

**UNIVERSIDAD AUTÓNOMA DE BAJA  
CALIFORNIA**  
**Facultad de Ingeniería, Arquitectura y Diseño**

**Ingeniero en Software y Tecnologías Emergentes**



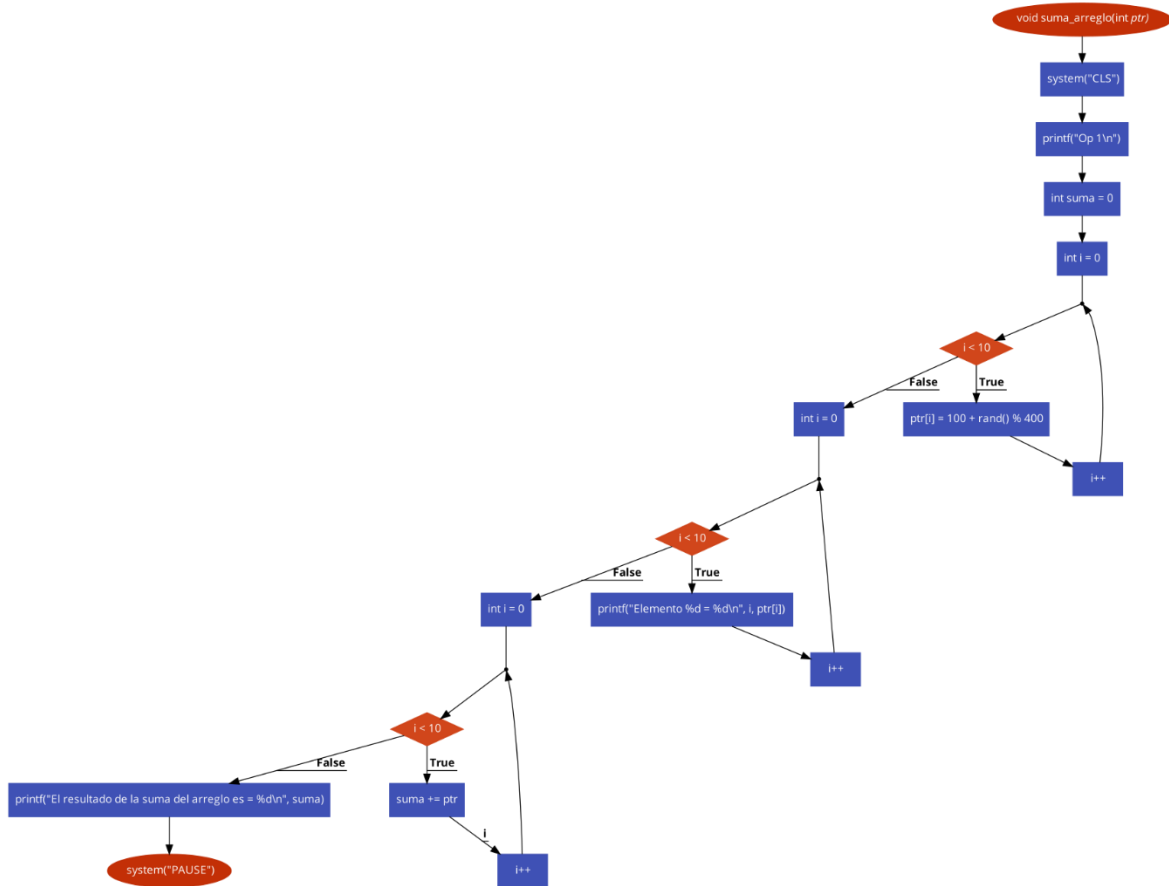
**Nombre Alumno:**  
Eliel Alfonso Ontiveros Ojeda

