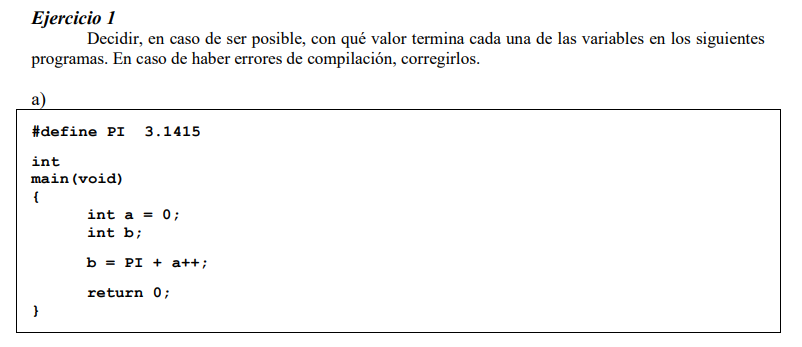
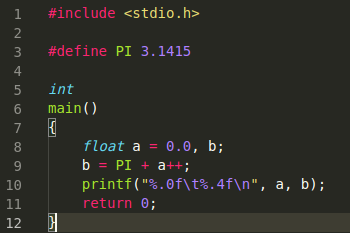
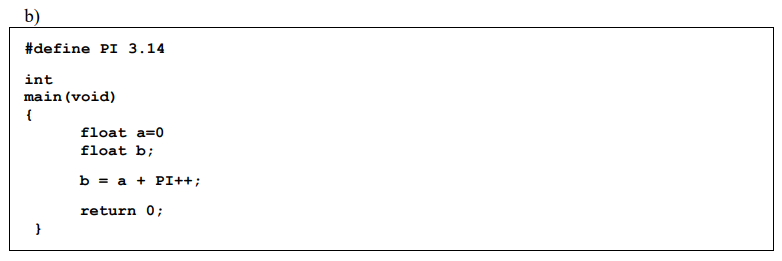
### Ejercicio 1

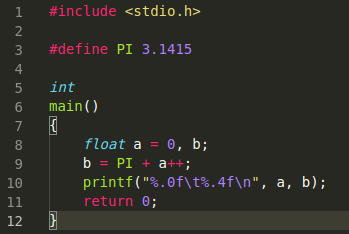


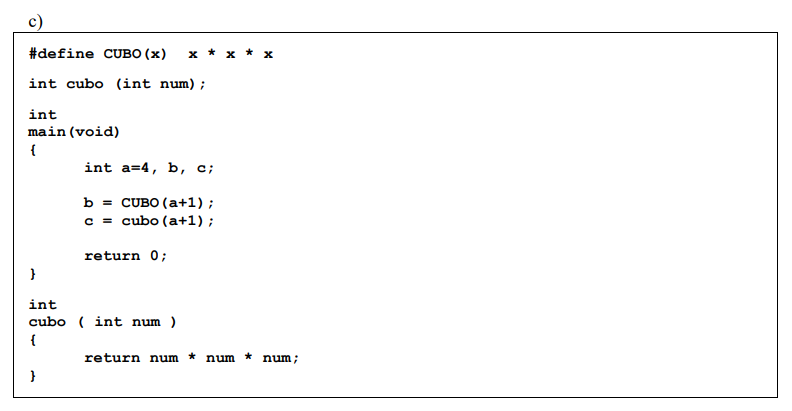
a: 1 b: 3. (b es tipo int, se pierden los decimales).



