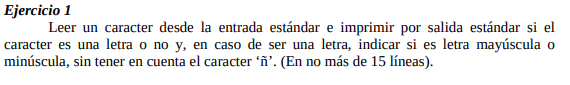
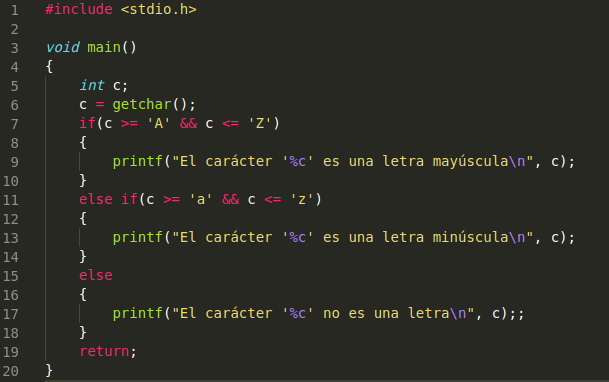
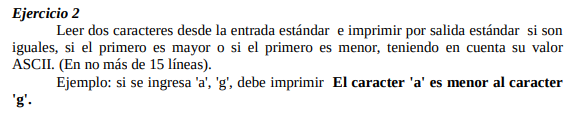
### Ejercicio 1

