Programming challenges in C++

1. Introduction to C++. Basic sentences and operators

Nacho Iborra

IES San Vicente

Table of Contents

Programming challenges in C++

1. Introduction to C++

C++ is a programming language created in the 80s as an evolution of C programming language. Its main target was to provide an object-oriented programming layer over the C structure.

What we are going to see in these documents is a correspondence between the concepts that you are going to learn in this module in C# and the same concepts applied to C++. So we are going to start with the software required to edit and compile C++ code, and then we will see some basic operations, such as console input/output and basic operations.

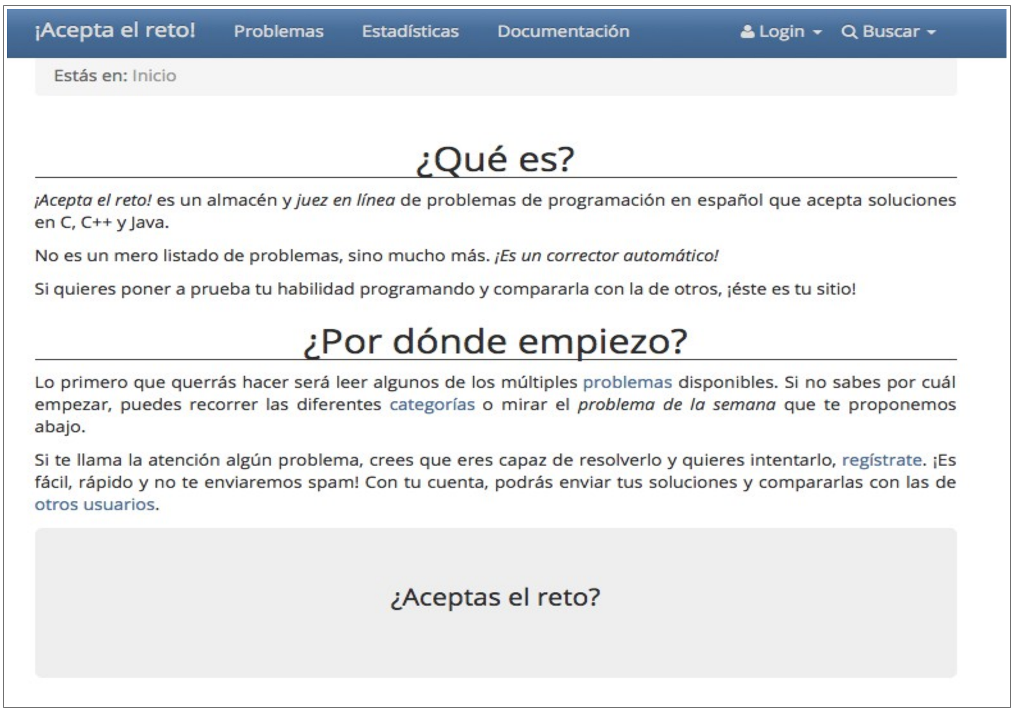
1.1. Required software

If you want to develop C++ applications, you can choose among several IDEs:

- In Windows, a good option to implement simple (one source file) programs is DevCPP, available at its SourceForge project web site.
- You can also use Visual Studio under Windows to develop C++ programs, although this option is heavier if you just want one source file to solve a small problem
- Geany is also available under several platforms (Windows, Linux, Mac OS X), but you need to have a C++ compiler installed.
- There are some other options, such as CodeBlocks (multi platform), XCode (for Mac OS X), and even some plugins for NetBeans or Eclipse (also multi platform).

1.2. The challenges web site

As we learn new concepts of C++, we are going to apply these concepts in some programming challenges. Most of them are published in *Acepta el reto* web site, so first of all, you must register in this site with the login and password that you want.



Estás en: Inicio

¿Qué es?

¡Acepta el reto! es un almacén y *juez en línea* de problemas de programación en español que acepta soluciones en C, C++ y Java.

No es un mero listado de problemas, sino mucho más. *¡Es un corrector automático!*

Si quieres poner a prueba tu habilidad programando y compararla con la de otros, ¡éste es tu sitio!

¿Por dónde empiezo?

Lo primero que querrás hacer será leer algunos de los múltiples problemas disponibles. Si no sabes por cuál empezar, puedes recorrer las diferentes categorías o mirar el *problema de la semana* que te proponemos abajo.

Si te llama la atención algún problema, crees que eres capaz de resolverlo y quieres intentarlo, regístrate. ¡Es fácil, rápido y no te enviaremos spam! Con tu cuenta, podrás enviar tus soluciones y compararlas con las de otros usuarios.

¿Aceptas el reto?