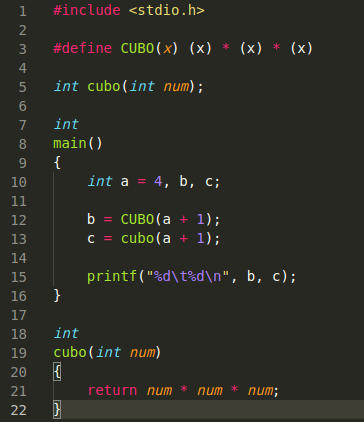


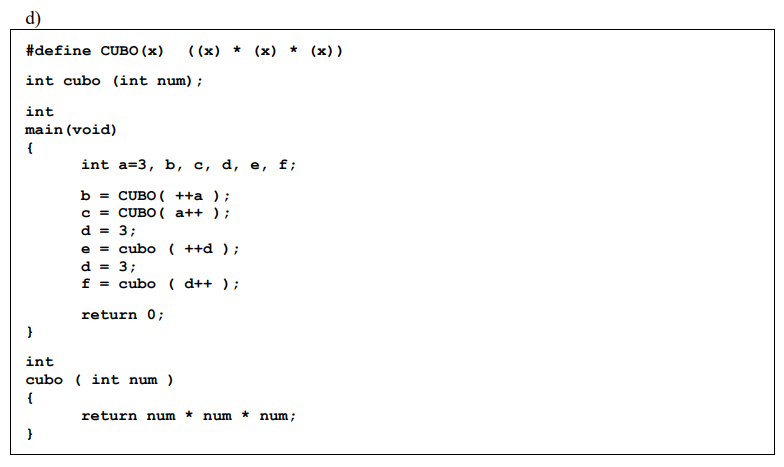
No compila porque está intentando incrementar una CONSTANTE. Las constantes no son un l-value.





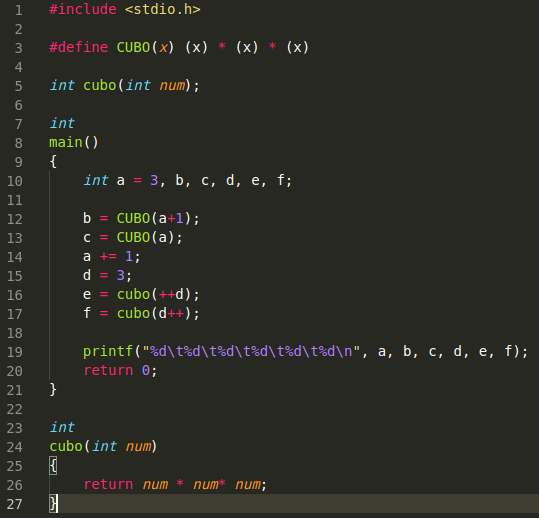
b: 13. c: 125.



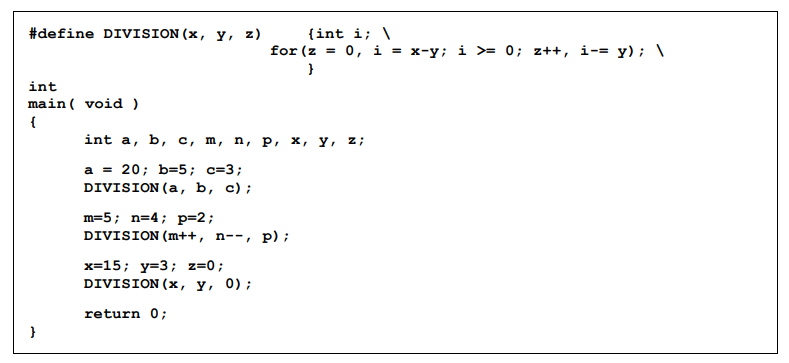


Las invocaciones a la macro son incorrectas, porque el postincremento se va a realizar tres veces en cada línea. No es aconsejable usar estos operadores en macros.

