|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NOMBRE DE LA PRÁCTICA** | **Prácticas de C ++** | | | | **No.** | | **1** |
| **ASIGNATURA:** | **Métodos númericos** | **CARRERA:** | **INGENIERIA EN SISTEMAS COMPUTACIONALES** | **PLAN:** | | **ISIC-2010-224** | |

**JOSÉ MARTÍNEZ HERNÁNDEZ 3042**

**I. COMPETENCIA(S) ESPECÍFICA(S): Aprender c++**

Insertamos el

estandar de entrada y salida

Estandar de bibliotecas

Iniciamos el método principal

Retornamos algo de nuestro método principal

Fin del método principal

**II. MATERIAL EMPLEADO:**

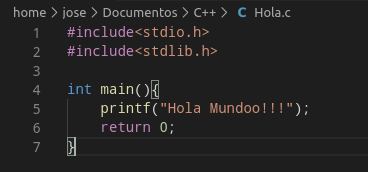
* Laptop con Linix Lite
* Editor de texto Visual Studio Code
* Terminal

**III.**

Mensaje a usuario

**DESARROLLO DE LA PRÁCTICA:**

**Práctica Número 1 “HOLA MUNDO”**

****

* **Guardamos el archivo como Hola.c**

****

* **Abrimos nuestra terminal en la ruta donde guardamos nuestro archivo y escribimos**

****

* **Luego para ejecutar nuestro .exe solo escribimos**



**PROGRAMA : Printf (print formated)**

Fin del método principal

Mensaje con salto de linea

* **Iniciamos escribiendo:**

Insertamos el

estandar de entrada y salida

Estandar de bibliotecas

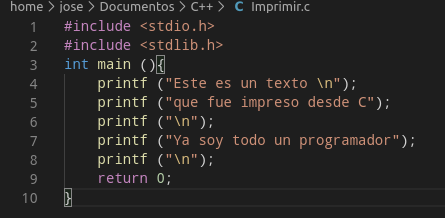
Iniciamos el método principal

Retornamos algo de nuestro método principal

Mensaje sin salto de linea

Salto de linea

Mensaje sin salto de linea

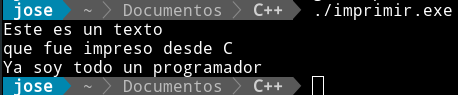
**Guardamos el archivo como Imprimir.c**

****

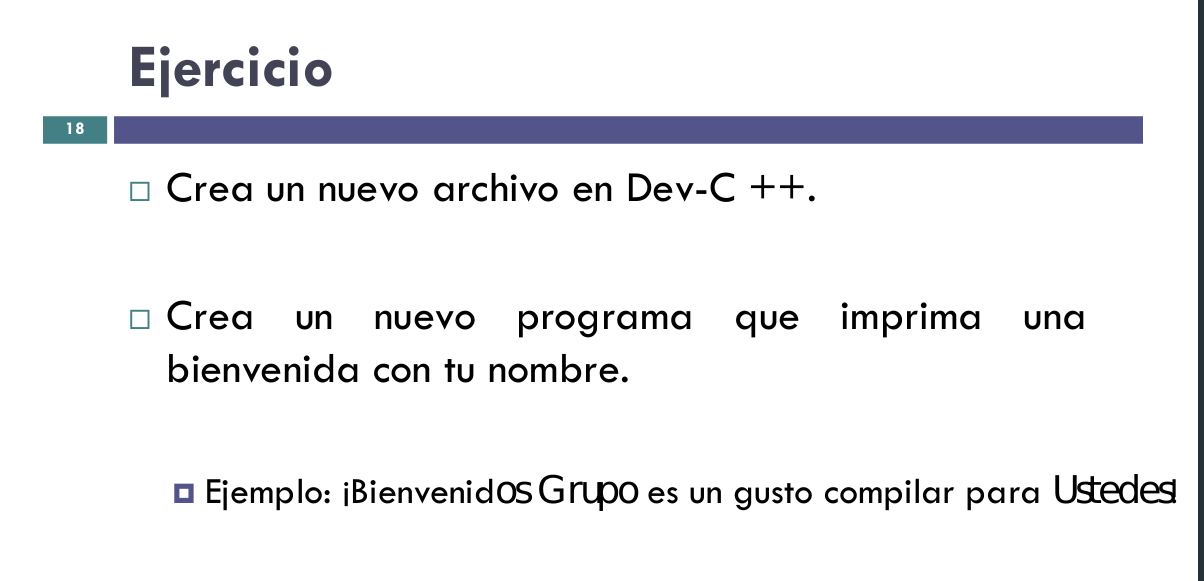
* **Abrimos nuestra terminal en la ruta donde guardamos nuestro archivo y escribimos**