2. First steps with C++

2.1. Our first C++ program

A basic C++ program structure to say "Hello world" would be like this:

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world";
    return 0;
}
```

We need to include the *iostream* library and the *std* namespace to use elements such as *cout*, that we will see later as basic input/output mechanisms.

2.2. Basic operations

The most common basic operations that can be done in C++ are:

- Arithmetic operations: +, -, *, /, %(module), ++(increment), --(decrement)
- Assignments: =, +=, -=, *=, /=, %=
- Comparisons: >, >=, <, <=, ==, !=
  - NOTE: the comparison == works even with string elements
- Logical: && (AND), || (OR), ! (NOT)

2.3. Basic input / output

2.3.1. Basic input: cin

The easiest way to get some input from the user is to use the *cin* instruction from the *iostream* library. We can assign a value directly to any basic type:

```
int number;
string text;
cin >> number;
cin >> text;
```

You can join multiple variables with the >> operator, and type them either separated by whitespaces or by new lines (*Intro*)

```
int number1, number2;
cin >> number1 >> number2;
```

Be careful with strings

If we are getting strings, we must be careful with whitespaces, since they cut the string. In other words, if we type "Hello world" and want to save it in a *string* variable with *cin*, only *Hello* will be saved. If we want to get the whole string, we can use *getline* instead.

```
string text;
getline(cin, text);
```

2.3.2. Basic output: cout

On the other side, you can use the *cout* instruction to output values to the standard output. As we have seen with *cin*, you can join multiple values to output by using multiple << symbols.

```
int result = 12;
cout << "The result is " << 12;
```

You can also use the special character *endl* to create a new line.

```
int result = 12;
cout << "The result is " << result << endl;
```

2.3.3. The namespace std

If we don't add the *using namespace std* line at the beginning of our program, then we have to use the full path to the *cin* instruction. See the difference: this is how it works with the namespace:

```
#include <iostream>

using namespace std;

int main()
{
    int number;
    cin >> number;
}
```

And this is how you should type it without using the namespace:

```
#include <iostream>

int main()
{
    int number;
    std::cin >> number;
}
```

Besides, you will need to use the same prefix std:: in other elements, such as *cout, endl* or even *strings*.

2.4. Constants declaration

We can declare constant values in C++ in two ways:

- Using the *define* directive (typically at the beginning of the source file, after all the *include* directives

  ```
  #include <iostream>
  #define PI 3.14159;
  ```

- Using *const* in the middle of the code (either inside a given function or as a global variable)

  ```
  int main()
  {
     const float PI = 3.14159;
     ...
  }
  ```

2.5. Introduction to control structures

As we will see in future sessions, you can use most of the control structures present in other programming languages, such as C# or Java. For instance, you can use `if`, `while` or `for` clauses:

```
if (i < 10)
    cout << i;


for (i = 0; i < 10; i++)
    cout << i;
i = 0;


while(i < 10)
{
    cout << i;
    i++;
}
```

3. Challenges for this session

3.1. Sample challenge: Hello world

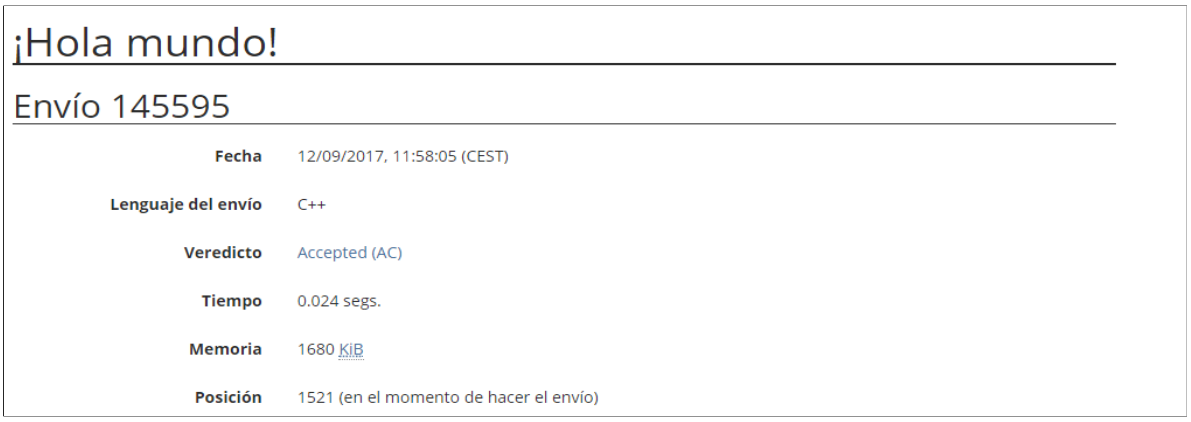
Have a look at this challenge from *Acepta el reto*. It asks you to read a number N and print N times the string "Hola mundo.". To solve this challenge, we could implement something like this in C++:

```
#include <iostream>

using namespace std;

int main()
{
    int i, times;
    cin >> times;
    for (i = 0; i < times; i++)
        cout << "Hola mundo." << endl;
    return 0;
}
```

Try to upload this code to *Acepta el reto* and see how it works fine:



¡Hola mundo!

Envío 145595

Fecha	12/09/2017, 11:58:05 (CEST)
Lenguaje del envío	C++
Veredicto	Accepted (AC)
Tiempo	0.024 segs.
Memoria	1680 KiB
Posición	1521 (en el momento de hacer el envío)

3.2. Try yourself: Christmas day

Let's try with another challenge. This time you will be provided with a list of dates (day and month). For each date, you must write "SI" if it's Christmas day (i.e. December 25th), or "NO" if it is not.