b: 4\*5\*6 = 120 c: 3\*4\*5 = 69 e: 64 f: 27



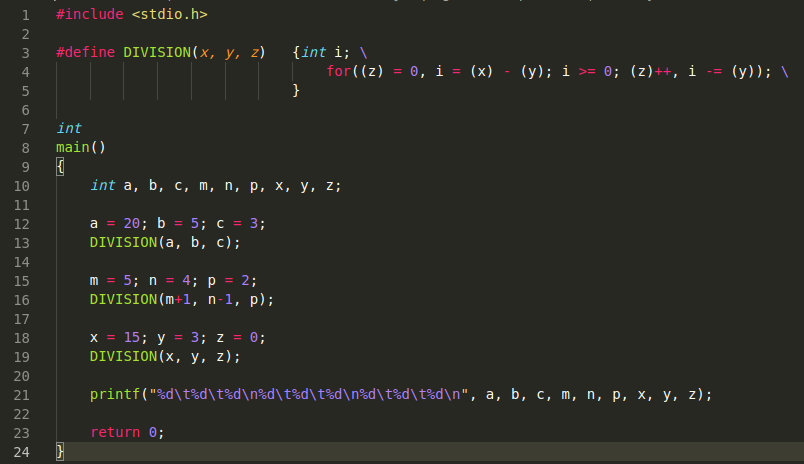




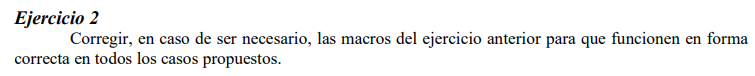
Primer invocación: a: 20 b: 5 c: 4

Segunda invocación: No es correcta. Al utilizar dentro de una macro operadores de incremento, no sabemos cuántas veces se utilizará el parámetro.

Tercera invocación: No se puede modificar una constante (0).

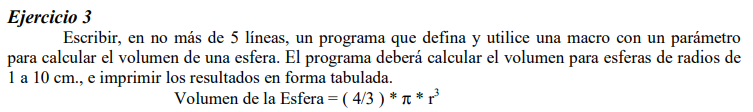


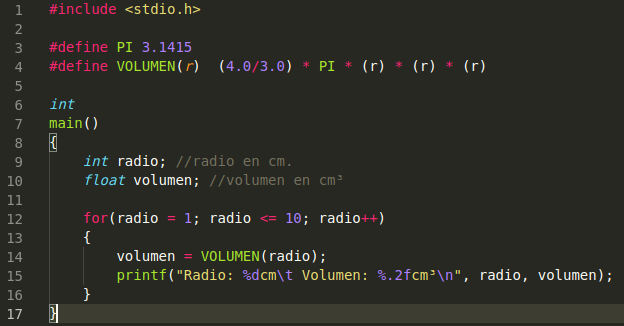
### Ejercicio 2

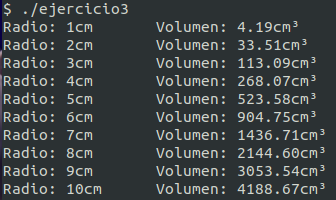


Corregidas en el ejercicio 1.