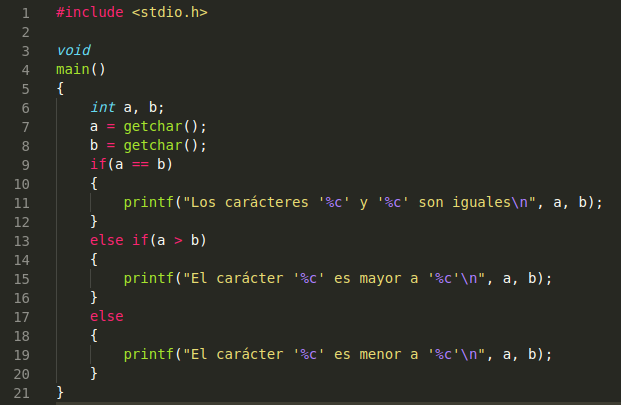




### 

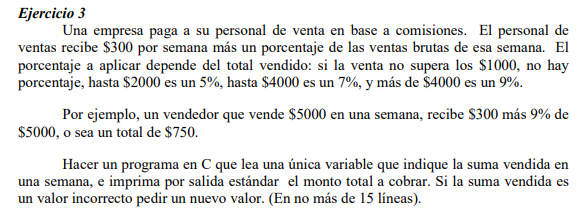
### Ejercicio 2

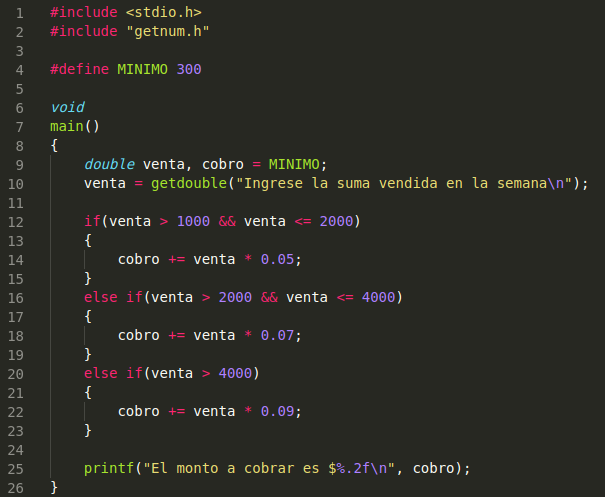




### 

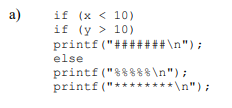
### Ejercicio 3

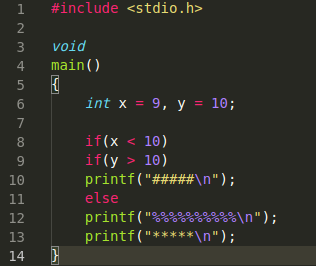




### Ejercicio 4

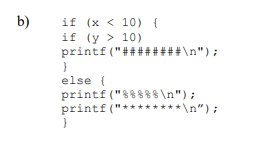
#### a

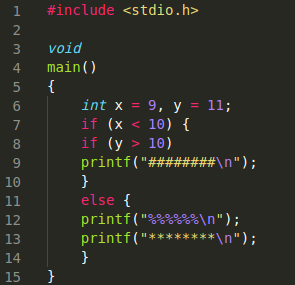






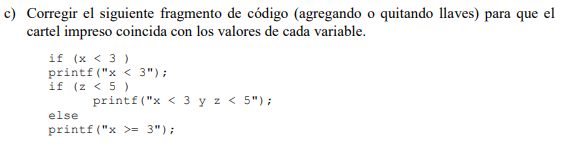
#### b

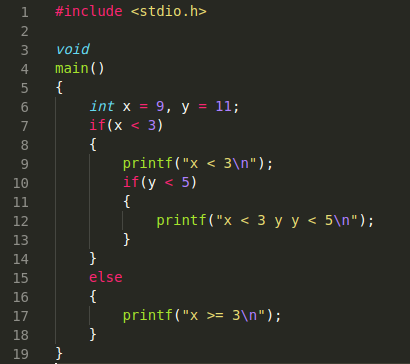






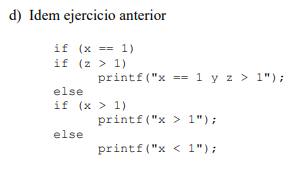
#### c







#### d

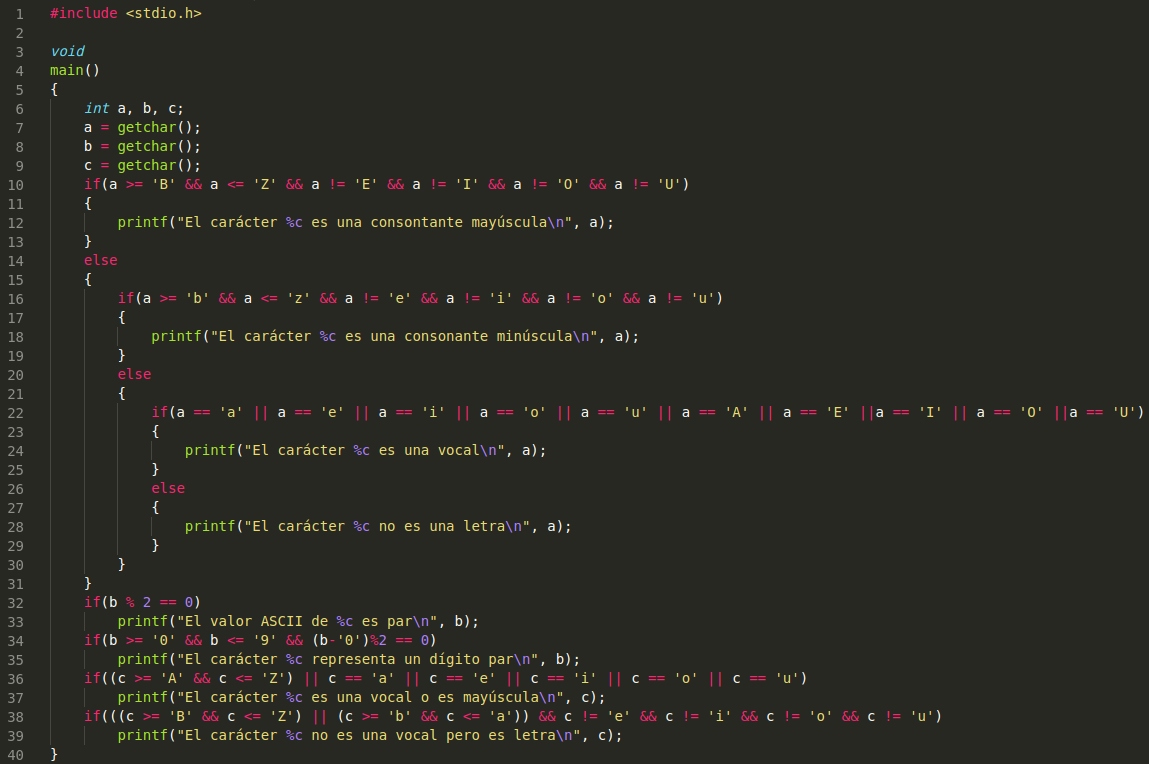


#### 

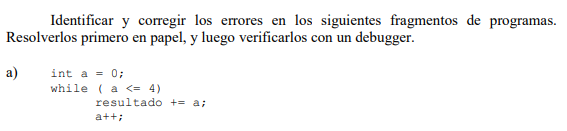


### Ejercicio 5



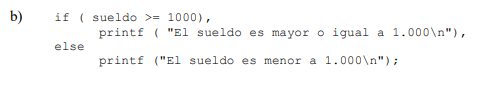


### Ejercicio 6



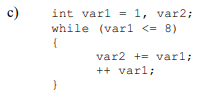
El programa se queda sumando ‘a’ a la variable ‘resultado’ porque al no haber un bloque, el while sólo ejecuta la instrucción inmediatamente inferior; con lo que nunca llega a incrementar la variable ‘a’. El indentado de líneas es sólamente visual, no produce ningún cambio en el programa.