****

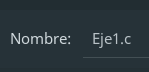
* **Luego para ejecutar nuestro .exe solo escribimos**

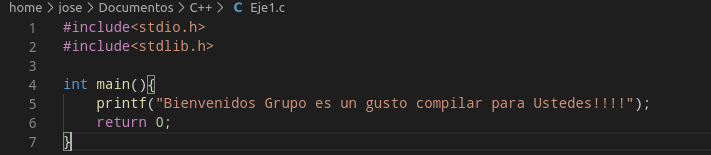
****

**Fin del programa**

****

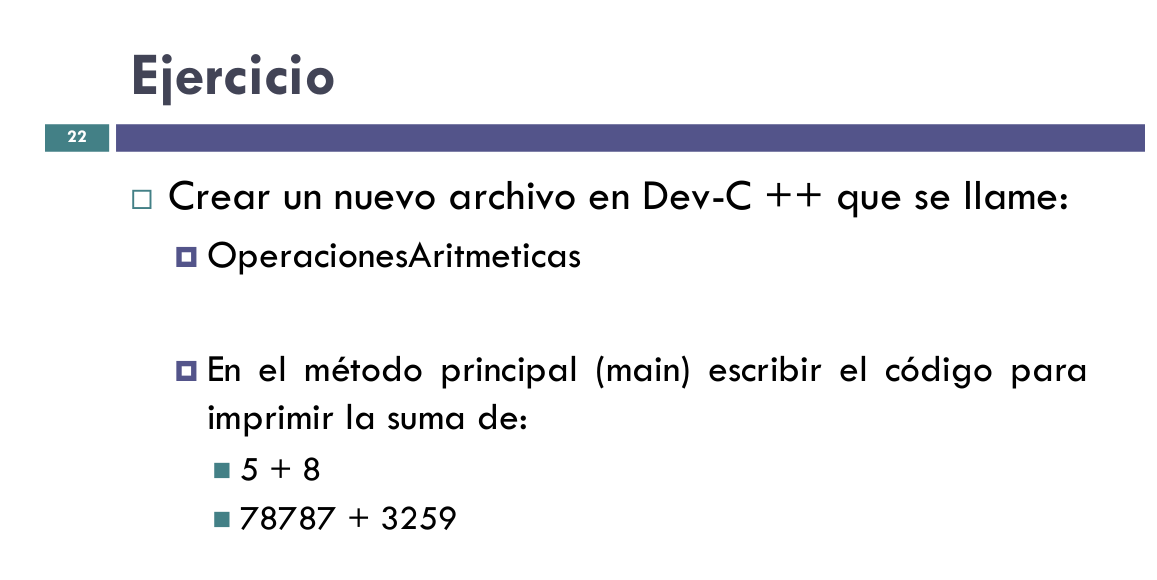
**Iniciaremos con guardando nuestro archivo como Eje1.c**