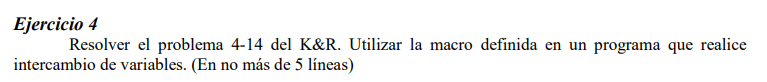
### Ejercicio 3



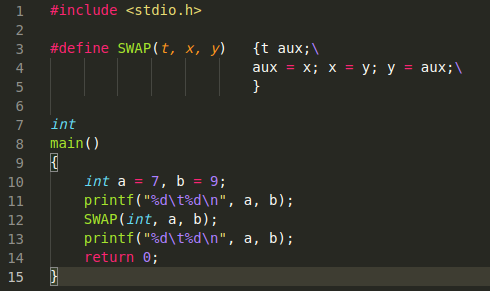




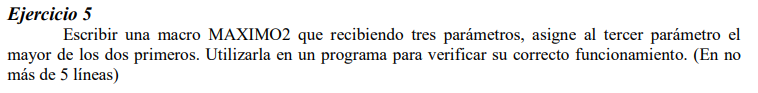
### Ejercicio 4

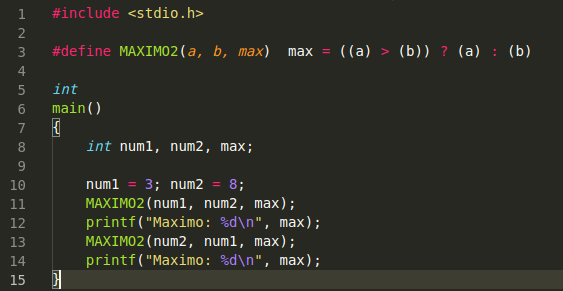




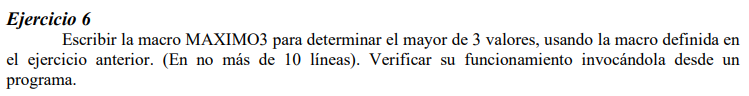


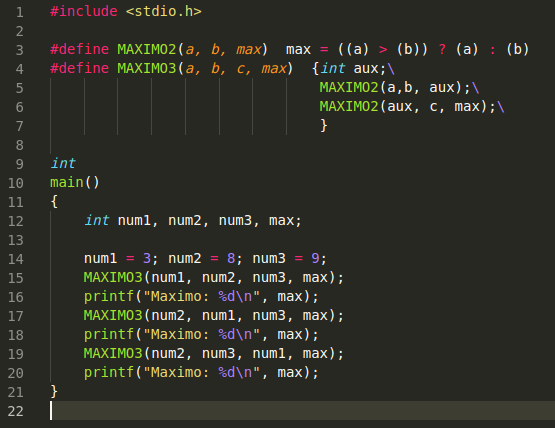
### Ejercicio 5



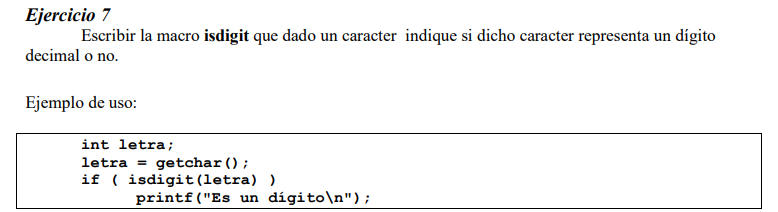


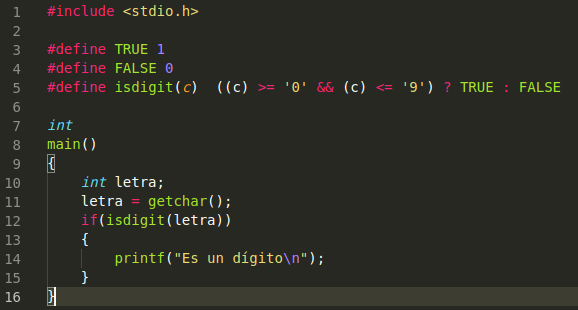
### Ejercicio 6



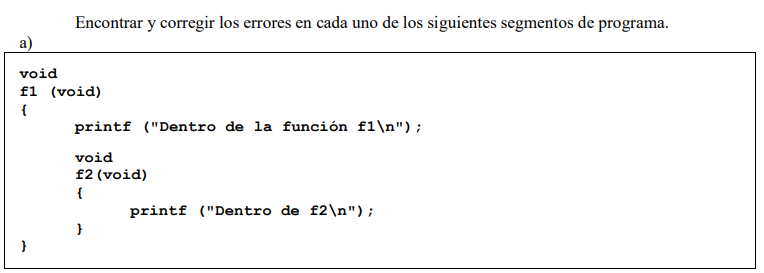


### Ejercicio 7

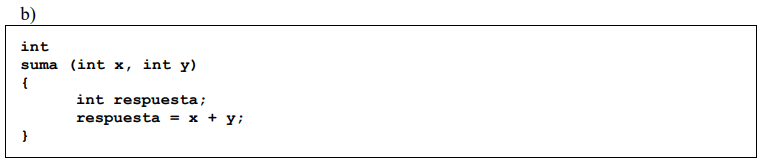




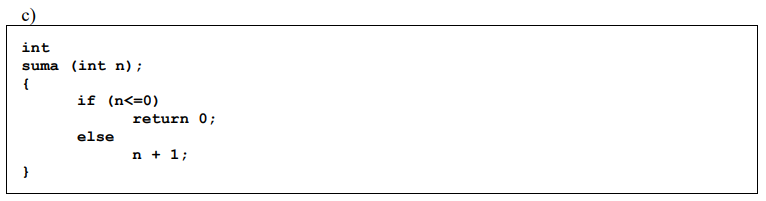
### Ejercicio 8



No se puede definir una función dentro de otra función.