**Grupo:**  
932

30/09/2023

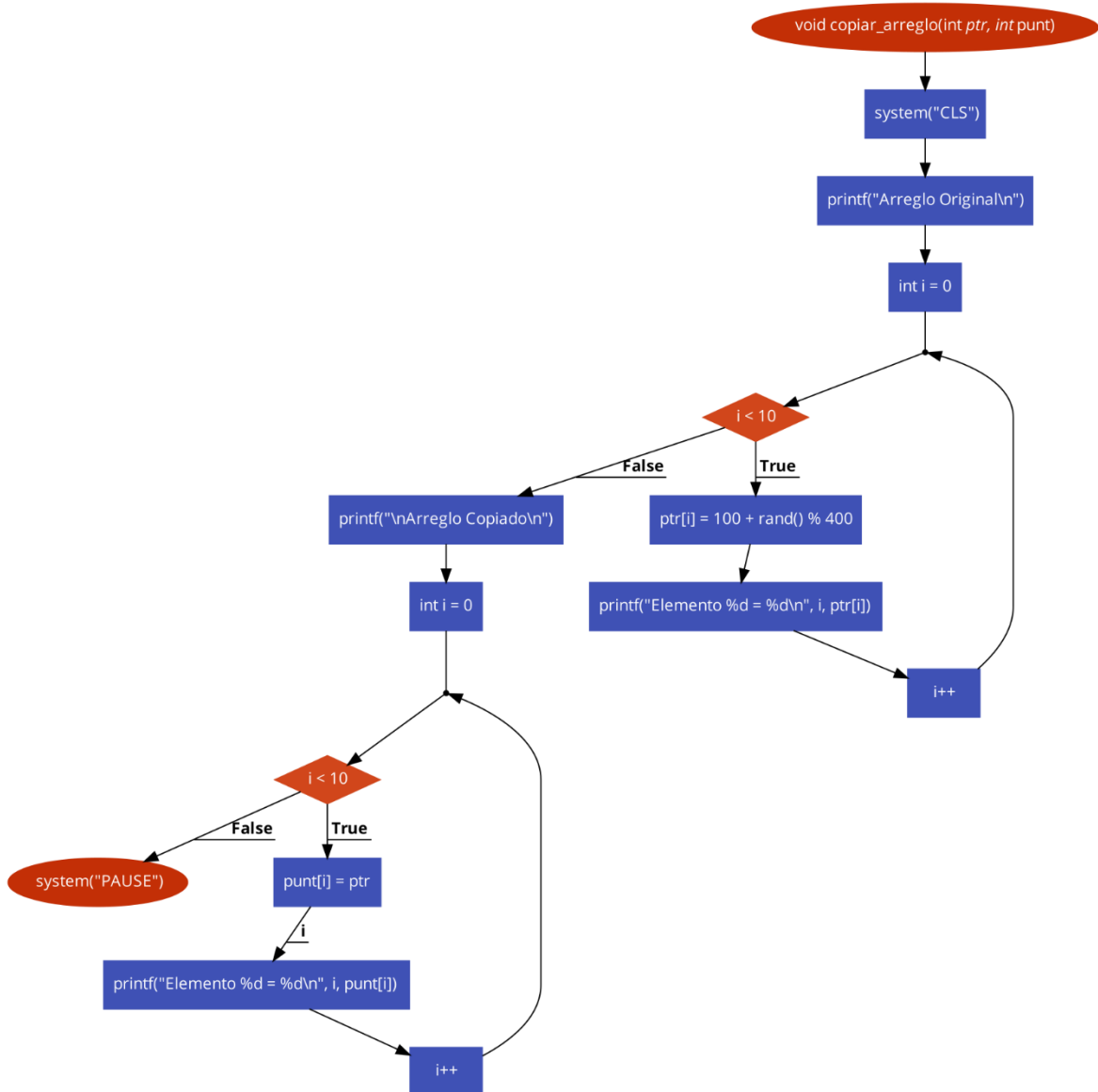
1. Suma de elementos en un arreglo: Crea una función que calcule la suma de los elementos en un arreglo utilizando apuntadores y aritmética de direcciones. La función debe tomar un apuntador al arreglo y devolver la suma.