Para que el programa se ejecute de acuerdo a lo deseado, las líneas debajo del while deben estar en un bloque entre llave {}.



No se deben colocar ‘,’ luego de una línea. Después de un if() se debe realizar un salto de línea.

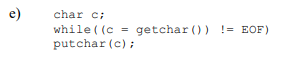
En el primer printf, al final se debe colocar ‘;’.



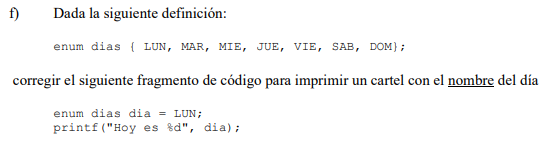
La variable var2 no fue inicializada. No se debe asumir que las variables se inicializan automáticamente en cero.



No se debe colocar el ; luego del while. Esto produce que el printf no se ejecute cuando debería.



Al tomar la variable c como char, ésta puede ser signed o no. Por lo tanto en caso de ser unsigned, nunca dará falsa la comprobación contra EOF (Ya que c no va a poder valer -1).



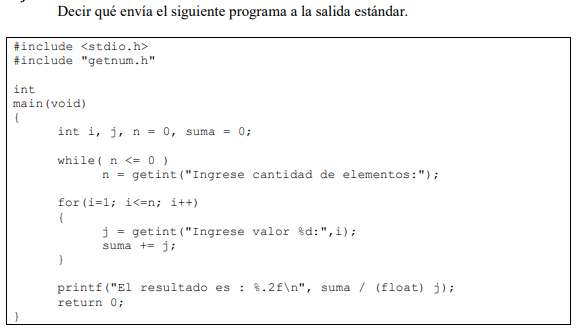
switch(dia)

case LUN: printf(“LUN”); break;

case MAR: printf(“MAR”; break;

….

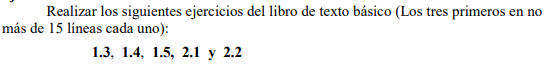
### Ejercicio 7



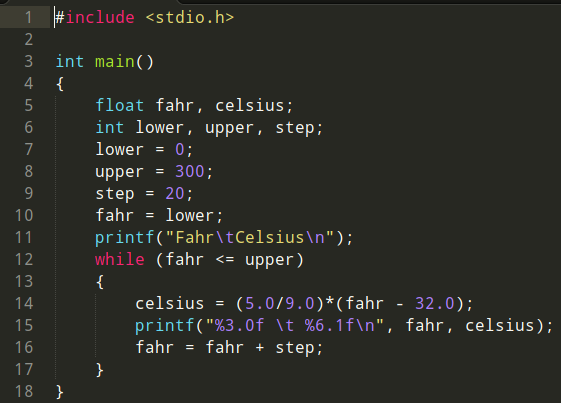
El programa suma n cantidad de números y a la suma la divide por el último número ingresado.