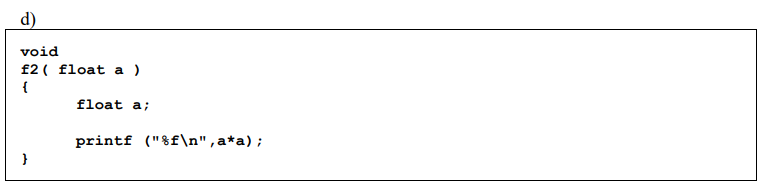


Falta el retorno de la función. Si no se retorna un valor, se pierde el resultado de la suma.

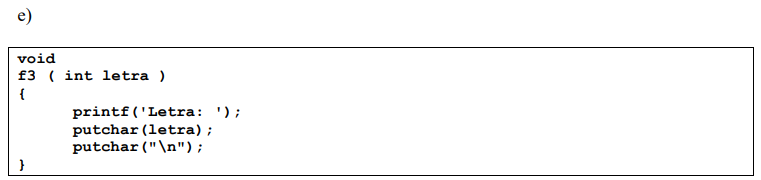


No se debe colocar ; en la línea dos.

Falta el retorno de la función para el caso else.



Se usando el mismo nombre para dos variables diferentes. No se puede usar ‘a’ como parámetro y redefinirla como otra variable.

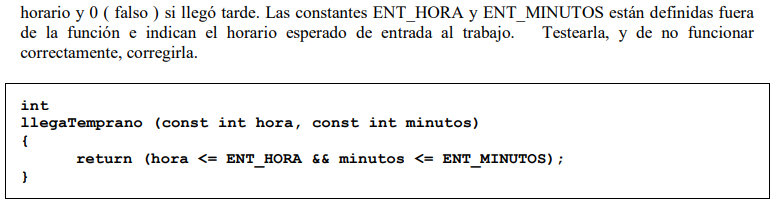


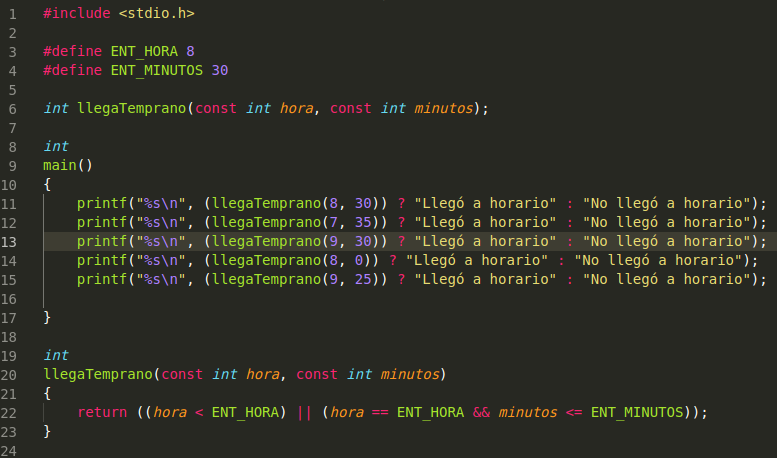
En el primer printf se debe usa “” (comillas dobles).

En el segundo putchar se debe usar ‘ ‘ (comillas simples).