## DIAGRAMA DE FLUJO



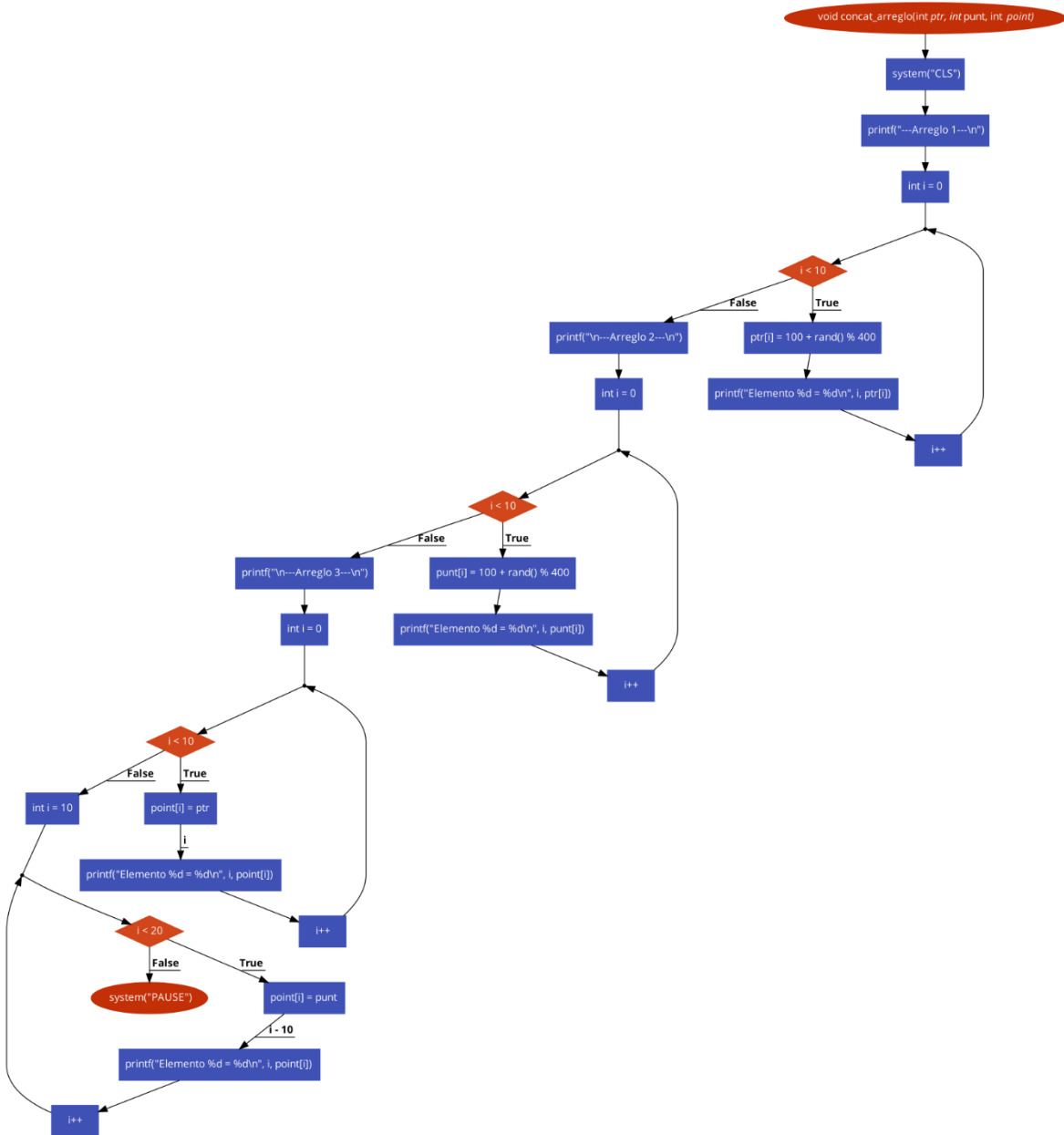
2. Copia de arreglos: Crea una función que copie un arreglo de origen en un arreglo de destino. Utiliza apuntadores para realizar esta operación. La función debe tomar dos apuntadores como argumentos, uno para el arreglo de origen y otro para el arreglo de destino.