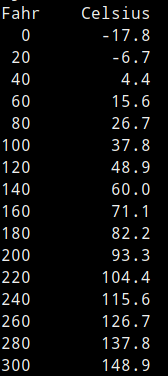
### 

### Ejercicio 8

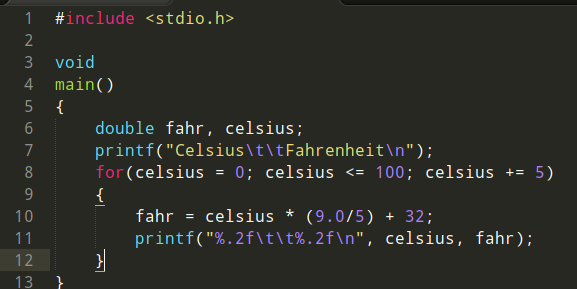


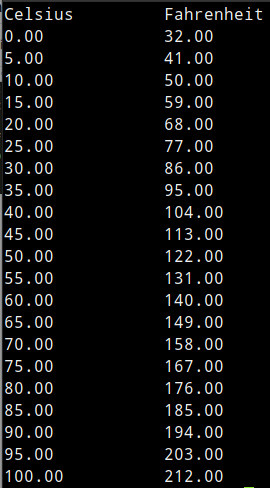




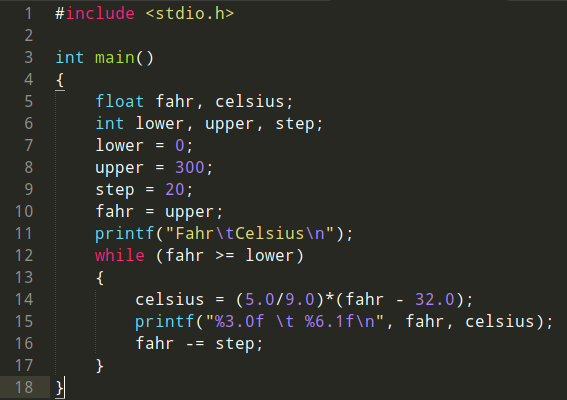


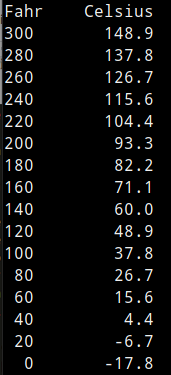


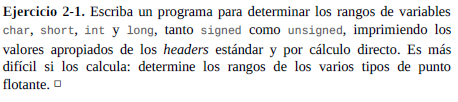




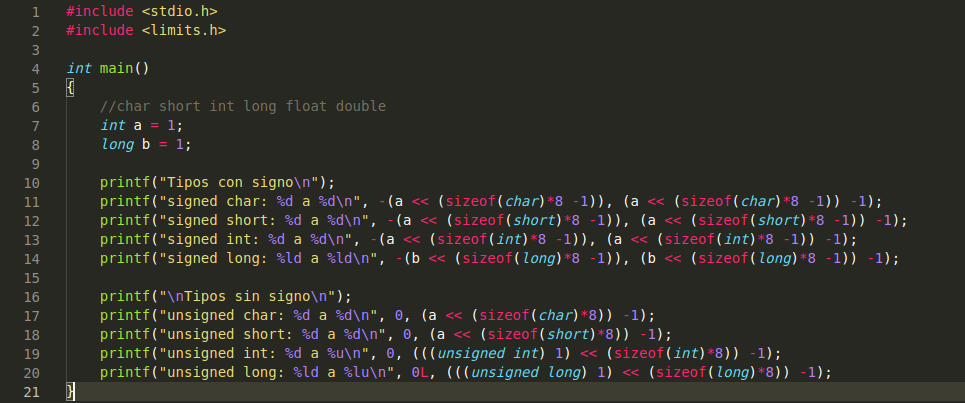






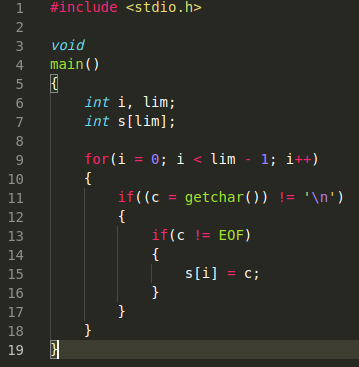


Los headers estándar que poseen los rangos de los diferentes tipos son **<limits.h>** y **<float.h>**.





