

JULY 11, 2021 - 11H

AUGUST 8, 2021 - 16H

JUNE 15, 2021 - 15H

// ABRIL 2023

***|UTN**

LABORATORIO 1 - E11

TURNO MAÑANA

Martes 11 Abril 2023 - 10:30hs

JUNE 15, 2021 - 15H

MARCH 22, 2021 - 15H

CABECERA

`#include < nombre de fichero >`

`#include <iostream.h>`

En `#include<stdlib>` se encuentra la función `system("pause")` para dar una pausa al programa que se está ejecutando y observar bien los resultados

Ej. de otras librerías utilizadas

`#include <conio>` → función `getch()`..

`#include <string>` → para manipular cadenas

`#include <math>` → para operaciones numéricas

`#include <time>` → para usar la hora

`#include <stdio>` → para usar alternativas de entrada – salida como `printf` , `scanf`

FUNCIÓN cin cout



cin, toma caracteres de la entrada estándar (teclado)

cout, muestra caracteres en la salida estándar (pantalla)

#include <iostream>

Sintaxis:

cin>>variable1>>...>>variablen;

cout<<variable1<<variable2<<...<<variablen;

No olvidar que las cadenas de texto son variables y se ponen entre " " (comillas dobles).

Permite mostrar tanto cadenas de texto como valores de variables

FUNCIÓN system()



Permite ejecutar subprocessos o comandos del sistema operativo.

#include<stdlib.h>

Sintaxis:

```
system("pause");
```

```
system("cls");
```

VARIABLES



Se deberá especificar su tipo y su identificador.

El **tipo** especifica el tamaño en memoria asignado para el dato almacenado por la variable y el correcto modo de presentarlo.

El **identificador** es el nombre que tendrá la variable y que utilizaremos para referenciar en el programa.

VARIABLES - IDENTIFICADORES



C++ especifica una regla que debe seguir todo identificador

- Debe comenzar con una letra o un carácter de guión bajo(underscore).
- Debe continuar con una letra, un número o un carácter de guión bajo.
- No debe coincidir con una palabra reservada

VARIABLES - IDENTIFICADORES



Sintaxis:

<tipo de dato> <identificador>

Ejemplo

int contador ;

tipo identificador fin

Se pueden definir variables globales y locales.

Asignación - Constantes



Asignación:

La asignación es una instrucción por medio de la cual podremos modificar el contenido de una variable.

```
int nro_orden;    // Declaro una variable de tipo entero llamada nro_orden  
nro_orden = 1;    // Le asigno el valor 1
```


VARIABLES - IDENTIFICADORES



Sintaxis:

<variable> = <expresión>;

Ejemplo

operador

nro_orden = 100 ;

variable expresión fin

VARIABLES - IDENTIFICADORES



También es posible declarar una variable al mismo tiempo que le asignamos un valor;

Ejemplo

operador

```
int nro_orden = 100 ;
```

tipo variable expresión fin

CONSTANTES



Las constantes son muy similares a las variables, con la diferencia que a éstas sólo se les podrá dar un valor al momento de su declaración.

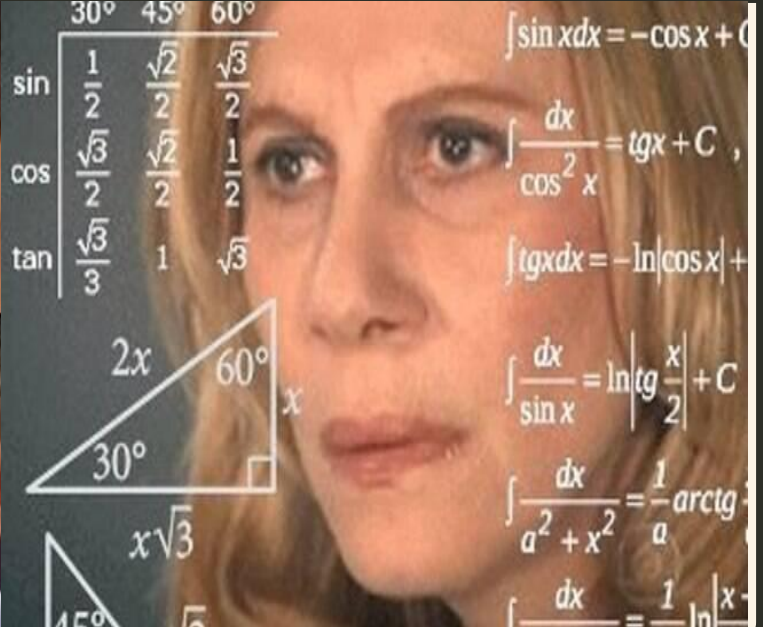
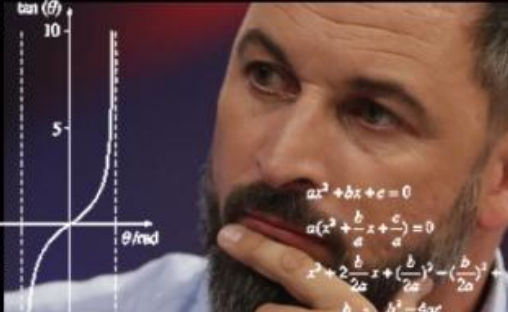
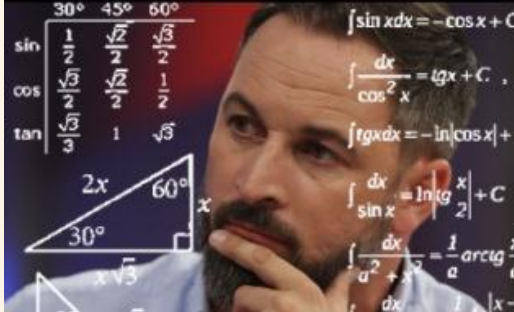
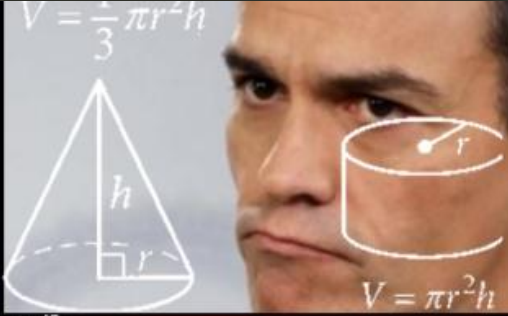
Luego cualquier intento de modificación será tomada como un **Error por parte del compilador**

Ejemplo:

const int max = 20; // (en el cuerpo del programa)

#define max 20 // (en archivos de cabecera)

EJERCICIO printf("CALCULAR AREA DE UN TRIANGULO");



EJERCICIOS PARA ENTREGAR



TAREA



Calcular Área del Triángulo

0 de 52 entregados **Vencimiento 21 de abril de 2023**

Grupos visibles



TAREA



EJERCICIO PARA CALCULAR - SUMA - RESTA - MULTIPLICACION - DIVISION

0 de 52 entregados **Vencimiento 25 de abril de 2023**

Grupos visibles



TAREA



Investigar y Responder

No publicado a los alumnos

Desarrollar un programa donde

- a) se utilice la siguiente función
system ("color 9F")
- b) ¿que efecto se logra?
- c) Pruebe las diversas variantes

0 de 52 entregados **Vencimiento 28 de abril de 2023**

Grupos visibles



EJERCICIOS PARA RESOLVER



EJERCICIOS BÁSICOS

1 - 3

**PARA ESTAR
AL DÍA**

MANUAL DE GUIA



<https://www.programarya.com/Cursos/C++>