### Diagrama De Flujo



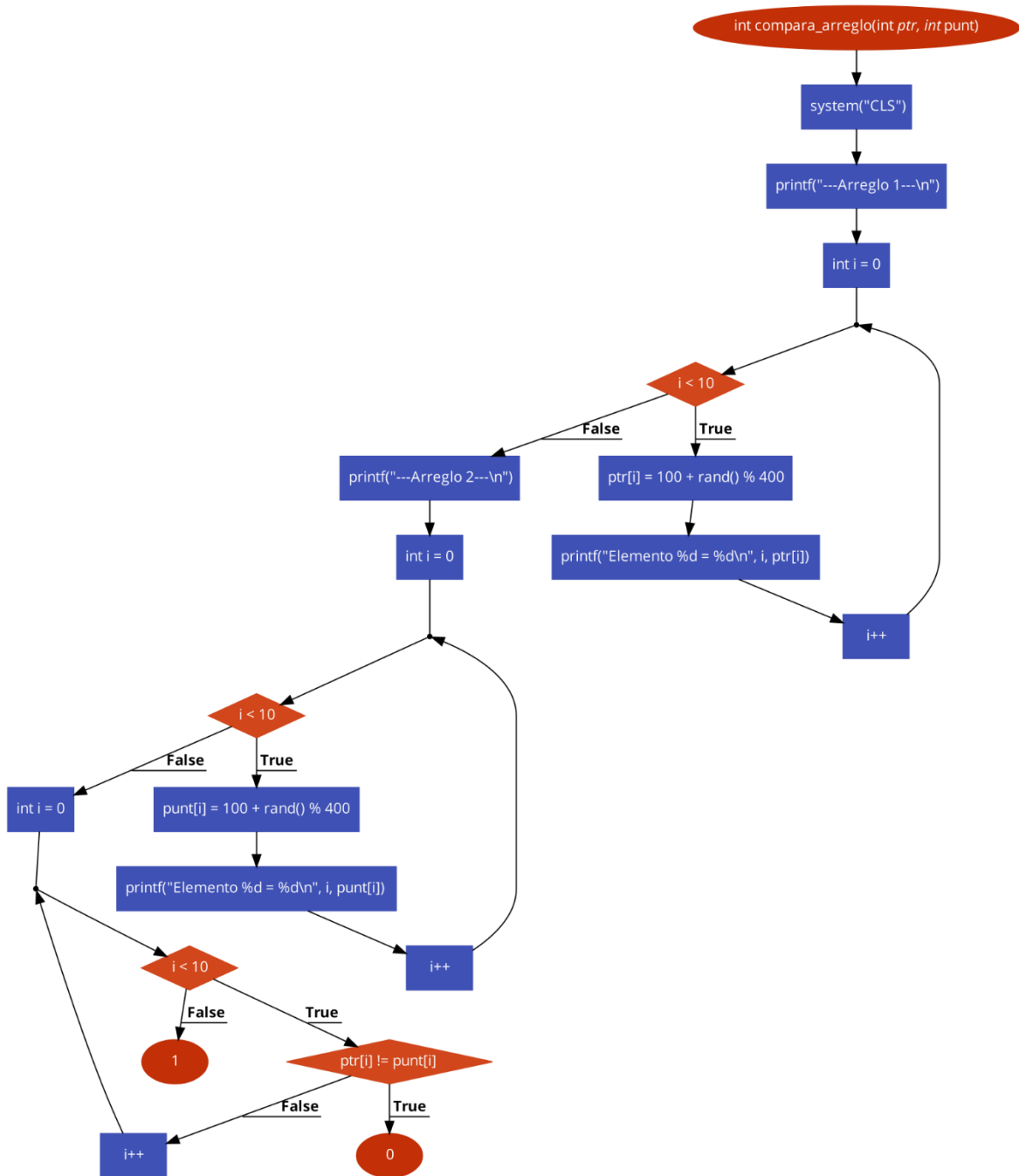
3. Concatenación de arreglos: Crea una función que tome dos arreglos de entrada y los concatene en un tercer arreglo. Utiliza apuntadores y aritmética de direcciones para realizar la concatenación. La función debe tomar tres apuntadores como argumentos: dos para los arreglos de entrada y uno para el arreglo de destino.