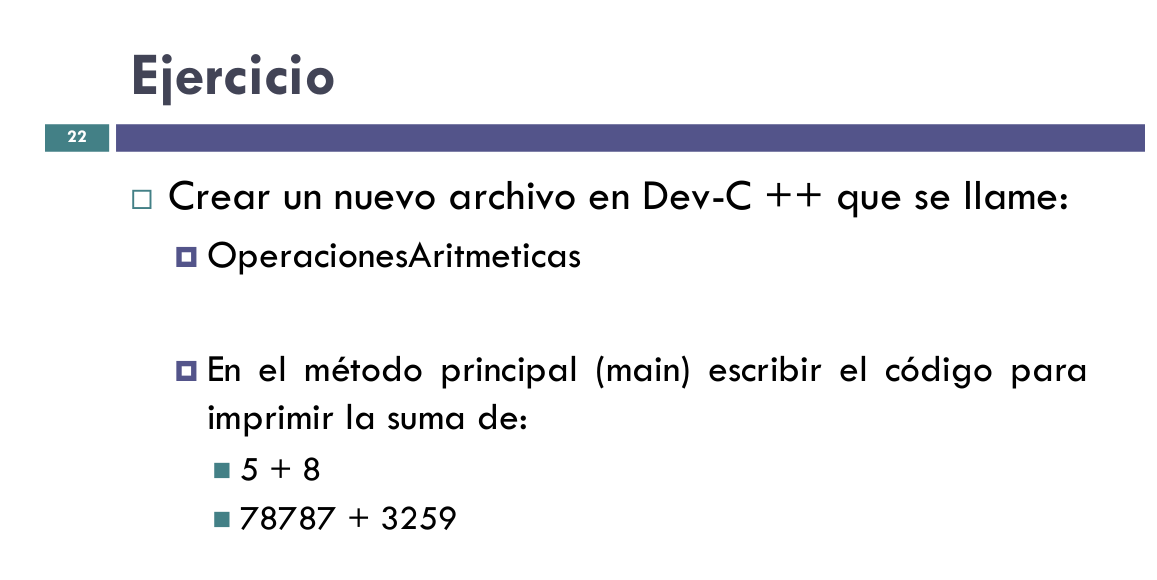
****

**Codificamos el ejercicio:**

**Compilamos y ejecutamos el programa :**

****

****

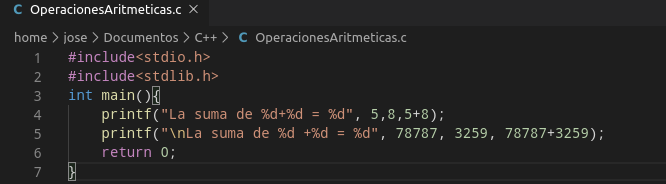
****

**Iniciaremos con guardando nuestro archivo como OperacionesAritmeticas.c**

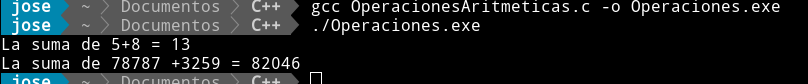
****

**Codificamos el ejercicio:**

* En este ejercicio vemos el %d que significa un \_\_\_ que sirve para decirle a un programa que cuando vea esto esperara una variable de tipo entero

****

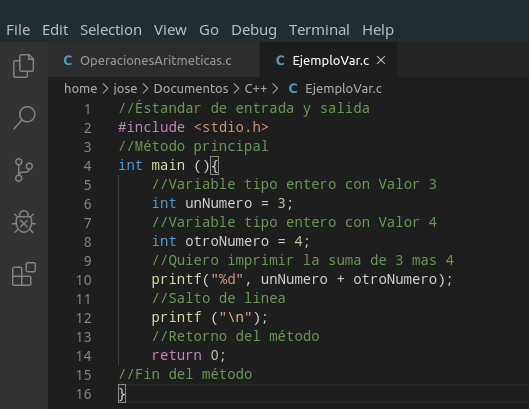
**Compilamos y ejecutamos el programa :**