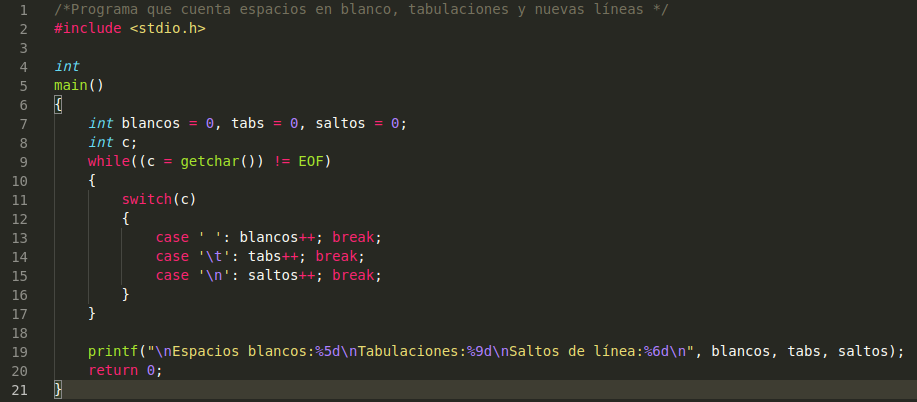


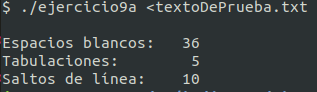


### Ejercicio 9





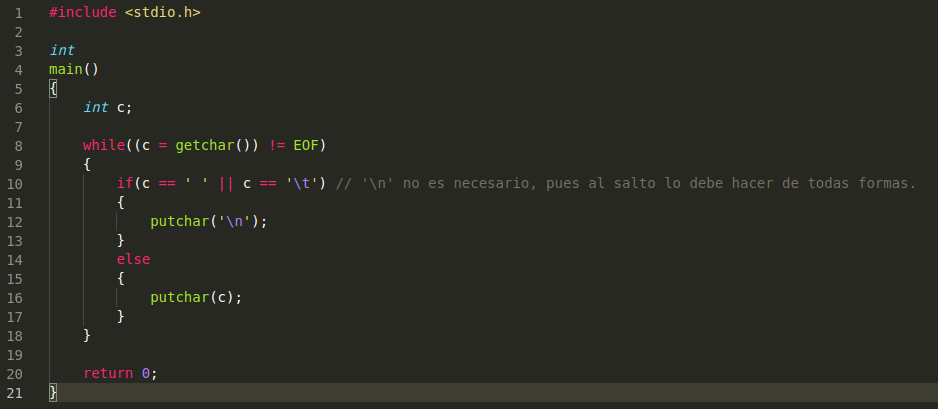




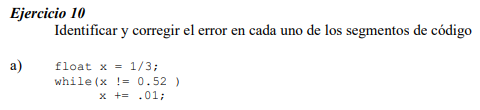


Para probar el programa de contar palabras, lo ejecutaría con un texto de dos párrafos, para que posea una extensión suficiente. Para descubrir errores es conveniente poner todos tipo de caracteres, espacios dobles, tabulaciones, saltos de línea.



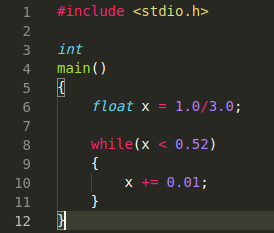


### Ejercicio 10



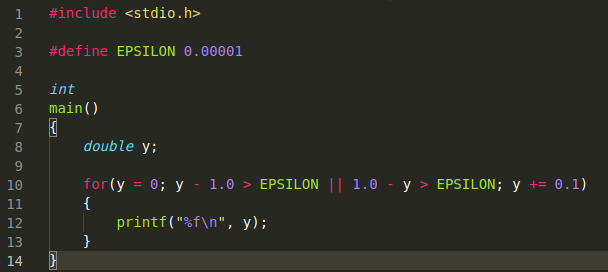
Realiza división entera y lo asigna a float.

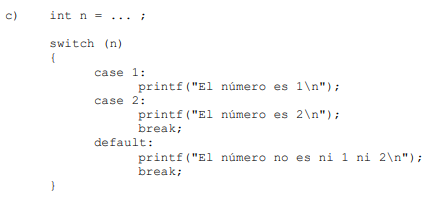
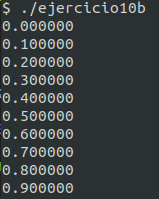
No se puede comparar directamente tipos de punto flotante por la forma en que se representan éstos.



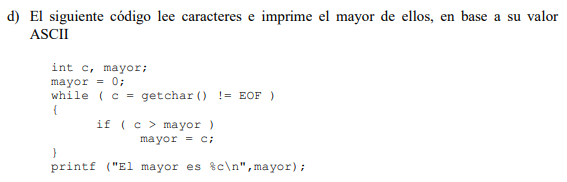


No se puede comparar directamente tipos de punto flotante por el error de representación. Para ello usamos un epsilon tal que la diferencia entre la variable y el valor querido sea menor a epsilon,

