídos desde teclado. Calcule el promedio e indique cuántos elementos del vector son mayores que el promedio y cuántos menores o iguales
22. Dado un vector con  $n$  elementos numéricos enteros ya almacenados, indique cuántos de ellos son múltiplos de 3.
23. Se tienen 3 arreglos  $A, B, C$  de  $M$  elementos. Se debe crear otro arreglo ( $X$ ) de tres elementos, donde cada elemento sea la suma de los elementos de cada arreglo.