****

* **Nota: el ; es utilizado para cargarlo a memoria**

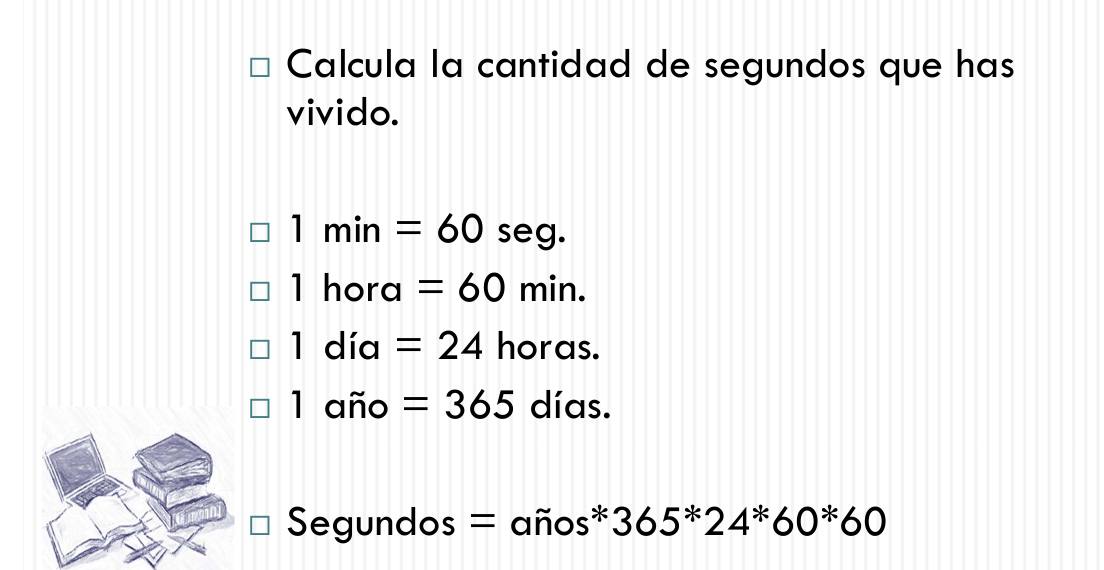
**Lamina 3 VARIABLES**

**Ejemplo numero 1 código**

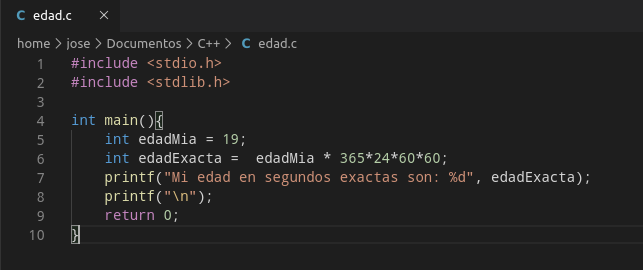


* Guardamos, codificamos y ejecutamos el ejemplo número 1:

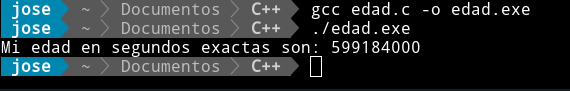


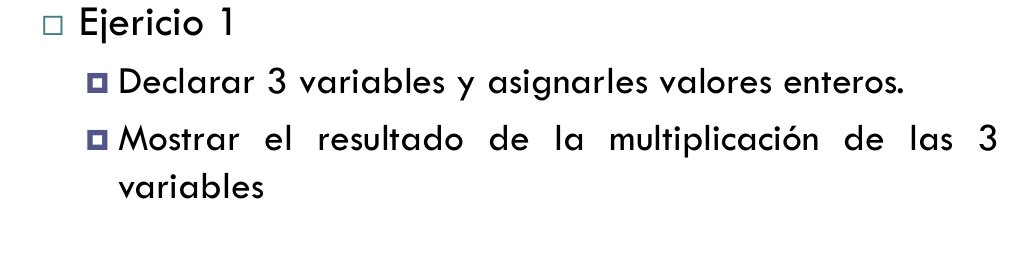


* Código del ejercicio

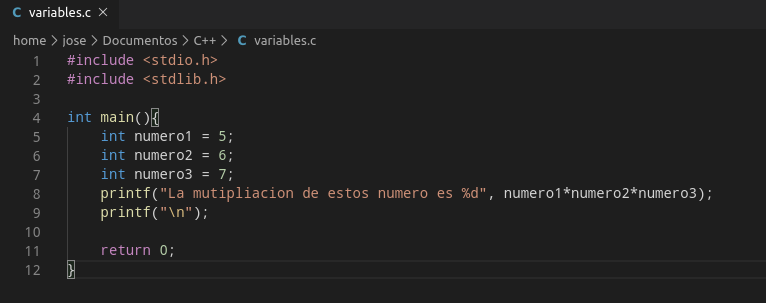


* Guardamos, codificamos y ejecutamos el ejercicio de edad



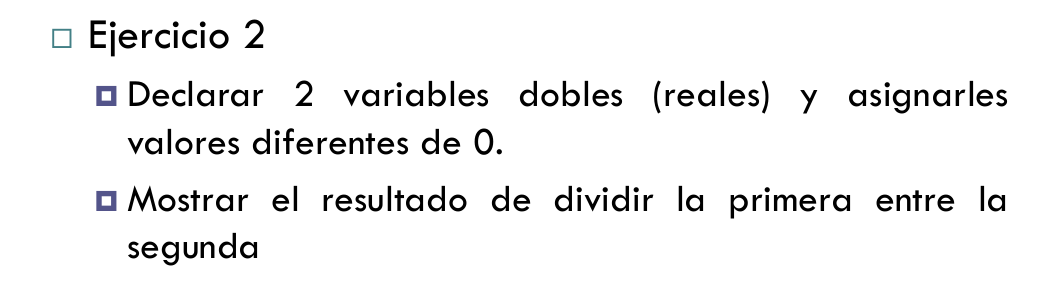


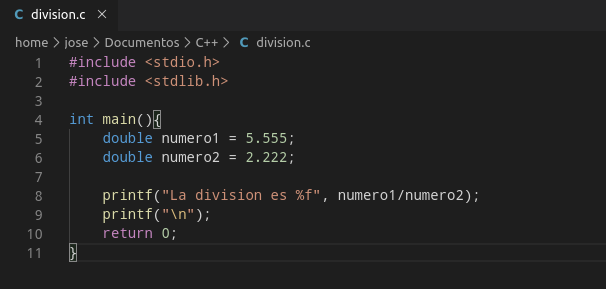
Código del ejercicio 1



Guardamos, compilamos y ejecutamos

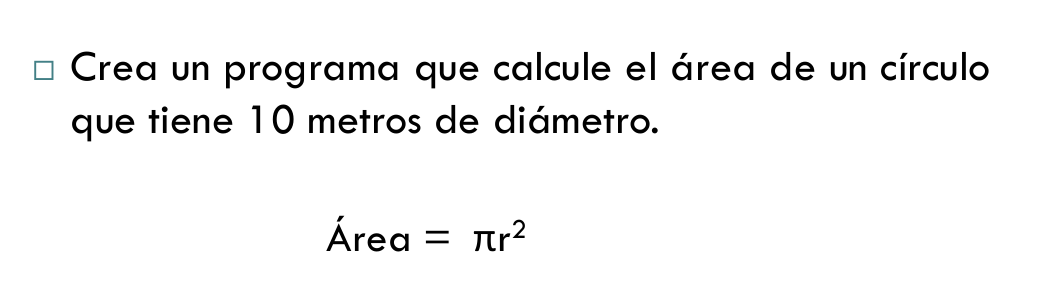


* Código de ejercicio 2

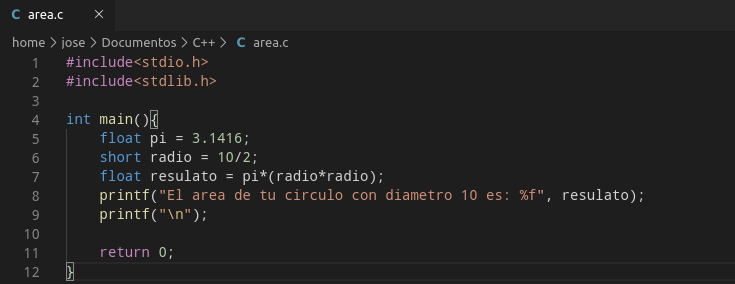


* Guardamos, codificamos y ejecutamos el ejercicio 2





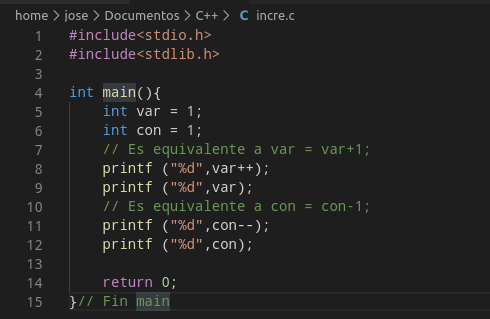
* Código de área del circulo:



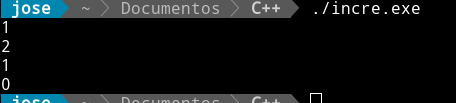
* Guardamos, codificamos y ejecutamos el ejercicio:



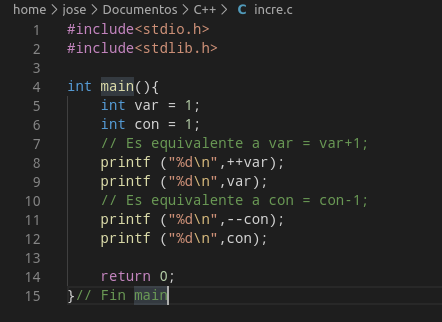
**Lamina 4 OPERADOR POST INCREMENTO/DECREMENTO**



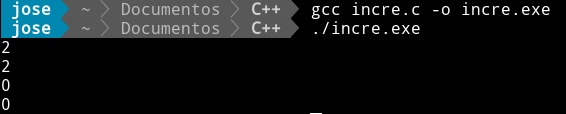
Guardamos, Compilamos y Ejecutamos el código:



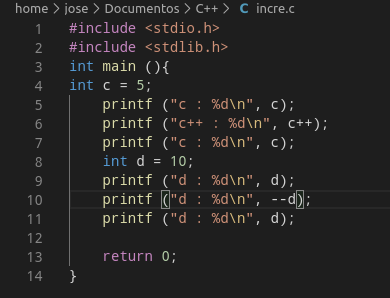
Vemos algo interesante en este código que depende mucho en donde pongas el signo de – o ++ ya que si lo pones después de variable primero cumple la instrucción anterior a esta y luego hace el decremento o incremento solicitado

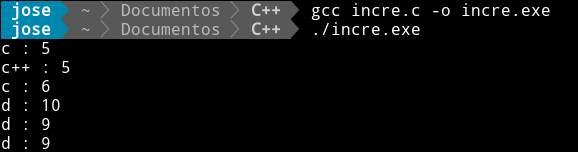


Guardamos, Compilamos y Ejecutamos el código:

