

## Guía de práctica para arreglos unidimensionales

Claudio J. Paz  
claudiojpaz@gmail.com

1 de julio de 2019

### Ejemplos

#### Recorrer arreglo

Se recomienda siempre recorrer los arreglos con sentencias `for`, ya que evita posibles errores. Queda explícito el inicio del arreglo desde el cero, y el tamaño del mismo.

```
#include <stdio.h>

#define N 5

int main(void) {
    int i;
    int vec[N] = {2,4,6,8,10};

    for (i = 0; i < N; i++)
        printf("%d", vec[i]);

    return 0;
}
```

#### Buscar elementos en el arreglo

Se recorre el arreglo y cada elemento se compara con el elemento buscado. En el ejemplo se busca un 8.

```
#include <stdio.h>

#define N 5

int main(void) {
    int i;
    int vec[N] = {2,4,6,8,10};

    for (i = 0; i < N; i++)
        if (vec[i] == 8)
            printf("Encontrado en posición %d", i);

    return 0;
}
```

## Buscar máximo (o mínimo) de los elementos en el arreglo

Se recorre el arreglo y cada elemento se compara con el máximo (o el mínimo). En el ejemplo se busca la máxima nota, entonces la variable con la que se compara cada elemento se inicializa en la mínima nota posible, en este caso 1. Cuando encuentra un valor mayor al máximo, se actualiza este último.

```
#include <stdio.h>

#define N 5

int main(void) {
    int i;
    int vec[N] = {2,9,4,8,10};
    int max = 1;

    for (i = 0; i < N; i++)
        if (vec[i] > max)
            max = vec[i];

    printf("El valor máximo es %d\n", max);

    return 0;
}
```

## Cargar elementos en el arreglo, pero validando con algún criterio

Se recorre el arreglo y se usa un scanf para cargar los datos, pero con un do-while para hacer la validación. En el ejemplo cargan solo valores positivos.

```
#include <stdio.h>

#define N 5

int main(void) {
    int i;
    int vec[N];

    for (i = 0; i < N; i++) {
        do {
            printf("Ingrese un número positivo: ");
            scanf("%d", &vec[i]);
        } while (vec[i] < 0);
    }

    for (i = 0; i < N; i++)
        printf("%d", vec[i]);

    return 0;
}
```

## Ejercicio 1

Escribir un programa que pida al usuario que ingrese  $N$  valores de un arreglo y luego lo muestre uno al lado del otro separados por un espacio

```
#include <stdio.h>

#define N 5

int main(void) {
    int i;
    int vec[N];

    for (i = 0; i < N; i++) {
        printf("Ingrese el elemento %d: ", i);
        scanf("%d", &vec[i]);
    }

    for (i = 0; i < N; i++) {
        printf("%d ", vec[i]);
    }

    return 0;
}
```

## Ejercicio 2

Escribir un programa que pida al usuario que ingrese  $N$  valores de un arreglo y luego muestre cuantos elementos pares fueron ingresados.

## Ejercicio 3

Escribir un programa que pida al usuario que ingrese  $N$  valores de un arreglo y luego muestre los elementos pares. Ejemplo

```
Ingrese los 5 elementos del arreglo "a"...
Ingrese el elemento a[0]: 1
Ingrese el elemento a[1]: 2
Ingrese el elemento a[2]: 3
Ingrese el elemento a[3]: 4
Ingrese el elemento a[4]: 5
Los elementos pares del arreglo son:
a[1]: 2
a[3]: 4
```

## Ejercicio 4

Escribir un programa que pida al usuario que ingrese  $N$  valores de un arreglo y luego muestre el valor máximo. Ejemplo

```
Ingrese los 5 elementos del arreglo "a"...
Ingrese el elemento a[0]: 3
Ingrese el elemento a[1]: 2
Ingrese el elemento a[2]: 5
Ingrese el elemento a[3]: 4
Ingrese el elemento a[4]: 1
El elemento mayor es a[2] con un 5
```

### Ejercicio 5

Escribir un programa que pida al usuario que ingrese  $N$  valores de un arreglo y luego calcule el promedio de todos los números ingresados.

### Ejercicio 6

Escribir un programa que pida al usuario que ingrese  $N$  notas (deben estar validadas del 1 al 10) y luego calcule el promedio, la nota máxima y la mínima.

### Ejercicio 7

Escribir un programa que pida al usuario que ingrese  $N$  valores de un arreglo y luego invierta el orden de sus elementos, de modo que, luego del cambio, el primer elemento contenga ahora el valor que tenía el último, y viceversa. Ejemplo

```
Ingrese los 5 elementos del arreglo "a"...
Ingrese el elemento a[0]: 1
Ingrese el elemento a[1]: 2
Ingrese el elemento a[2]: 3
Ingrese el elemento a[3]: 4
Ingrese el elemento a[4]: 5
El arreglo con los elementos invertidos son:
a[0]: 5
a[1]: 4
a[2]: 3
a[3]: 2
a[4]: 1
```