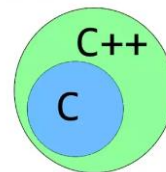




El Lenguaje C++

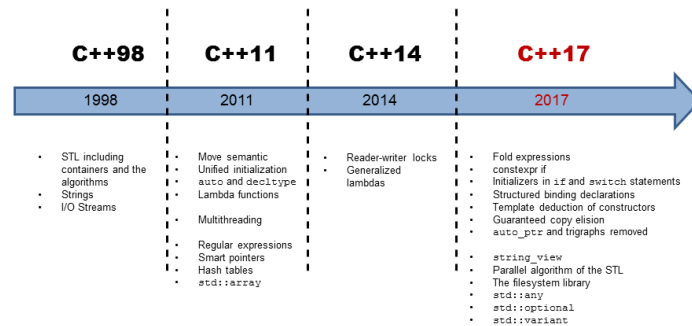
¿Qué es C++?

- ❑ Lenguaje de programación compilado y de “alto nivel”, creado en 1980 por Bjarne Stroustrup.
- ❑ Es un superconjunto del lenguaje C, las principales funcionalidades agregadas son la programación orientas a objetos y la librería estándar.



Versiones estándar de C++

- ❑ **C++98** : tiene la mayor parte de la librería estándar.
- ❑ **C++11** : auto, for basado en rangos, hash tables.
- ❑ **C++14**
- ❑ **C++17**



Programa básico en C++

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      cout << "Hello world";
7      return 0;
8  }
```

Directivas para el preprocesador

Espacio de nombres

Función principal

Instrucciones en C++

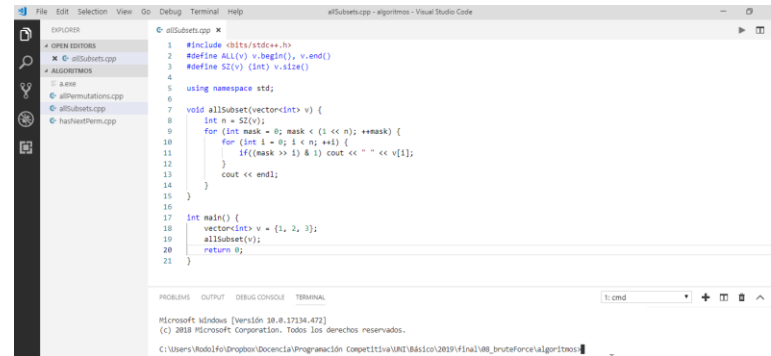
Ambiente de trabajo

❑ Editor de Texto

Visual Studio Code

❑ Compilador GCC (g++)

- Linux: GNU Compiler Collection (GCC)
- Windows: MinGW (versión GCC para Windows)

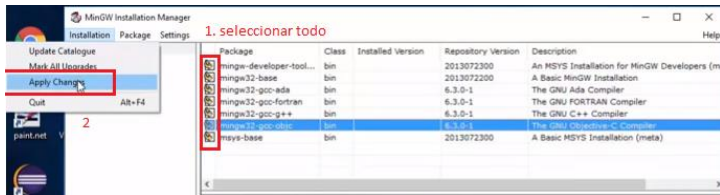