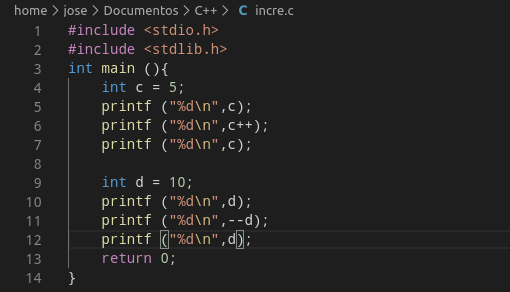


Aquí el caso es contrario porque primero hace el incremento y luego la instrucción, en este momento no se ve demasiado utilidad pero la tendrá aposteriori

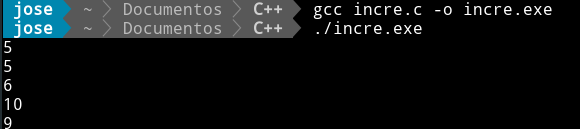


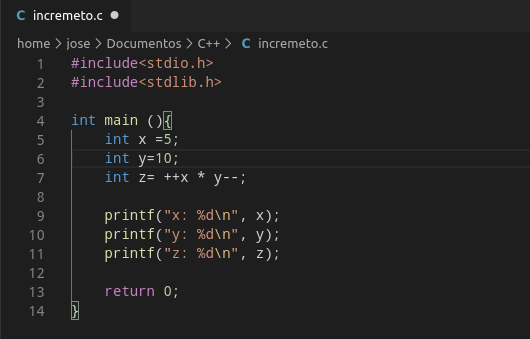
Guardamos, compilamos y ejecutamos

Aquí vemos la importancia del preincremento ya que ejecutara siempre primero el incremento o decremento antes de la instrucción



Guardamos, compilamos y ejecutamos:



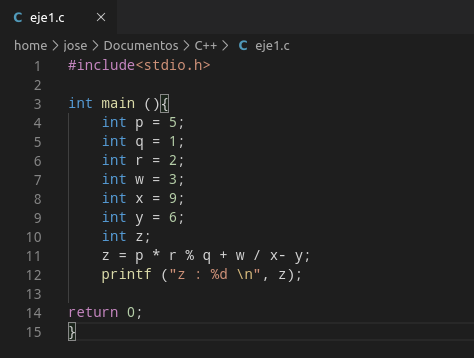


Guardamos, compilamos y ejecutamos



La explicación del porque la operación da 60 es por la razon del pre y pos incremento/decremento primero la x se le hace un preincremento por eso su valor es 6 y la y se le hace un postincremento lo cual su valor de 10 se conservera hasta la siguiente instrucción por eso da 60

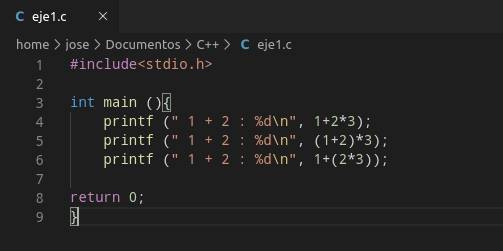
**Lámina 5 PRECEDENCIA DE OPERADORES**



El resultado apriori es -6 ya que primero se hace la multiplicación luego el se le sacara el residuo lo cual dará como resultado el 0, después sigue la división que da 0.33 pero como la variable es int solo guardara un 0 el cual se le restara el 6 por eso da -6

