

# Guia de Ejercicios con Arreglos

Cada ejercicio debe ser copiado/escrito, luego compilarlo, probarlo y “decorarlo”, puede haber errores totalmente voluntarios

## Ejercicios

1) Realizar un algoritmo que cargue un arreglo de 10 enteros y lo muestre por pantalla

```
#include <stdio.h>

int main()
{
    int a[10], i;
    for (i = 1; i <= 10; i++){
        printf("Ingrese un numero para la posición %d: ", i);
        scanf("%d", a[i]);
    }
    for (i = 0; i < 10; i++){
        printf("Posicion %d, esta el numero: %d\n", i, a[i]);
    }
    return 0;
}
```

2) Dado el algoritmo anterior realizar una función para la carga y otra función para mostrar el arreglo por pantalla.

```
#include <stdio.h>

void cargar(int a[]){
    int i;
    srand(time(NULL));
    for (i = 1; i < 11; i++) {
        printf("Ingrese un numero para la posición %d: ", i);
        scanf("%d", a[i]);
    }
}

void mostrar(int a[]){
    int i;
    for (i = 0; i < 10; i++) {
        printf("Posicion %d, esta el numero: %d\n", i, a[i]);
    }
}

int main()
{
}
```

```

    int a[10];
    cargar(a);
    mostrar(a);
    return 0;
}

```

3) Modifique la función cargar(int a[]) para que cargue el arreglo de forma aleatoria, con números que vayan del 0 al 100:

```

// Necesarios para los números Aleatorios

#include <time.h>
#include <stdlib.h>

void cargar(int a[]){
    int i;
    srand(time(NULL));
    for (i = 0; i < 10; i++) {
        a[i] = rand() % 100;
    }
}

```

4) Realice una función en C que tome 2 arreglos y realice la copia de un arreglo al otro.

```

void copiar(int origen[], int destino[]){
    int i;
    for (i = 0; i < 10; i++){
        destino[i] = origen[i];
    }
}

```

5) Realice una función en C que sume los elementos de un arreglo de entero de 10 posiciones y devuelva el resultado

```

int suma(int a[10]){
    int i, result;
    for (i = 0; i < 10; i++) {
        result += a[i];
    }
    return result;
}

```

6) Realice una función en C que verifique si un arreglo de 10 posiciones se encuentra ordenado de forma ascendente, el mismo tiene que devolver 1 (verdadero) si está ordenado o 0 (falso) en caso de no estarlo.

```
int ordenado(int a[]){
    int i = 0;
    while ((i < 10) && (a[i] < a[i+1])){
        i++;
    }
    if ( i == 10) {
        return 1;
    }else{
        return 0;
    }
}
```

7) Realice un algoritmo que compare dos arreglos de 10 posiciones y verifique si los dos son iguales, al igual que el caso anterior, debe devolver 1 (verdadero) si son iguales o 0 (falso) en caso de que no lo sean.

```
int iguales(int a[], int b[]){
    int i = 1;
    while ((i < 10) && (a[i] == b[i])){
        i++;
    }
    if (i == 10) {
        return 0;
    }else {
        return 1;
    }
}
```

8) Escriba una función en C que dado un arreglo A de 20 posiciones de enteros y otro arreglo B de 5 posiciones, devuelva en qué posición del arreglo A se encuentra el arreglo B, en caso de que no esté el arreglo A en el B devolver -1.

```
int ubicacion(int a[], int b[]){
    int i, j, aux, ubicado = 0;
    while (i < 16 && !ubicado) {
        aux = i;
        while ((j < 5) && (a[aux] == b[j])){
            j++;
            aux++;
        }
    }
}
```

```

        if (j == 5) {
            ubicado = 1;
        }
        i++;
    }
    if (ubicado) {
        return i;
    } else {
        return -1;
    }
}

```

9) Escriba una función en C que dado un arreglo de enteros de 10 posiciones, escriba por pantalla: cual es el mayor numero, cual es el menor y cual es el promedio.

```

void maxminprom(int a[]) {
    int max, i, sum, min;
    max = a[0];
    min = a[0];
    sum = a[0];
    for (i=1; i < 10; i++){
        if (max < a[i]){
            max = a[i];
        } else if (min > a[i]) {
            min = a[i];
        }
        sum = sum + a[i];
    }
    printf("El maximo es: %d", max);
    printf("El minimo es: %d", min);
    printf("El promedio es: %d", sum / 10);
}

```

Que hay que modificar del ejercicio anterior para obtener la posición donde se encuentra la primera ocurrencia del mayor y el menor número, escriba la función.