```
1 #include <bits/stdc++.h>
2 #define ALL(v) v.begin(), v.end()
3 #define SZ(v) (int) v.size()
4
5 using namespace std;
6
7 void allSubset(vector<int> v) {
8     int n = SZ(v);
9     for (int mask = 0; mask < (1 << n); ++mask) {
10         for (int i = 0; i < n; ++i) {
11             if ((mask >> i) & 1) cout << " " << v[i];
12         }
13         cout << endl;
14     }
15 }
16
17 int main() {
18     vector<int> v = {1, 2, 3};
19     allSubset(v);
20     return 0;
21 }
```

Instalación y Configuración del MinGW

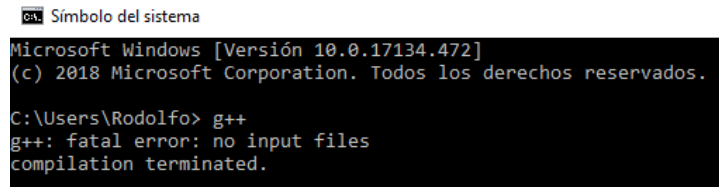
1. Instalar MinGW con las opciones por defecto.

Cuando nos consulte por los paquetes que queremos instalar, seleccionamos todos.

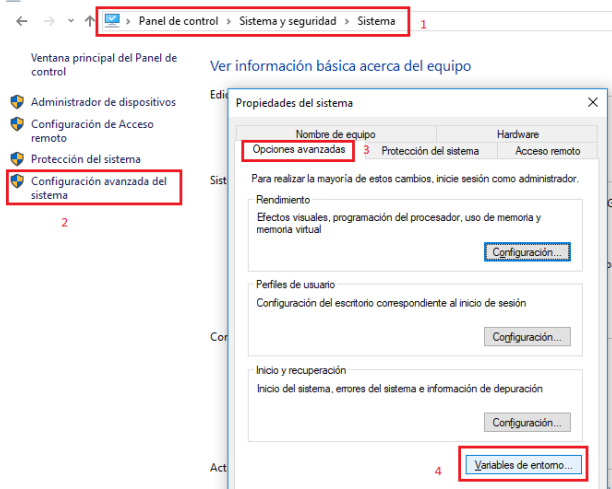


2. Verificar que el compilador esté en el *Path* del sistema:

Abrimos la terminal de Windows (window + R, escribimos cmd y damos enter), escribimos g++ y damos enter. Si nos aparece un mensaje como en la imagen entonces esta todo ok.



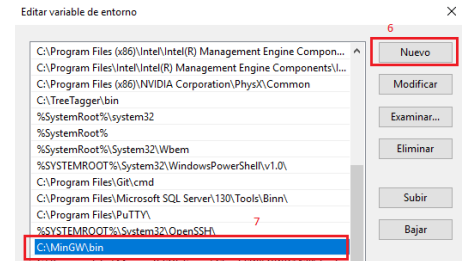
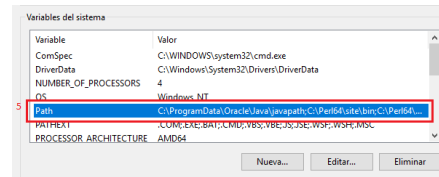
Instalación y Configuración del MinGW



3. En caso no sea exitoso el paso 2, tenemos que ir a :

