

Programación en C++

Rubén Agapito

2021-1

Contenidos

| | |
|---------------------------------------|------------|
| Listado de Programas | iii |
| Prefacio | v |
| 1 Arreglos | 1 |
| 2 Vectores | 5 |

Programas

| | |
|---|---|
| programas/1.cc | 1 |
| programas/2.cc | 1 |
| programas/3.cc | 1 |
| programas/4.cc | 2 |
| programas/salida4 | 2 |

Prefacio

Los contenidos de estas notas están bajo constante supervisión. Si existe alguna sugerencia o error, por favor mandar un correo electrónico al autor: ruben.agapito@pucp.pe

Rubén Agapito
San Miguel, marzo de 2021
Lima, Perú.

Capítulo 1

Arreglos

Esta teoría es parte del legado de C++. Un arreglo es un tipo de data compuesto o data estructurada. Todos los elementos deben ser del mismo tipo y cada elemento puede ser accesado directamente. Algunas características:

- No pueden cambiar de tamaño una vez creados.
- Todos los elementos son del mismo tipo.
- Sus elementos son guardados contiguamente en memoria.
- Elementos individuales son accesibles por su posición o índice.
- Su primer elemento tiene índice cero.
- Su último elemento tiene índice `size-1`.
- Es responsabilidad del programador acceder los elementos de un arreglo dentro de sus límites.
- Es una buena costumbre inicializar los arreglos.

Al declarar arreglos debemos usar la siguiente sintaxis:

`TipoDeElemento NombreDeArreglo [número constante de elementos];`

Veamos algunos ejemplos con inicialización de elementos del arreglo:

```
int puntaje_examen[5]{18, 20, 8, 15, 13};

int puntaje_mas_alto[10]{18, 20}; // inicializa dos primeras entradas y resto son 0

double temp_altas[365]{0}; // inicializa todas las entradas con 0

int otro_arreglo[]{1, 2, 3, 4, 5}; // tamaño de arreglo es calculado automáticamente
```

Para acceder a los elementos de un arreglo la sintaxis es:

`nombre_arreglo[indice_de_elemento]`

Veamos un ejemplo

```
int puntajes[5]{18, 20, 17, 19, 13};

cout << "1er puntaje en índice 0: " << puntajes[0] << endl;
cout << "5to puntaje en índice 4: " << puntajes[4] << endl;
```

Usamos la misma sintaxis si deseamos guardar información en el arreglo:

```
int puntajes[5]{0};

cin >> puntajes[0];
cin >> puntajes[4];

puntajes[0] = 20; // sentencia de asignación
```


Veamos ahora un programa que resuma lo que hemos visto hasta el momento:

```
1 // Arreglos
2
3 #include <iostream>
4
5 using namespace std; // mejor no utilizar si invocamos más librerías
6
7 int main()
8 {
9
10     char vocales[]{'a', 'e', 'i', 'o', 'u'};
11     cout << "\nLa primera vocal es: " << vocales[0] << endl;
12     cout << "La última vocal es: " << vocales[4] << endl;
13
14     // cin >> vocales[5]; fuera de límites - no hacer esto!!
15
16     double alta_temp[] {90.1, 89.8, 77.5, 81.6};
17     cout << "\nLa primera temperatura es: " << alta_temp[0] << endl;
18
19     alta_temp[0] = 100.7; // asignar al primer elemento el valor de 100.
20
21     cout << "La primera temp más alta es ahora: " << alta_temp[0] << endl;
22     //
23
24     int puntajes[] {100, 90, 80, 70, 60};
25
26     cout << "\nPrimer puntaje en índice 0: " << puntajes[0] << endl;
27     cout << "Segundo puntaje en índice 1: " << puntajes[1] << endl;
28     cout << "Tercer puntaje en índice 2: " << puntajes[2] << endl;
29     cout << "Cuarto puntaje en índice 3: " << puntajes[3] << endl;
30     cout << "Quinto puntaje en índice 4: " << puntajes[4] << endl;
31
32     cout << "\nIngrese 5 puntajes: ";
33     cin >> puntajes[0];
34     cin >> puntajes[1];
35     cin >> puntajes[2];
36     cin >> puntajes[3];
37     cin >> puntajes[4];
38
39     cout << "\nEl arreglo actualizado es:" << endl;
40     cout << "Primer puntaje en índice 0: " << puntajes[0] << endl;
41     cout << "Segundo puntaje en índice 1: " << puntajes[1] << endl;
42     cout << "Tercer puntaje en índice 2: " << puntajes[2] << endl;
43     cout << "Cuarto puntaje en índice 3: " << puntajes[3] << endl;
44     cout << "Quinto puntaje en índice 4: " << puntajes[4] << endl;
45
46     cout << "\nObserve que el valor del nombre del arreglo es : " << puntajes << endl;
47
48     cout << endl;
49     return 0;
50 }
```

Luego de compilar este archivo y e ingresar una data cuando el archivo es ejecutado, obtenemos:

```
La primera vocal es: a
La última vocal es: u
```

La primera temperatura es: 90.1

La primera temp más alta es ahora: 100.7

Primer puntaje en índice 0: 100

Segundo puntaje en índice 1: 90

Tercer puntaje en índice 2: 80

Cuarto puntaje en índice 3: 70

Quinto puntaje en índice 4: 60

Ingrese 5 puntajes:

El arreglo actualizado es:

Primer puntaje en índice 0: 30

Segundo puntaje en índice 1: 20

Tercer puntaje en índice 2: 40

Cuarto puntaje en índice 3: 56

Quinto puntaje en índice 4: 84

Observe que el valor del nombre del arreglo es : 0x7ffd1eccc450

Capítulo 2

Vectores

Un vector es un tipo de arreglo (realmente un objeto) que es parte de la librería estándar de C++ (STL). A diferencia de los arreglos que son parte del legado de C++

