

Informe

Laura Olivero

abril 2025

Introducción

Este informe presenta el análisis de varias funciones escritas en el lenguaje C++, enfocadas en el manejo de punteros, referencias, arreglos y memoria dinámica. Cada sección incluye una breve explicación y el código correspondiente para facilitar la comprensión de los conceptos.

1. Función modify()

Esta función muestra cómo se puede modificar una variable mediante un puntero.

```
1 void modify() {  
2     int var = 4;  
3     cout << "dir: " << &var << " val:" << var << endl;  
4     int *neew = &var;  
5     *neew = *neew + 5;  
6     cout << "val:" << var << " dir: " << &var << endl;  
7 }
```

Explicación: Se declara una variable 'var', luego se le asigna un puntero y se modifica su valor usando el puntero. Se imprimen direcciones y valores antes y después de la modificación.

2. Función referencias()

Demuestra el uso combinado de punteros y referencias para modificar valores.

```
1 void referencias() {  
2     int var2;  
3     int *p = &var2;  
4     *p = 5;  
5     int &ref = var2;  
6     ref += 2;  
7     cout << "puntero: " << p << " ref:" << &ref << endl;  
8 }
```

Explicación: Un puntero se usa para asignar un valor a una variable, y luego una referencia se usa para modificar ese mismo valor. Se imprime la dirección de memoria desde ambos enfoques.

3. Función arreglo()

Muestra cómo manipular un arreglo usando punteros.

```
1 void arreglo() {
2     int array[5] = {1, 2, 3, 4, 5};
3     int *p = array;
4     for (int i = 0; i < 5; ++i) {
5         p += i;
6         *p += 2;
7         cout << "puntero: " << p << "arreglo:" << &array << endl;
8     }
9 }
```

Explicación: El puntero se desplaza dentro del arreglo en cada iteración y modifica los valores. Esto puede causar comportamiento indefinido si se exceden los límites del arreglo.

4. Función dinamica()

Ejemplo de uso de memoria dinámica con una matriz.

```
1 void dinamica() {
2     int** matriz = new int*[5];
3     for (int i = 0; i < 5; ++i) {
4         matriz[i] = new int[5];
5     }
6     for (int i = 0; i < 5; ++i) {
7         for (int j = 0; j < 5; ++j) {
8             matriz[i][j] = i + j;
9         }
10    }
11    for (int i = 0; i < 5; ++i) {
12        delete[] matriz[i];
13    }
14    delete[] matriz;
15 }
```

Explicación: Se crea una matriz de 5x5 usando punteros dobles. Cada posición se llena con la suma de sus índices y luego se libera la memoria correctamente para evitar fugas.