### Diagrama De Flujo



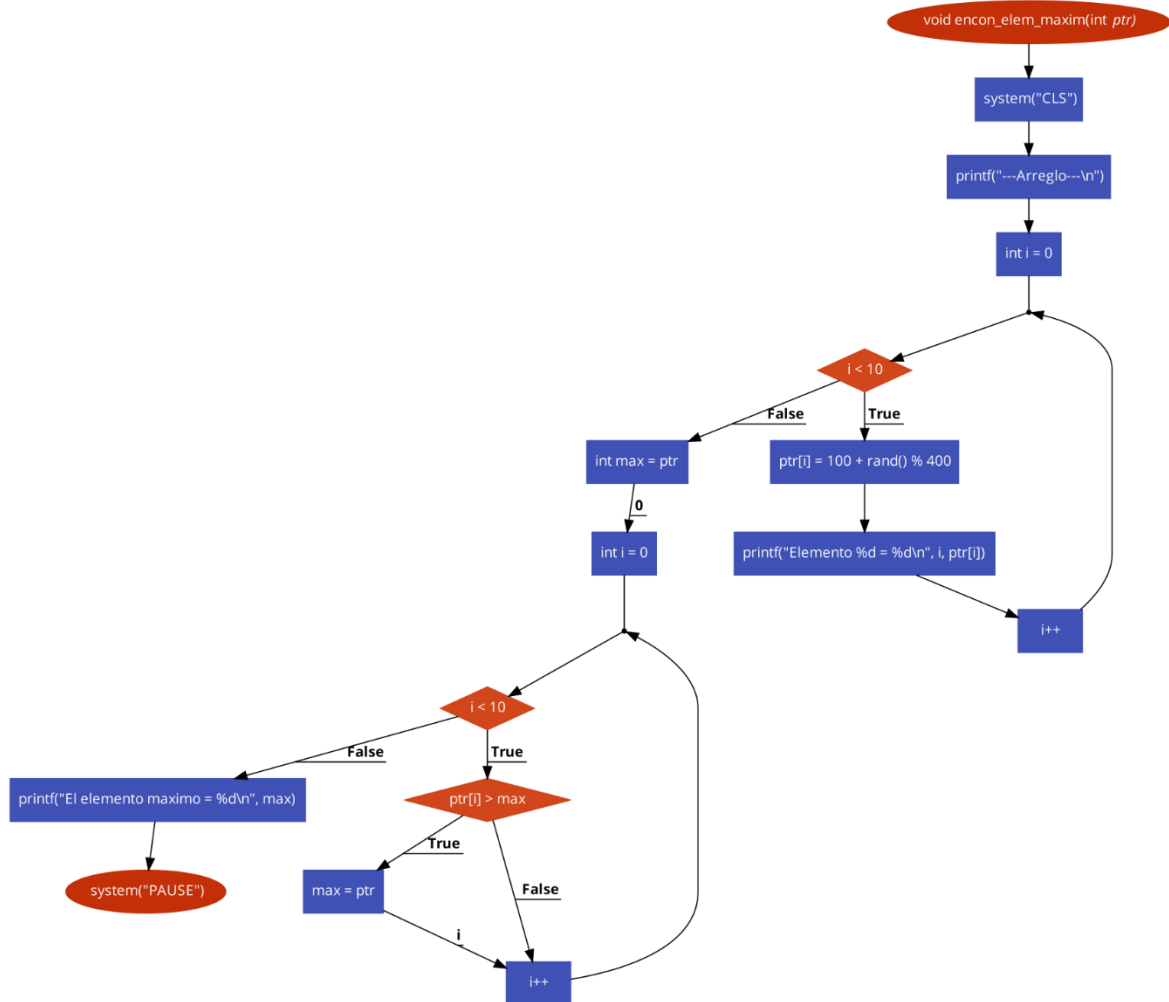
4. Comparación de arreglos: Crea una función que compare dos arreglos y determine si son iguales. Utiliza apuntadores y aritmética de direcciones para realizar la comparación. La función debe tomar dos apuntadores como argumentos y devolver un valor que indique si los arreglos son iguales.