Panel de control → Sistema y Seguridad → Sistema → Configuración avanzada del sistema → Opciones avanzadas → Variables de entorno.

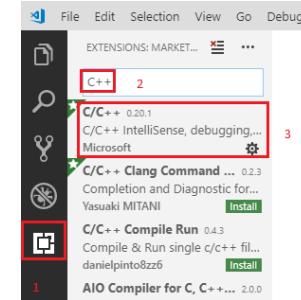
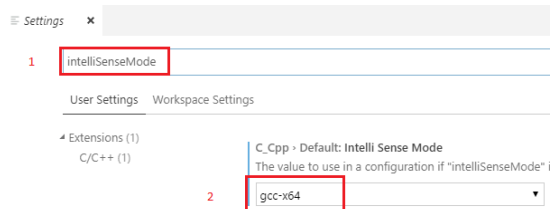
En variables del sistema damos doble click en *Path* y agregamos la ruta de la carpeta bin del MinGW.



Instalación y Configuración del VSCode

1. Instalar Visual Studio Code con las opciones por defecto.
2. Instalar la extensión de Microsoft para C/C++ (nos brinda IntelliSense)
3. Colocar el parámetro de configuración *intelliSenseMode* en la opción “gcc-x64”:

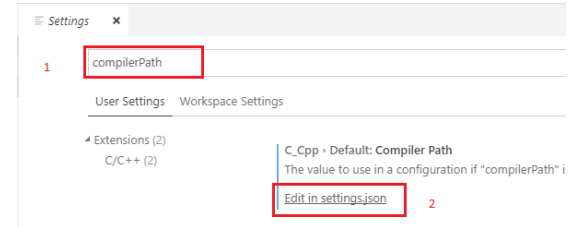
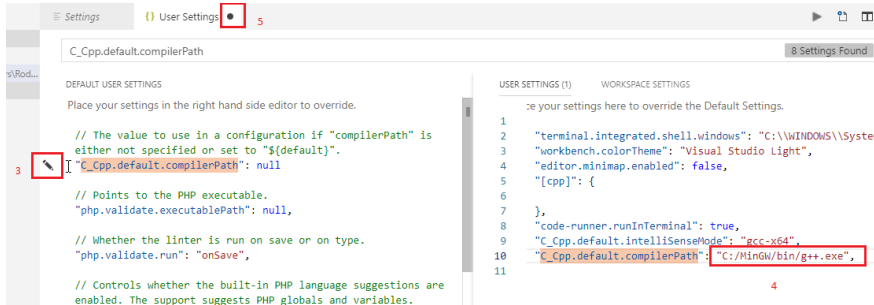
File → Preferences → Settings



Instalación y Configuración del VSCode

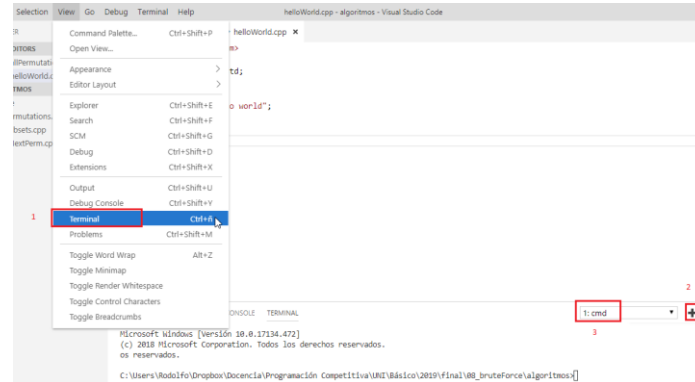
4. Cambiar el path de compilación que utiliza el VSCode, colocar la ruta del compilador g++ que instalamos.

File → Preferences → Settings

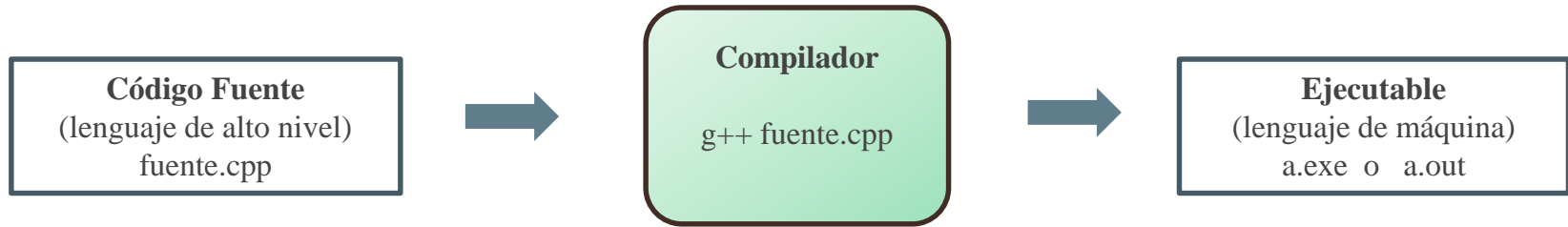


Configuración de la terminal

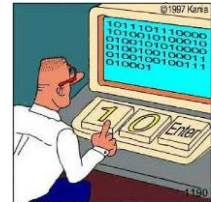
Agregamos una terminal de VSCode que sea del tipo “cmd” para poder realizar la compilación y ejecución de nuestro programa por ese medio .



Compilación

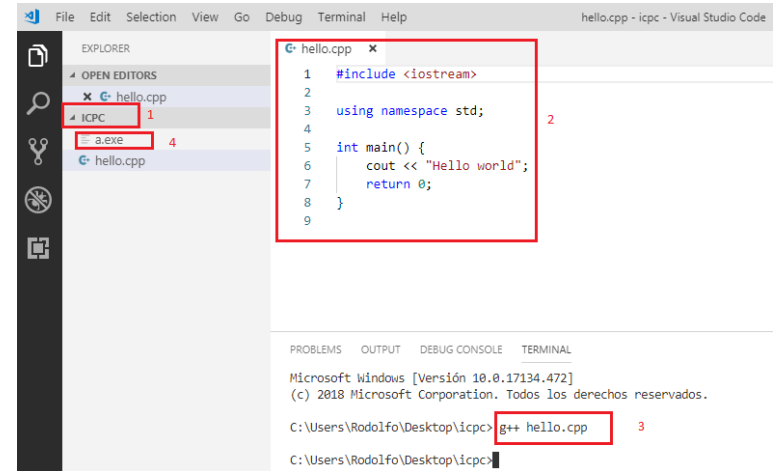