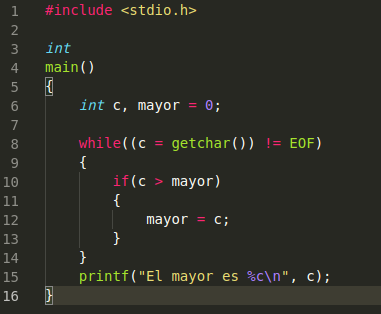




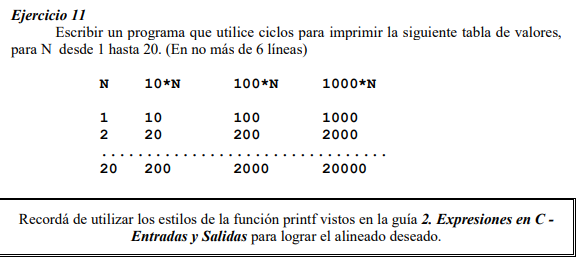
En case 1: es necesario break; sino cuando es 1, ejecuta case 1 y case 2.

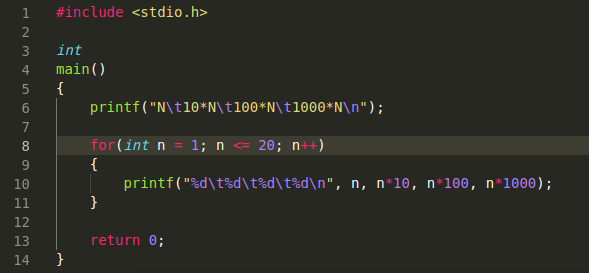


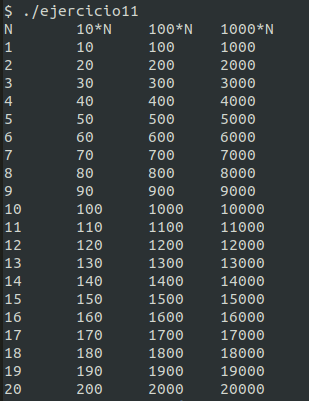
c = getchar() debe estar entre paréntesis por la precendecia de los operadores. Sino realiza primero la comparación de getchar contra EOF y ese resultado lo asigna a c, siendo posibles los valores 0 o 1 únicamente.