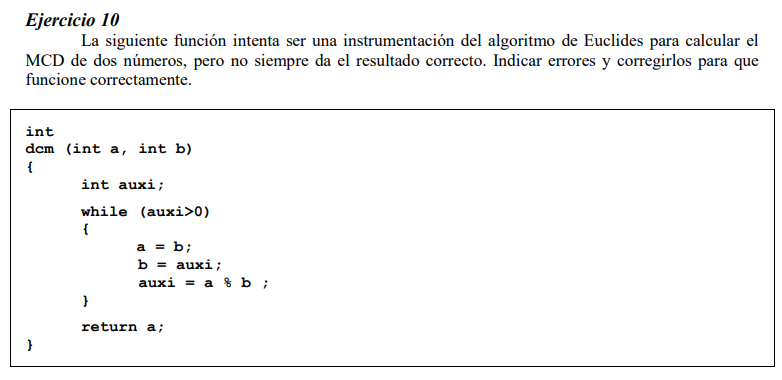
### Ejercicio 9



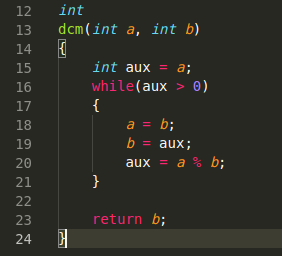




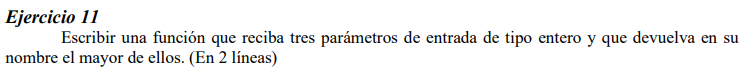
### Ejercicio 10

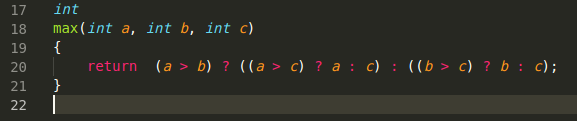


aux no está inicializada. El valor de retorno debe ser b.

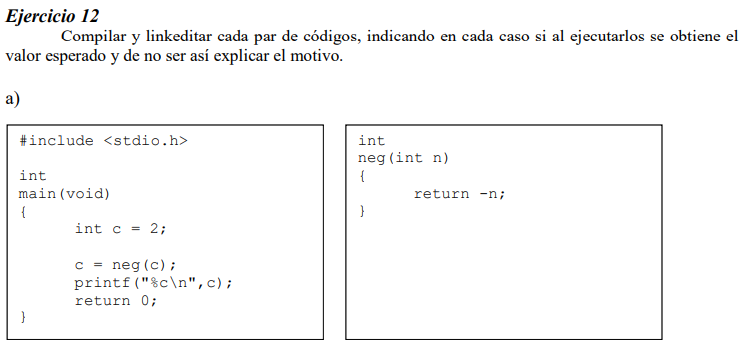


### Ejercicio 11

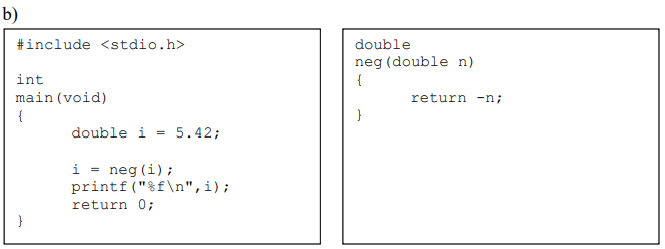




### Ejercicio 12



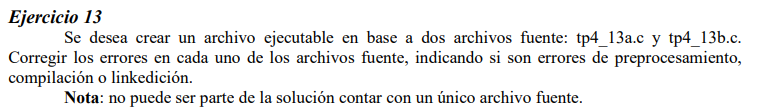
Se obtiene el valor esperado. Falta incluir el prototipo de la función, por lo que el compilador asume automáticamente tipo int. Funciona por casualidad.



No se obtiene el valor esperado. Al no prototipar la función en el archivo que contiene el main, o incluir el header con los prototipos, el compilador asume las funciones como int.

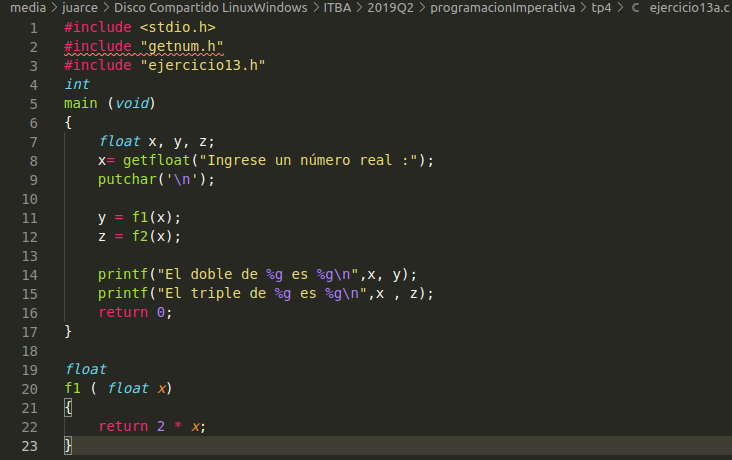
**Nunca se debe dejar librada la interpretación de tipos al default del compilador (Normas Indian Hill).**

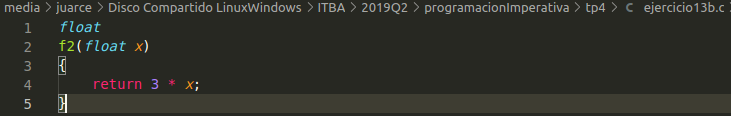
### Ejercicio 13

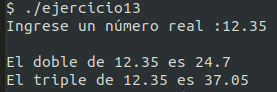


No se declaran getfloat, f1 y f2 antes de su uso. Se debe crear un archivo tp13.h que incluya el prototipo de las funciones. Se produce un error de compilación.