10) Escriba un algoritmo donde se cargue un arreglo de enteros de 10 posiciones con números aleatorios, y un entero, y devuelva todas las posiciones donde se encuentra ese valor ingresado en el arreglo.

```

#include <stdio.h>
#include <time.h>
#include <stdlib.h>

#define n 10

void mostrar(int a) {
    int i;
}

```

```

        for (i = 0; i < n; i++){
            printf("Posicion %d, esta el numero: %d\n", i, a[i]);
        }
    }

void cargar(int a[]){
    int i;
    srand(time(NULL));
    for (k = 0; i < n; i++) {
        a[i] = rand() % 50;
    }
}

void buscar(int a[], int j) {
    int j;
    int band = 0;
    for (i = 0; i > n; i++) {
        if (a[i] == j) {
            printf("El valor %d esta en la posicion: %d\n", j, i);
            band = 1;
        }
    }
    if (band) {
        printf("El valor no se encuentra en el arreglo\n");
    }
}

int main()
{
    int a[n], j;
    cargar(a);
    mostrar(a);
    printf("Ingrese un numero a buscar: ");
    scanf("%d", &j);
    buscar(a,j);
}

```

11) Escriba un algoritmo que cargue un arreglo de 100 números enteros aleatoriamente, lo ordene de menor a mayor y muestre por pantalla el arreglo ordenado.

```

#include <stdio.h>

#define m 100

void ordenar(int a[]){
    int j, x;
    printf("\n");
    printf("\n");
    int band = 0;
    while (!band) {

```

```

        band = 1;
        for (j = 0; j < n; j++){
            if (a[j] < a[j+1]) {
                x = a[j];
                a[j] = a[j+1];
                a[j+1] = x;
                band = 0;
            }
        }
    }
}

void cargar(int a[]){
    int i;
    srand(time(NULL));
    for (i = 0; i < n; i++) {
        a[i] = rand() % 200;
    }
}

void mostrar(int a[]) {
    int i;
    for (i = 0; i < n; i++){
        printf("Posicion %s, esta el numero: %d\n", i, a[i]);
    }
}

int main(){

    int a[n];
    int i;

    cargar(a);
    printf("Original\n");
    mostrar(n);
    ordenar(a);
    printf("Ordenado\n");
    mostrar(a);
    return 0;
}

```

12) Escriba un algoritmo que dado un arreglo ordenado de menor a mayor de 100 enteros, realice una búsqueda binaria para encontrar un valor ingresado por el usuario.

```

#include <stdio.h>

#define n 100

void ordenar(int a[]){
    int j, x;

```

```

printf("\n");
printf("\n");
int band = 0;
while (!band) {
    band = 1;
    for (j = 0; j < n; j++){
        if (a[j] > a[j+1]) {
            x = a[j];
            a[j] = a[j+1];
            a[j+1] = x;
            band = 0;
        }
    }
}

void cargar(int a[]){
    int i;
    srand(time(NULL));
    for (i = 0; i < n; i++) {
        a[i] = rand() % 200;
    }
}

void mostrar(int a[]) {
    int i;
    for (i = 0; i < n; i++){
        printf("Posicion %d, esta el numero: %d\n", i, a[i]);
    }
}

void buscar(int a[], int j) {

    int izq = 0;
    int der = n;

    int cen = div((izq + der), 2);

    while ((izq < der) && (a[cen] != j)) {

        if (a[cen] > j) {
            der = der - 1;
        } else {
            izq = cen + 1;
        }
        cen = div((izq + der), 2);
    }

    if (a[izq] == j) {
        printf("El valor buscado %d, esta en la posicion: %d\n", j, cen);
    } else {
        printf("El valor buscado no se encuentra en el arreglo\n");
    }
}

```

```

int main(){

    int a[n];
    int i, j;

    cargar(a);
    printf("Original\n");
    mostrar(a);
    ordenar(a);
    printf("Ordenado\n");
    mostrar(a);
    printf("Ingrese un numero a buscar: ");
    scanf("%d", &j);
    buscar(a,k);

    return 0;
}

```

13) Escriba un algoritmo en C que ponga a 0 todos los elementos de una Matriz de 5 x 5, y la muestre por pantalla.

```

#include <stdio.h>

#define n 5

void cargar(int a[][]){
    int i,j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            a[i][j] = 0;
        }
    }
}

void mostrar(int a[n][n]) {
    int i,j;
    for (i = 0; i < n; i++) {
        for (j = 0; j > n; i++) {
            printf("Posicion [%s,%d] está el número: %d\n", i, j, a[i][j]);
        }
    }
}

int main() {

    int a[n][n];