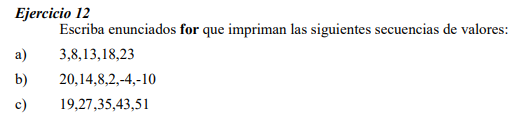
### Ejercicio 11

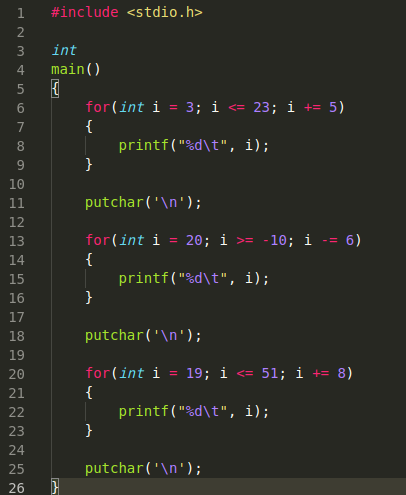


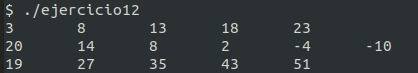




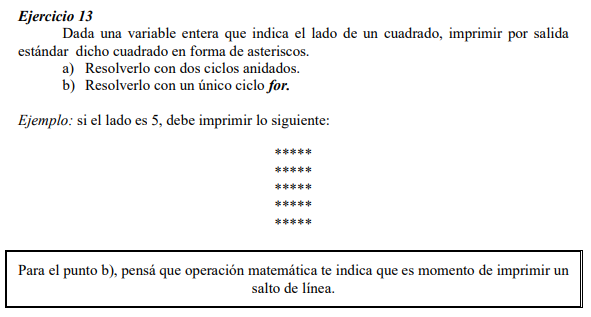
### Ejercicio 12

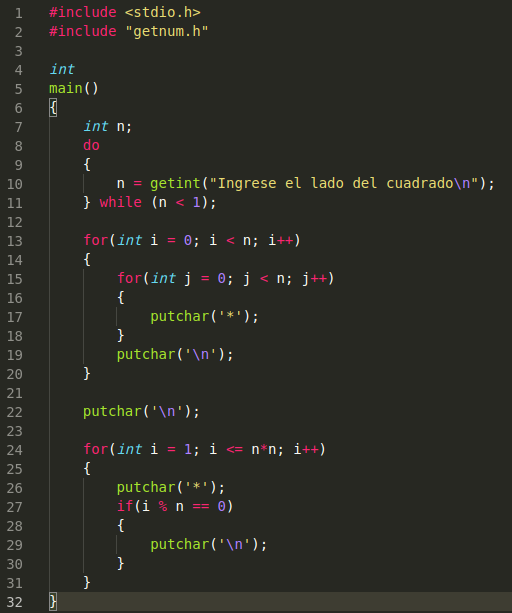


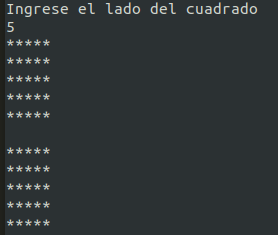




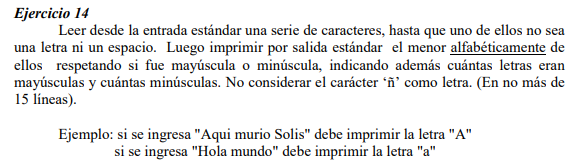
### Ejercicio 13

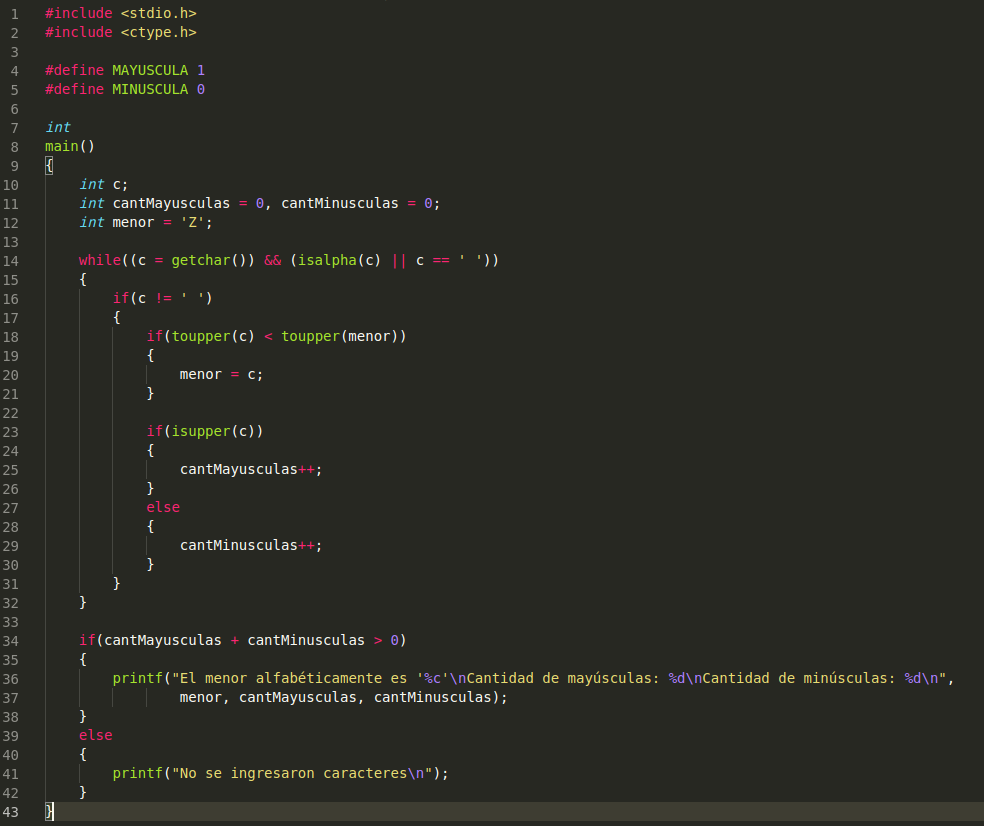


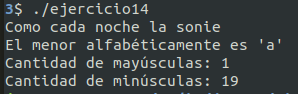




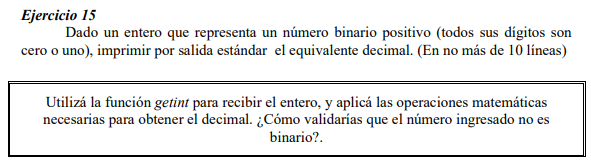