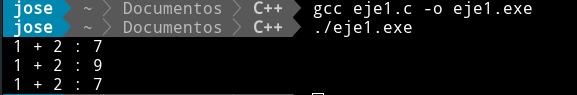
Guardamos, compilamos y ejecutamos

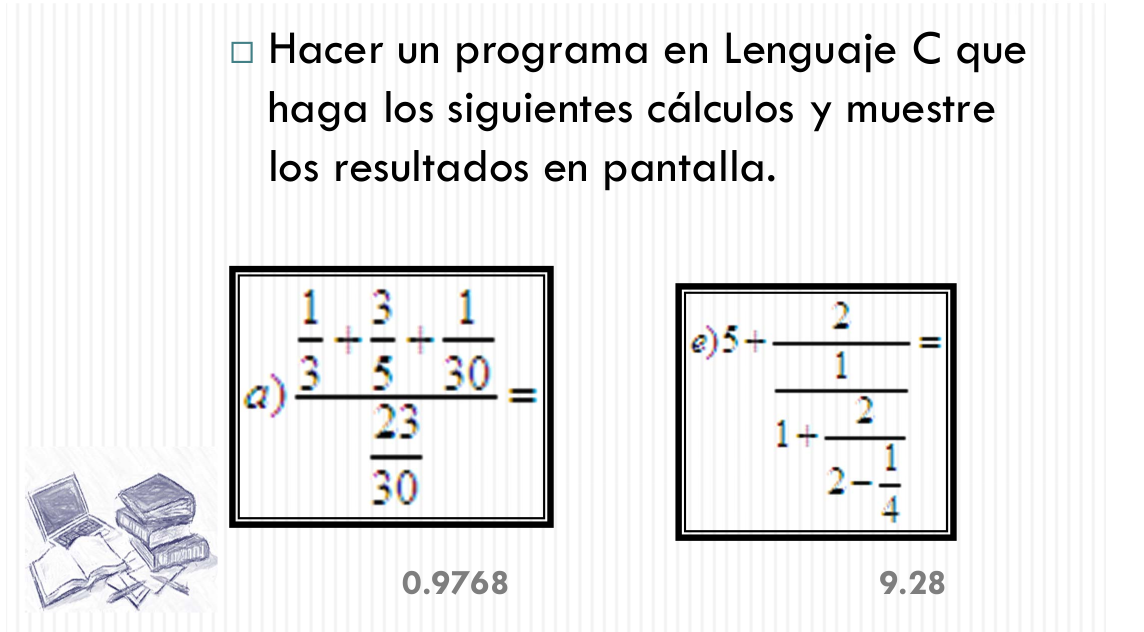




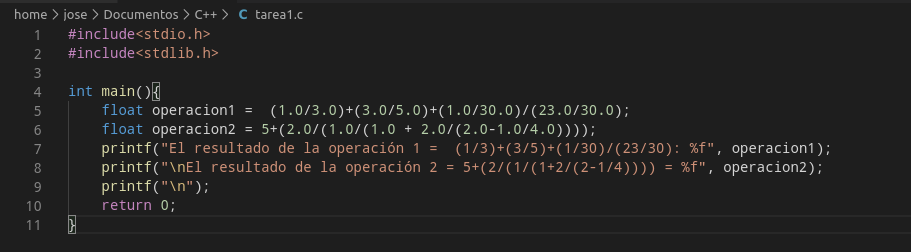
A priori la operación uno da 7 primero hará la multiplicación y luego la suma, la dos da 9 porque primero se hará la suma y luego la multiplicación la tercera dará como resultado 7 ya que aunque tenga el parentesis no sirve de nada porque aun asi se haria primero la multiplicación

