



Laboratorio 1

Parámetros Nontype en Templates

Supongamos que queremos una función que imprima los elementos de cualquier arreglo sin importar el tipo del elemento y el tamaño del arreglo. Sabemos que el tipo de cada elemento puede variar de un arreglo a otro, pero el tamaño de un arreglo es siempre un número entero (o entero sin signo). Lo hacemos usando el Programa 1. Tenemos dos parámetros de plantilla, T y N. El parámetro T puede ser de cualquier tipo; el parámetro N no es typename (el tipo está predefinido como entero).

Programa 1.

```
template<typename T, int N>
void print(T(&array)[N]){
    for(int i=0; i < N; i++){
        cout << array[i] << " ";
    }
    cout << endl;
}

int main() {
    int arr1[4] = {7, 3, 5, 1};
    double arr2[3] = {7.5, 6.1, 4.6};
    print(arr1);
    print(arr2);
    return 0;
}
```

Ejercicios

1. Escriba una función template para encontrar el índice del elemento más pequeño de un array de cualquier tipo. Pruebe la función con tres arrays de tipo int, double y char. Entonces imprima el valor del elemento más pequeño.
2. Si encontramos el elemento más pequeño en un array, podemos ordenar un array usando el algoritmo selection sort. En este algoritmo, encontramos el elemento más pequeño en el array y lo intercambiamos con el primer elemento. Después encontramos el elemento más pequeño de los elementos restantes y lo intercambiamos con el segundo elemento. Continuamos hasta que el array esté completamente ordenado. Escriba un programa que ordene tres arrays de tipo int, double y char.
3. Escriba una función template para encontrar un valor en un array. Pruebe la función con dos arrays de tipo int y char.
4. Defina una función para revertir el orden de los elementos de un array de cualquier tipo. Pruebe el programa con un array de int, double, char y string. Use una función auxiliar para intercambiar dos elementos cualesquiera. Use una función auxiliar para imprimir el contenido de un arreglo antes y después del intercambio.