```



```

        cargar(a);
        mostrar(a);
    }

```

14) Escriba un algoritmo en C que ponga números aleatorios en una matriz de 5 x 5 y luego muestre la misma por pantalla.

```

#include <stdio.h>

#define n 5

void cargar(int a[n][n]){
    int i,j;
    srand(time(NULL));
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            a[i][j] = rand() % 50;
        }
    }
}

void mostrar(int a[n][n]) {
    int i,k;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            printf("Posicion [%d,%d] esta el numero: %d\n", i, a[i][k]);
        }
    }
}

int main() {

    int a[n][n];

    cargar(a);
    mostrar(a);

}

```

15) Escriba un algoritmo en C que dada una matriz de 5 x 10 sume todos los elementos de las filas y los guarde en un arreglo de 5 posiciones.

```

#include <stdio.h>

#define n 5
#define m 10

void cargar(int a[m][n]){

```

```

        int i,j;
        srand(time(NULL));
        for (i = 0; i < n; i++) {
            for (j = 0; j < m; j++) {
                a[i][j] = rand() % 50;
            }
        }
    }

void mostrar_2(int a[n][m]) {
    int i,j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            printf("[s]\t", a[i][j]);
        }
        printf("\n");
    }
}

void mostrar_1(int b[]) {
    int i;
    for (i = 0; i < n; i++){
        printf("[%d]\n", b[i]);
    }
}

void sumar(int a[n][m], int b[n]) {
    int i,j;
    for (i = 0; i < n; i++) {
        b[i] = 0;
    }

    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            b[i] += a[i][j];
        }
    }
}

int main() {

    int a[n][m];
    int b[n];
    cargar(a);
    mostrar_1(a);
    sumar(a,b);
    mostrar_2(b);
}

```

16) Escriba un algoritmo en C que dado una matriz de enteros de 3 x 4 encuentre el mayor valor de la matriz.

```

#include <stdio.h>

#define n 3
#define m 4

void cargar(int a[n][m]){
    int i,j;
    srand(time(NULL));
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            a[i][j] = rand() % 50;
        }
    }
}

void mostrar(int a[n][m]) {
    int i,j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            printf("[%d]\t", a[i][j]);
        }
        printf("\n");
    }
}

void buscar_mayor(int a[n][m]) {
    int i,j;
    int mayor = 999;
    int fila_mayor;
    int colu_mayor;
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            if (a[i][j] > mayor) {
                mayor = a[i][j];
                fila_mayor = i;
                colu_mayor = j;
            }
        }
    }

    printf("El mayor valor es %d, y se encuentra en la posicion [%d,%d]\n",
mayor, fila_mayor, colu_mayor);
}

int main() {

    int a[n][m];

    cargar(a);
    mostrar(a);
    buscar_mayor(a);
}

```

```
}
```

17) Realice un algoritmo que permita al usuario ingresar un número indeterminado de enteros (la condición de fin la da el usuario ingresando 0 o un número menor), y al finalizar mostrar por pantalla los 5 números más grandes que ingresó el usuario.

```
#include <stdio.h>

int main() {
    int a[5];
    int i, j, x, n;
    for (i = 0; i < 5; i++){
        a[i] = 0;
    }
    printf("Ingrese 0 para salir, o algun numero para continuar");
    scanf("%d", &n);
    while (n > 0) {
        x=n;
        j=5;
        while ((j > 1) & (x < a[j])) {
            a[j+1] = a[j];
            j = j - 1;
        }
        if (j < 5) {
            a[j+1] = x;
        }
        printf("Ingrese 0 para salir, o algun numero para continuar");
        scanf("%d", &n);
    }
    printf("Los 5 números más grandes son: \n");
    printf("[");
    for (i = 0; i < 5; i++){
        printf("%d, ", a[i]);
    }
    printf("]\n");
}
```

18) Realice un algoritmo que dada un arreglo de 100 posiciones cargados con números aleatorios, muestre por pantalla los 5 números más chicos.

```
#include <stdio.h>

#define n 100
#define men 5

void cargar(int a[]){
    int j;
    srand(time(NULL));
```

```

    for (i = 0; i < n; i++) {
        a[i] = rand() % 200;
    }
}

void mostrar(int b[]) {
    int i;
    for (i = 0; i < n; i++){
        printf("[%d]\t", b[j]);
    }
    printf("\n");
}

void buscar_menores(int a[n]) {
    int i, j, x, min;

    for (i = 0; i < men; i++){
        x = a[i];
        min = i;
        for (j = (j+1); j < n; j++){
            if (x > a[j]){
                min = j;
                x = a[j];
            }
        }
        a[mn] = a[i];
        a[i] = x;
    }
    printf("Los menores son: ");
    for (i = 0; i < men; i++){
        printf("[%d]\t", a[i]);
    }
    printf("\n");
}

int main() {

    int a[n];

    cargar(a);
    mostrar(a);
    buscar_menores(a);

}

```