```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      cout << "Hello world";
7      return 0;
8  }
```

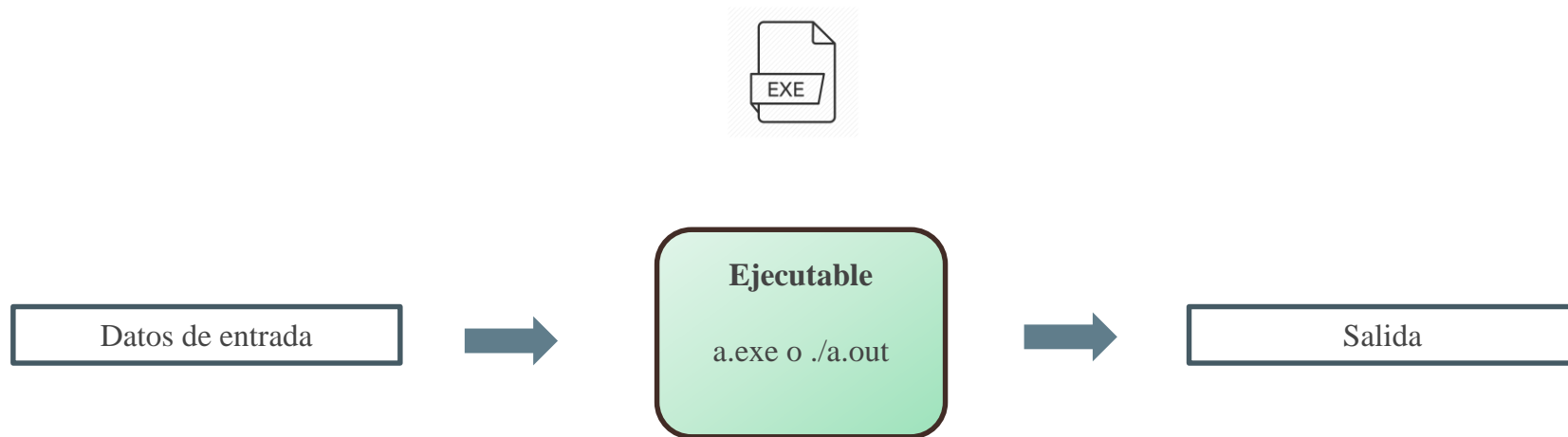


Compilación

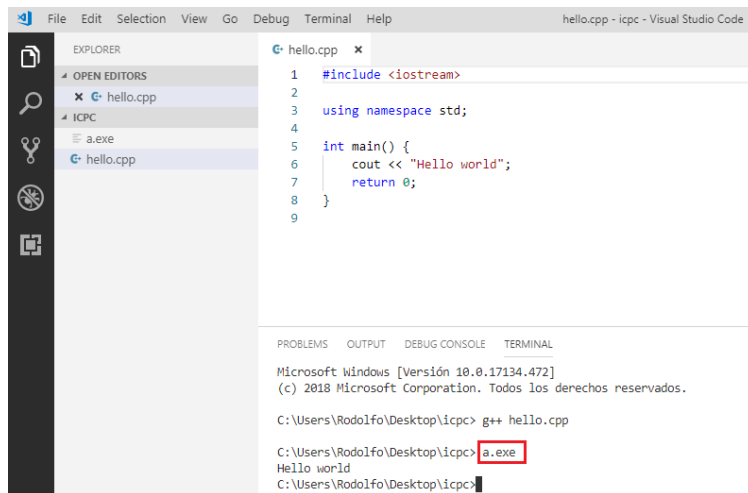
1. Es recomendable tener una carpeta de trabajo (en el ejemplo se llama ICPC) lo abrimos con File → Open Folder
2. Creamos un nuevo archivo en nuestra carpeta y lo guardamos con la extensión .cpp (en el ejemplo se llama hello.cpp)
3. Compilamos en la terminal.
4. Automáticamente se creó el ejecutable de nuestro programa.



Ejecución



Ejecución



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists the project files: 'hello.cpp' (selected), 'a.exe', and 'ICPC'. The main editor window displays the source code of 'hello.cpp':

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     cout << "Hello world";
7     return 0;
8 }
9
```