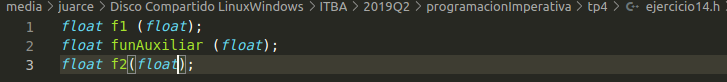


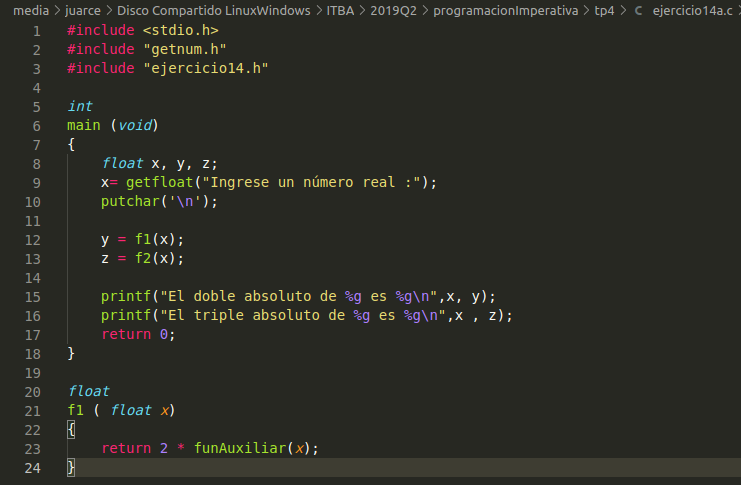
### Ejercicio 14

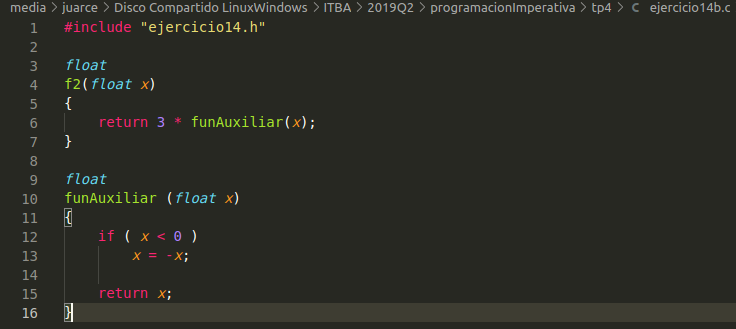


Error de linkedición: funAuxiliar está definida dos veces. Se debe eliminar una de las dos definiciones.

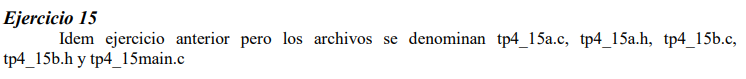
Error de compilación: Al invocar a “getfloat” y “f2”.







