

Introducción a la Programación Competitiva

Rodolfo Christian Catunta Uturunco (Elemental Bolivia)



19 de Febrero de 2025

Contenido

- 1 Presentación
- 2 Jueces en Línea
- 3 IDEs
- 4 Compilación y Ejecución externa
- 5 Estructuras de Datos Estáticas Simples
- 6 Impresión y Lectura Rápida

Contenido

- 1 Presentación
- 2 Jueces en Línea
- 3 IDEs
- 4 Compilación y Ejecución externa
- 5 Estructuras de Datos Estáticas Simples
- 6 Impresión y Lectura Rápida

- Horario
 - Lunes, Miércoles y Viernes 18:30 a 20:00
- Módulos: 8 (cada módulo de 12 clases)
- Nivel: Intermedio - Introducción a Avanzado
- Herramientas
 - GitHub - Repositorio de Código
 - Presentaciones y Apuntes
 - Cuestionarios Teóricos (e.g. Google Forms, Quizziz, Quizlet)
 - Concursos en Juki o VJudge
- Libros
 - Programación Competitiva 3/4 - Steven & Felix Halim
 - Introducción en C++ a la Programación Competitiva - ICPC UMSA
 - Introduction to Algorithms - Thomas H. Cormen - MIT
 - Competitive Programmer's Handbook - Antti Laaksonen

Reglas

- 1 Se debe tener la cámara encendida todo el tiempo que dure la clase.
- 2 Responder rápidamente cuando el instructor pregunte algo. Si no se responden 3 veces durante la clase, se tomará como falta del estudiante.
- 3 Tomar las clases a través de una laptop o PC de escritorio, no se permiten tablets o smartphones, salvo para conectarlos como cámara.
- 4 En cuanto el profesor lo indique deben compartir la pantalla. Si no se comparte pantalla 3 veces durante la clase, se tomará como falta del estudiante.
- 5 Si es que se tiene algún percance durante la clase (ej. Llamada de un padre de familia, ida al baño) en el que el estudiante se tenga que retirar momentáneamente de la clase (no más de 5 minutos) esta se debe notificar al instructor mediante un mensaje privado de Zoom.
- 6 Prohibido enviar código generado por IA al juez virtual, si se descubre se hará la llamada de atención correspondiente.

Contenido

- 1 Presentación
- 2 Jueces en Línea
- 3 IDEs
- 4 Compilación y Ejecución externa
- 5 Estructuras de Datos Estáticas Simples
- 6 Impresión y Lectura Rápida

Los Jueces en línea son importantes para la práctica y desarrollo de la programación competitiva. Estos son los jueces donde deben tener cuentas para el desarrollo del curso:

- Codeforces
- Codechef
- VJudge
- OmegaUp
- OnlineJudge
- Juki Judge
- AtCoder

Contenido

- 1 Presentación
- 2 Jueces en Línea
- 3 IDEs**
- 4 Compilación y Ejecución externa
- 5 Estructuras de Datos Estáticas Simples
- 6 Impresión y Lectura Rápida

- Un IDE es un Entorno de Desarrollo Integrado, que permite desarrollar código en algún lenguaje de programación. Existen IDEs generales y específicos.
- Los IDEs específicos se encargan de especializarse en el desarrollo en cierta tecnología o lenguaje de programación, como por ejemplo:
 - Code::Blocks
 - Dev-C++
 - CLion
- Los IDEs generales, son versátiles y permiten trabajar en distintos lenguajes de programación, algunos son por ejemplo:
 - Sublime Text
 - Visual Studio Code
 - Emacs

Contenido

- 1 Presentación
- 2 Jueces en Línea
- 3 IDEs
- 4 Compilación y Ejecución externa**
- 5 Estructuras de Datos Estáticas Simples
- 6 Impresión y Lectura Rápida

Compilación y Ejecución Externa

Compilación

Es la etapa en la que el código escrito, en un lenguaje de programación compilado, pasa a ser escrito en lenguaje de máquina. En el caso de C++ esta operación permite generar un archivo .exe. En esta etapa el **compilador** puede detectar errores de sintaxis en el código.

Ejecución

Es la etapa en la que el código es ejecutado. Este puede presentar distintas alternativas, desde un archivo determinado de entrada o un archivo donde se guardará la salida.