Guardamos, compilamos y ejecutamos

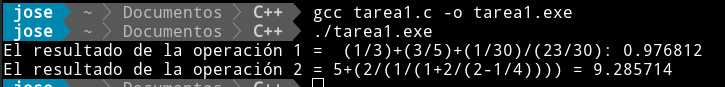


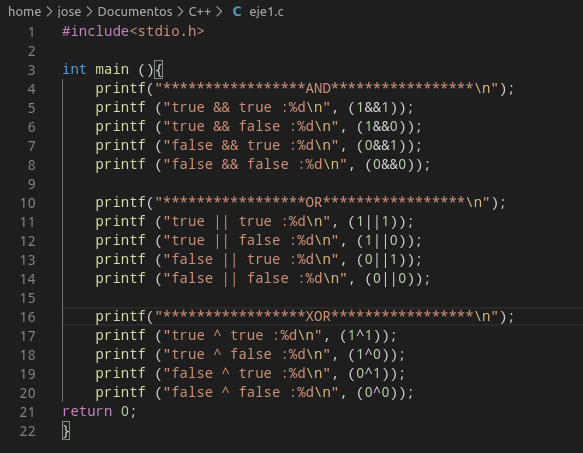


Código en c++



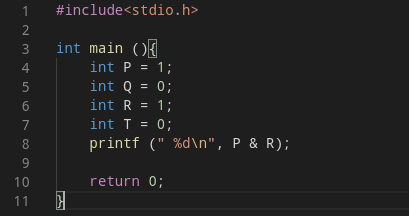
Guardamos, compilamos y ejecutamos



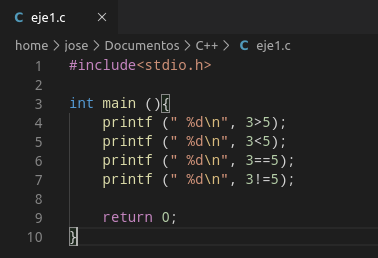


Guardamos, compilamos y ejecutamos

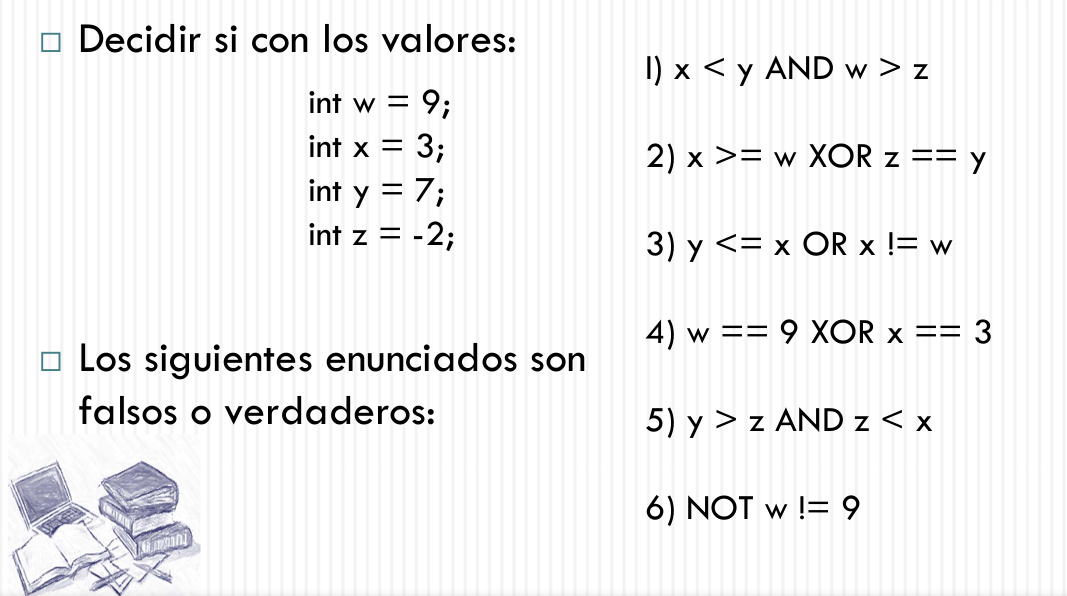




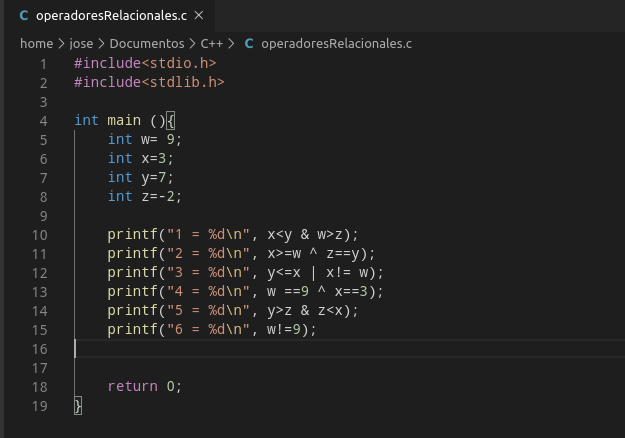
Guardamos, compilamos y ejecutamos



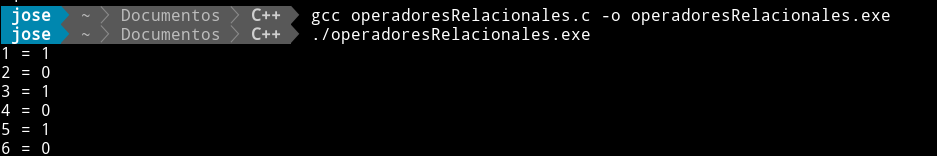
Guardamos, compilamos y ejecutamos



Código en c++



Guardamos, compilamos y ejecutamos



**Lamina 7 ENTRADA Y SALIDA DE DATOS**

En esta lamina nos dieron una nuevo sintaxis sobre el uso de una instrucción que es puts, esta sirve para solo escribir texto o un mensaje sin la necesidad de que poner el \n para salto de linea, esto nos ahorra 4 letras que en son como 2 segundos de escritura pero que en momentos de presión son vitales, el scanf también se trato en esta lamina que es sumamente importante ya que siempre o casi siempre hay que pedir datos al usuario, esta sintaxis solo permite que ingresemos un número y no un mensaje sobre esta instrucción

**IV CONCLUSION.**

En mi caso particular puedo decir que esto curso me esta pareciendo fantástico ya que las laminas vienen muy bien explicadas y con ejemplos de la vida real para aprender cosas básicas como el orden de los operadores aritméticos. También como funcionan la sentencia condicional del if y el else, c++ me parece un lenguaje demasiado bueno ya que el profesor nos ha comentado que se especializa en el manejo del hardware. El lenguaje c es muy parecidos a otros lenguajes en sintaxis. En conclusión me encanta programar en c y quiero seguir aprendiendo