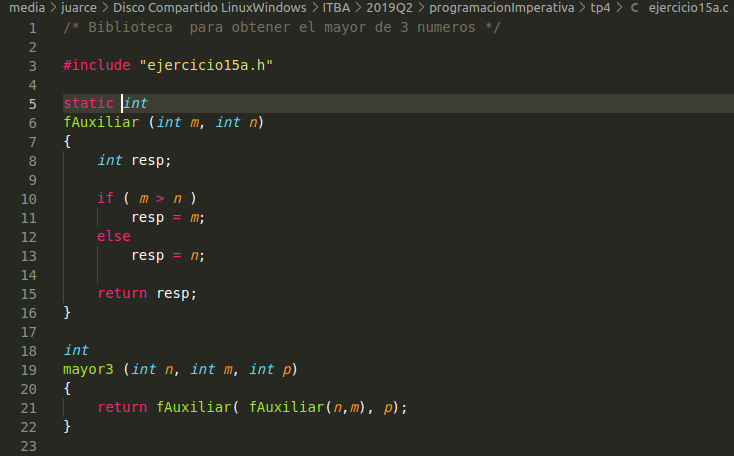
### Ejercicio 15

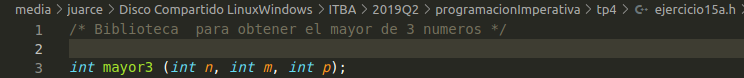


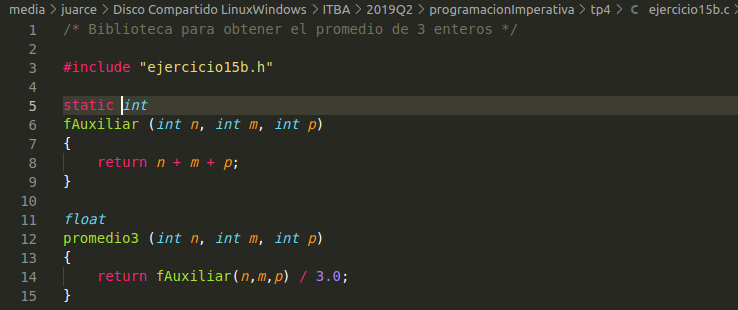
Error de conflicto de tipos: Al compilar el main, fAuxiliar está declarada más de una vez y con distinto prototipo.

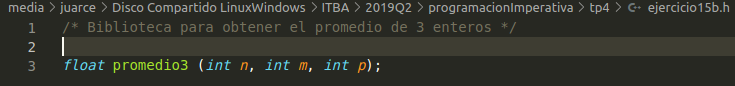
Error de linkedición: Múltiple declaración de fAuxiliar.

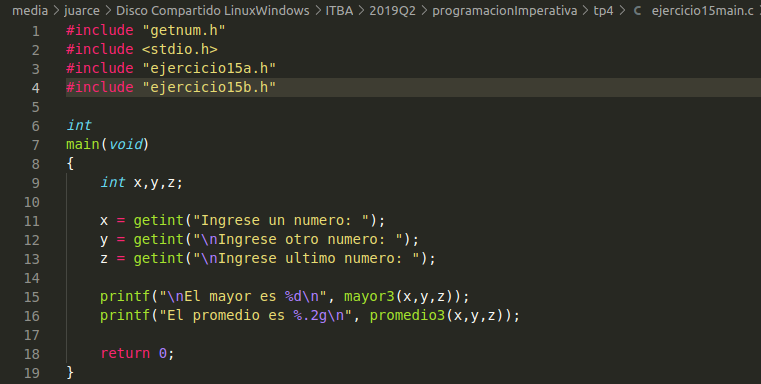
Declarar fAuxiliar como static y no incluirlos en los .h ya que hay dos funciones distintas con el mismo nombre.



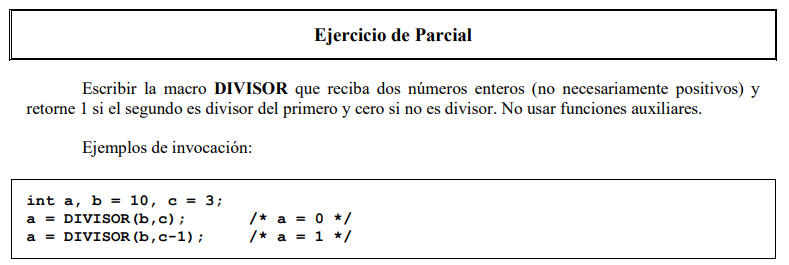


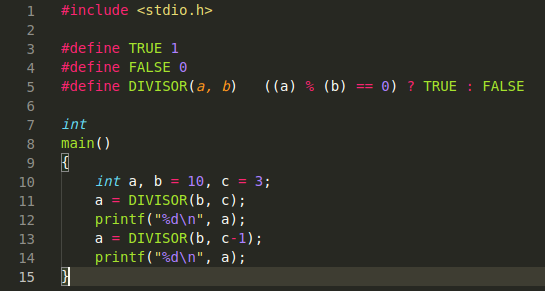






### Ejercicio 16





### Ejercicio 17