At the bottom, the TERMINAL panel shows the command prompt output:

```
Microsoft Windows [Versión 10.0.17134.472]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Rodolfo\Desktop\icpc> g++ hello.cpp

C:\Users\Rodolfo\Desktop\icpc> a.exe
Hello world
C:\Users\Rodolfo\Desktop\icpc>
```

The file 'a.exe' is highlighted with a red box in the terminal output.

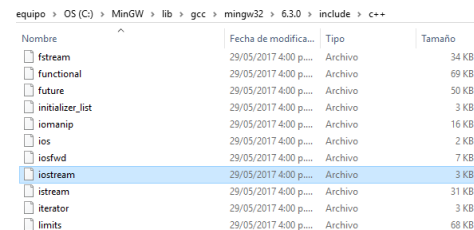


Preprocesador

- ❑ Procesa comandos (directivas) antes de la compilación real de un programa.
- ❑ Las directivas inician con el símbolo `#` y terminan con el final de línea.
- ❑ Utiliza una sintaxis diferente a C++.
- ❑ Las directivas más comunes son `#include` y `#define`.

Directiva #include

- ❑ Directiva que le indica al preprocesador incluir al programa actual el contenido de otro archivo.
- ❑ Los archivos que se incluyen en otros programas se denominan archivos de cabecera (headers) y son necesarios para usar ciertas funcionalidades.
- ❑ Por ejemplo `#include <iostream>` le dice al preprocesador que copie el contenido del archivo “iostream”, el cual nos permite manejar el ingreso y salida de datos.



equipo > OS (C:) > MinGW > lib > gcc > mingw32 > 6.3.0 > include > c++

Nombre	Fecha de modifica...	Tipo	Tamaño
fstream	29/05/2017 4:00 p...	Archivo	34 KB
functional	29/05/2017 4:00 p...	Archivo	69 KB
future	29/05/2017 4:00 p...	Archivo	50 KB
initializer_list	29/05/2017 4:00 p...	Archivo	3 KB
iomanip	29/05/2017 4:00 p...	Archivo	16 KB
ios	29/05/2017 4:00 p...	Archivo	2 KB
iosfwd	29/05/2017 4:00 p...	Archivo	7 KB
iostream	29/05/2017 4:00 p...	Archivo	3 KB
istream	29/05/2017 4:00 p...	Archivo	31 KB
iterator	29/05/2017 4:00 p...	Archivo	3 KB
limits	29/05/2017 4:00 p...	Archivo	68 KB