### Diagrama De Flujo



5. Encontrar el elemento máximo: Crea una función que encuentre y devuelva el elemento máximo en el arreglo.

### Diagrama De Flujo



## Documentación

```
1  /*
2  --> Nombre:
3  suma_arreglo
4  --> Descripcion:
5  Realiza la operación de suma de elementos en un arreglo
6  genera un arreglo aleatorio de 10 elementos, muestra sus valores,
7  calcula la suma de los elementos y la imprime en la consola
8  --> Parametros:
9  o *ptr = Puntero donde se guardara el arreglo que se utilizara para sumar sus elementos
10 */
11 void suma_arreglo(int *ptr)
12 {
13     system("CLS");
14     printf("Op 1\n");
15     int suma = 0;
16     for (int i = 0; i < 10; i++)
17     {
18         ptr[i] = 100 + rand() % 400;
19     }
20
21     for (int i = 0; i < 10; i++)
22     {
23         printf("Elemento %d = %d\n", i, ptr[i]);
24     }
25
26     for (int i = 0; i < 10; i++)
27     {
28         suma += ptr[i];
29     }
30
31     printf("El resultado de la suma del arreglo es = %d\n", suma);
32
33     system("PAUSE");
34 }
```

```
1  /*
2  --> Nombre:
3  copiar_arreglo
4  --> Descripcion:
5  Crea un arreglo original, copia sus elementos en otro arreglo y muestra ambos arreglos en la consola
6  se utiliza para demostrar cómo se copian elementos de un arreglo a otro
7  --> Parametros:
8  o *ptr = Puntero donde se guardara el arreglo original
9  o *punt = Puntero donde se guardara el arreglo copiado
10 */
11 void copiar_arreglo(int *ptr, int *punt)
12 {
13     system("CLS");
14     printf("Arreglo Original\n");
15     for (int i = 0; i < 10; i++)
16     {
17         ptr[i] = 100 + rand() % 400;
18         printf("Elemento %d = %d\n", i, ptr[i]);
19     }
20
21     printf("\nArreglo Copiado\n");
22     for (int i = 0; i < 10; i++)
23     {
24         punt[i] = ptr[i];
25         printf("Elemento %d = %d\n", i, punt[i]);
26     }
27
28     system("PAUSE");
29
30 }
```



```
1  /*
2  --> Nombre:
3  concat_arreglo
4  --> Descripcion:
5  Crea dos arreglos originales, luego concatena sus elementos en un tercer arreglo
6  y muestra los tres arreglos en la consola, esto demuestra cómo se pueden combinar
7  elementos de dos arreglos en un tercero
8  --> Parametros:
9  o *ptr = Puntero donde se guardara un arreglo de 10 elementos
10 o *punt = Puntero donde se guardara un arreglo de 10 elementos
11 o *point = Puntero donde se va a concatenar los elementos de los dos arreglos ptr y punt
12 */
13 void concat_arreglo(int *ptr, int *punt, int *point)
14 {
15     system("CLS");
16     printf("---Arreglo 1---\n");
17     for (int i = 0; i < 10; i++)
18     {
19         ptr[i] = 100 + rand() % 400;
20         printf("Elemento %d = %d\n", i, ptr[i]);
21     }
22
23     printf("\n---Arreglo 2---\n");
24     for (int i = 0; i < 10; i++)
25     {
26         punt[i] = 100 + rand() % 400;
27         printf("Elemento %d = %d\n", i, punt[i]);
28     }
29
30     printf("\n---Arreglo 3---\n");
31     for (int i = 0; i < 10; i++) {
32         point[i] = ptr[i];
33         printf("Elemento %d = %d\n", i, point[i]);
34     }
35
36     for (int i = 10; i < 20; i++) {
37         point[i] = punt[i - 10];
38         printf("Elemento %d = %d\n", i, point[i]);
39     }
40
41     system("PAUSE");
42 }
```

```
1  /*
2  --> Nombre:
3  compara_arreglo
4  --> Descripcion:
5  Compara dos arreglos elemento por elemento y devuelve 1 si son iguales y 0 si son
6  diferentes se utilizan arreglos aleatorios para demostrar la comparación de arreglos
7  --> Parametros:
8  o *ptr = Puntero donde se guardara un arreglo de 10 elementos
9  o *punt = Puntero donde se guardara un arreglo de 10 elementos
10 --> Valor de Retorno:
11 o Regresa 0 si los elementos de los arreglos son diferentes
12 o Regresa 1 si los elementos de los arreglos son iguales
13 */
14 int compara_arreglo(int *ptr, int *punt)
15 {
16     system("CLS");
17     printf("---Arreglo 1---\n");
18     for (int i = 0; i < 10; i++)
19     {
20         ptr[i] = 100 + rand() % 400;
21         printf("Elemento %d = %d\n", i, ptr[i]);
22     }
23
24     printf("---Arreglo 2---\n");
25     for (int i = 0; i < 10; i++)
26     {
27         punt[i] = 100 + rand() % 400;
28         printf("Elemento %d = %d\n", i, punt[i]);
29     }
30
31     for (int i = 0; i < 10; i++)
32     {
33         if (ptr[i] != punt[i])
34         {
35             return 0;
36         }
37     }
38
39     return 1;
40
41 }
```



```
1  /*
2  --> Nombre:
3  encon_elem_maxim
4  --> Descripcion:
5  Genera un arreglo aleatorio de 10 elementos
6  y encuentra el valor máximo en ese arreglo
7  Luego muestra el elemento máximo en la consola
8  --> Parametros:
9  o *ptr = Puntero donde se guardara un arreglo de 10 elementos
10 */
11 void encon_elem_maxim(int *ptr)
12 {
13     system("CLS");
14     printf("---Arreglo---\n");
15     for (int i = 0; i < 10; i++)
16     {
17         ptr[i] = 100 + rand() % 400;
18         printf("Elemento %d = %d\n", i, ptr[i]);
19     }
20
21     int max = ptr[0];
22     for (int i = 0; i < 10; i++)
23     {
24         if(ptr[i] > max)
25         {
26             max = ptr[i];
27         }
28     }
29
30     printf("El elemento maximo = %d\n", max);
31
32     system("PAUSE");
33 }
```