Compilación

```
C:\Users\USER>cd Desktop
```

Ingresar a la ubicación del archivo

```
C:\Users\USER\Desktop>g++  
g++: fatal error: no input files  
compilation terminated.
```

Revisar que se tenga el compilador
instalado

```
C:\Users\USER\Desktop>g++ ejemplo.cpp -o ejemplo.exe
```

Comando de Compilación

```
C:\Users\USER\Desktop>g++ ejemplo.cpp -o ejemplo.exe  
ejemplo.cpp: In function 'int main()':  
ejemplo.cpp:6:7: error: 'num' was not declared in this scope  
    cin>>num;  
      ^  
ejemplo.cpp:8:2: error: expected ';' before 'cout'  
    cout<<num<<endl  
    ^
```

Errores de
Compilación

Ejecución

```
C:\Users\USER\Desktop>ejemplo.exe  
100  
Hola Creadores Digitales Senior  
100
```

Ejecución Normal

```
C:\Users\USER\Desktop>ejemplo.exe < archivo.in > archivo.out
```

Ejecución mediante
archivos de
entrada y salida

Contenido

- 1 Presentación
- 2 Jueces en Línea
- 3 IDEs
- 4 Compilación y Ejecución externa
- 5 Estructuras de Datos Estáticas Simples**
- 6 Impresión y Lectura Rápida

Datos Primitivos en C++

1 byte = 8 bits

0 1 1 0 1 1 1 0

Nombre	Descripción	Tamaño	Rango de Valores	Uso
bool	Booleano	1 byte	Verdadero o Falso 1 0	Preguntas, Flags
char	Caracter (ASCII) a, 2, @ *, #	1 byte	256 caracteres	Letras, Comandos
int	Entero 49, 0, -23	4 bytes	De -2 mil millones hasta 2 mil millones (9 dígitos)	Números enteros, División entera, Sumatorias
long long	Entero Largo	8 bytes	Números de 18 dígitos	Números enteros largos, Factoriales, Conteos.
float	Número de punto flotante Puntos Reales	4 bytes	7 dígitos de precisión	Operaciones que no necesitan mucha precisión
double	Flotante con doble precisión	8 bytes	15 dígitos de precisión	Operaciones con precisión
long double	Flotante con doble precisión *	12 bytes	15 dígitos de precisión + 18 dígitos enteros	Operaciones con precisión y números grandes
string	Cadena No ES PRIMITIVO	Variable	Infinito *	Cadenas de Texto, Números de mas de 18 dígitos

Si usamos la palabra unsigned crearemos enteros sin signo.

Contenido

- 1 Presentación
- 2 Jueces en Línea
- 3 IDEs
- 4 Compilación y Ejecución externa
- 5 Estructuras de Datos Estáticas Simples
- 6 Impresión y Lectura Rápida**

printf y scanf

→ Es la forma antigua de imprimir y recibir datos

```
1 #include <iostream> // cin y cout
2 #include <cstdio> // scanf y printf
3
4 using namespace std;
5
6 int main(){
7     int version=4,anio;
8     // printf para imprimir
9     printf("Hola Creadores Digitales\n");
10    // Imprimir una variable
11    printf("Esta es la version %d \n",version);
12    // scanf para pedir datos
13    scanf("%d",&anio);
14    // Imprimir varias variables
15    printf("La version %d el anio %d \n",version ,anio);
16    return 0;
17 }
```

Identificadores de printf y scanf

Identificador	Salida
%c	Carácter o entero pequeño
%s	Tira de caracteres
%d, %i	Entero decimal
%u	Entero decimal sin signo
%lld	Entero largo
%llu	Entero largo sin signo
%f	Número de punto flotante en notación decimal
%lf	Número de punto flotante en notación decimal con doble precisión
%o	Entero octal sin signo
%x	Entero hexadecimal sin signo

cin y cout rápidos

```
1 #include <iostream> // cin y cout
2 using namespace std;
3
4 int main()
5 {
6     // Lineas para mejorar el tiempo de cin y cout
7     ios_base::sync_with_stdio(false);
8     cin.tie(NULL);
9
10    // ...Codigo ...
11    return 0;
12 }
```