Directiva #include

<algorithm>	<initializer_list>	<numeric>	<strstream>
<array>	<iomanip>	<ostream>	<system_error>
<atomic>	<ios>	<queue>	<thread>
<bitset>	<iosfwd>	<random>	<tuple>
<chrono>	<iostream>	<ratio>	<type_traits>
<codecvt>	<istream>	<regex>	<typeindex>
<complex>	<iterator>	<scoped_allocator>	<typeinfo>
<condition_variable>	<limits>	<set>	<unordered_map>
<deque>	<list>	<shared_mutex>	<unordered_set>
<exception>	<locale>	<sstream>	<utility>
<forward_list>	<map>	<stack>	<valarray>
<fstream>	<memory>	<stdexcept>	<vector>
<functional>	<mutex>	<streambuf>	
<future>	<new>	<string>	

<cassert>	<cinttypes>	<csignal>	<cstdio>	<cwchar>
<ccomplex>	<ciso646>	<cstdalign>	<cstdlib>	<cwctype>
<cctype>	<climits>	<cstdarg>	<cstring>	
<cerrno>	<ctype>	<cstdbool>	<ctgmath>	
<cfenv>	<cmath>	<cstddef>	<ctime>	
<cfloat>	<csetjmp>	<cstdint>	<cuchar>	

Directiva #define

- ❑ Permite definir las denominadas macros, las cuales nos permiten manejar sustitución de términos en el código.
- ❑ La macros tienen el siguiente formato:

#define *identificador_macro* *secuencia_reemplazo*

- ❑ El preprocesador busca todas la ocurrencias de *identificador_macro* en el programa y las sustituye por *secuencia_reemplazo*, antes de iniciar la compilación real.

```
#define REP(i, a, b) for(int i = a; i < b; ++i)
```

Namespace

- ❑ Para evitar homonimia en los nombres de las variables o funciones, en C++ se define el espacio de nombres o **namespace**.
- ❑ C++ define sus funciones estándar en el espacio de nombres **std**.

```
namespace std{  
    istream cin;  
    ostream cout;  
    ...  
}
```

```
#include <iostream>  
using namespace std;  
  
namespace spanish {  
    void print() {  
        cout << "hola";  
    }  
}  
  
namespace english {  
    void print() {  
        cout << "hello";  
    }  
}  
  
int main() {  
    english :: print();  
    return 0;  
}
```

Función main

- ❑ Es la primera función a ser llamada, el resto de funciones son llamadas directa o indirectamente por ésta.
- ❑ El retornar 0 indica que el programa terminó con éxito.

```
int main() {  
    return 0;  
}
```

Variables y constantes

```
char letra;  
string nombre;  
bool flag;  
int edad;  
long long nro_conexiones;  
float costo;  
double area;
```

```
const int MESES = 12;  
const int DIAS_SEMANA = 7;  
const double PI = acos(-1.0);
```

Pair

Agrupar un par de elementos, los cuales pueden ser de distintos tipos de datos.

```
pair<string, int> p1, p2;  
p1.first = "Lima", p1.second = 1;  
p2 = {"Jesus Maria", 11};
```

Lectura y escritura

- ❑ La entrada estándar viene desde teclado.
- ❑ La salida estándar se muestra en consola.
- ❑ Ambos pueden ser redireccionados mediante línea de comandos.

```
#include <iostream>
using namespace std;

int main() {
    int a, b, c;
    cin >> a >> b >> c; //lectura
    c = a + b;
    cout << c; //escritura
    return 0;
}
```

Operadores

La mayoría de operadores son iguales a como los representamos en pseudocódigo, a excepción de:

Operador	Significado
%	Resto o módulo
/	División entera si ambos términos son enteros, en otro caso devuelve un real
!	Negación
&&	Conjunción
	Disyunción

Estructura Selectiva / If Else

```
#include <iostream>
using namespace std;

int main() {
    int mi_edad, tu_edad;
    cin >> mi_edad >> tu_edad;
    if (mi_edad < tu_edad) {
        cout << "soy menor";
    } else {
        cout << "soy mayor o tenemos la misma edad";
    }
    return 0;
}
```

Estructura Repetitiva / While

```
#include <iostream>
using namespace std;

int main() {
    int i = 0;
    while (i <= 10){
        cout << i << endl;
        i = i + 1;
    }
    return 0;
}
```

Estructura Repetitiva / For

```
#include <iostream>
using namespace std;

int main() {
    for (int i = 0; i <= 10; ++i) {
        cout << i << endl;
    }
    return 0;
}
```

Estructura Repetitiva / For basado en rango

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int arr[] = {3, 6, 2, -1, 8};
    for (int elem : arr) {
        cout << elem << endl;
    }
    return 0;
}
```

Arreglos

```
//max size  
const int N = 20;
```

```
double nota[N];  
int edad[N];  
char grid[N][N];
```

Dimensions	Example	Terminology									
1	<table><tr><td>0</td><td>1</td><td>2</td></tr></table>	0	1	2	Vector						
0	1	2									
2	<table><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	5	6	7	8	Matrix
0	1	2									
3	4	5									
6	7	8									
3	<table><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	5	6	7	8	3D Array
0	1	2									
3	4	5									
6	7	8									

Funciones y procedimientos

```
int factorial(int n) {  
    int ans = 1;  
    for (int i = 1; i <= n; ++i) {  
        ans = ans * i;  
    }  
    return ans;  
}
```

```
void print_array(int arr[], int n) {  
    for (int i = 0; i < n; ++i) {  
        cout << arr[i] << endl;  
    }  
}
```

Auto

Deduce el tipo de dato de una variable, a partir de la expresión usada para su inicialización.

```
//auto
auto cadena = "hola";
auto real = 5;
cout << "type: " << typeid(real).name() << endl;
```

```
int arr[] = {5, 10, 6};
//auto en for basado en rango
for (auto elem : arr) cout << elem << endl;

//auto&
for (auto& elem: arr) elem++;
for (int elem : arr) cout << elem << endl;
```

⟨bits/stdc++.h⟩

- ❑ Header especial que incluye a todos los headers del estándar C++.
- ❑ No pertenece al estándar C++ , pero si está en el compilador GCC.
- ❑ En las competencias de programación se usa el compilador GCC.

“Estándares” de codificación en C++

Nombre de variables, funciones y estructuras

```
string table_name; // OK - uses underscore.  
string tablename; // OK - all lowercase.
```

```
struct UrlTableProperties {  
    string name;  
    int num_entries;  
};
```

“Estándares” de codificación en C++

Estructura condicional

```
if (condition) { // no spaces inside parentheses
    ...
} else if (...) { // The else goes on the same line as the closing brace.
    ...
} else {
    ...
}
```

Short conditional statements may be written on one line if this enhances readability. You may use this only when the line is brief

```
if (x == kFoo) return new Foo();
if (x == kBar) return new Bar();
```

This is not allowed when the if statement has an else:

```
// Not allowed IF statement on one line when there is an ELSE clause
if (x) DoThis();
else DoThat();
```

“Estándares” de codificación en C++

Estructuras repetitivas

Braces are optional for single-statement loops.

```
for (int i = 0; i < kSomeNumber; ++i)
    printf("I love you\n");

for (int i = 0; i < kSomeNumber; ++i) {
    printf("I take it back\n");
}
```

```
while (condition) {
    // Repeat test until it returns false.
}
```

Problemas

[HackerRank - Input and Output](#)

[HackerRank - Conditional Statements](#)

[HackerRank - Compare the Triplets](#)

[HackerRank - For Loop](#)

[HackerRank - Array Introduction](#)

[HackerRank - Strings](#)

Referencias

- ❑ Steve Oualline - **Practical C++ programming**
- ❑ HackerEarth - **The Build Process-C/C++**
- ❑ Google - **Google C++ Style Guide**

¡ Good luck and have fun !