### Ejercicio 14







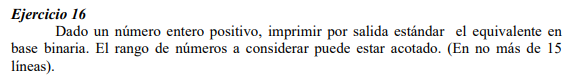
### Ejercicio 15

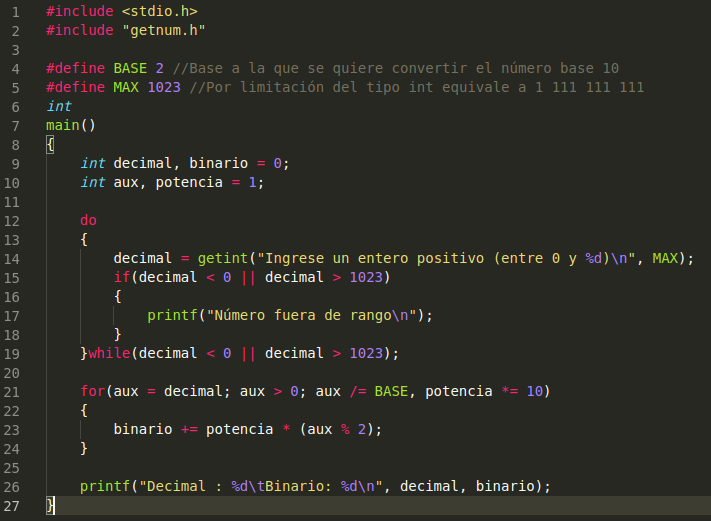




### 

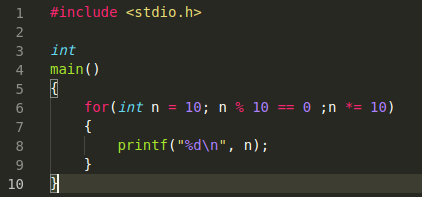
### Ejercicio 16

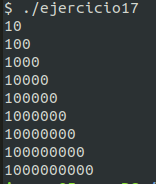




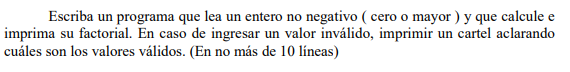
### Ejercicio 17

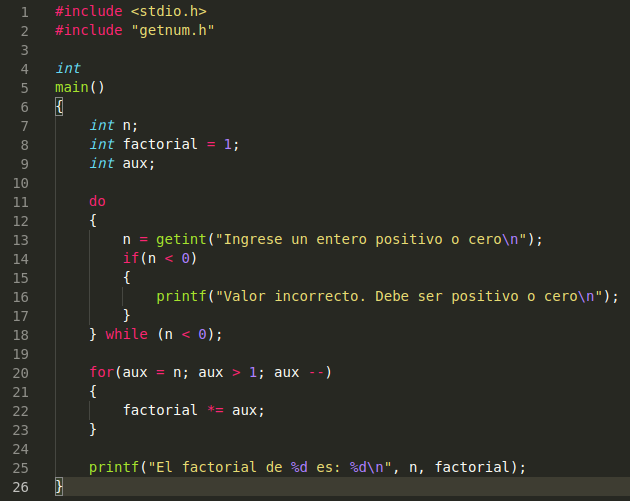




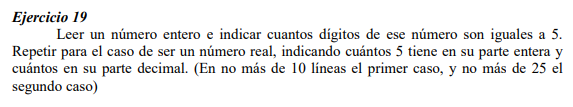


### Ejercicio 18

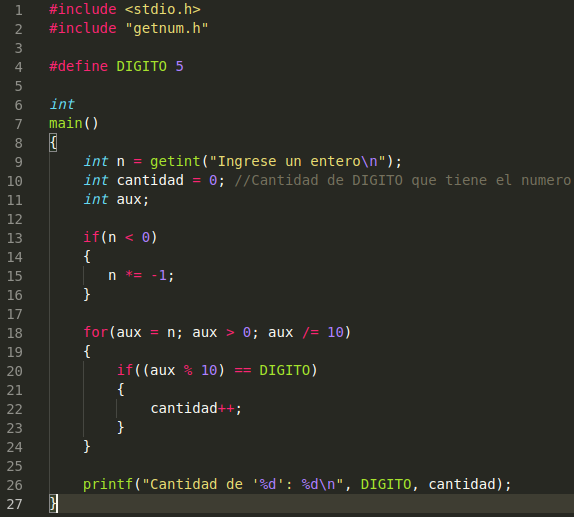




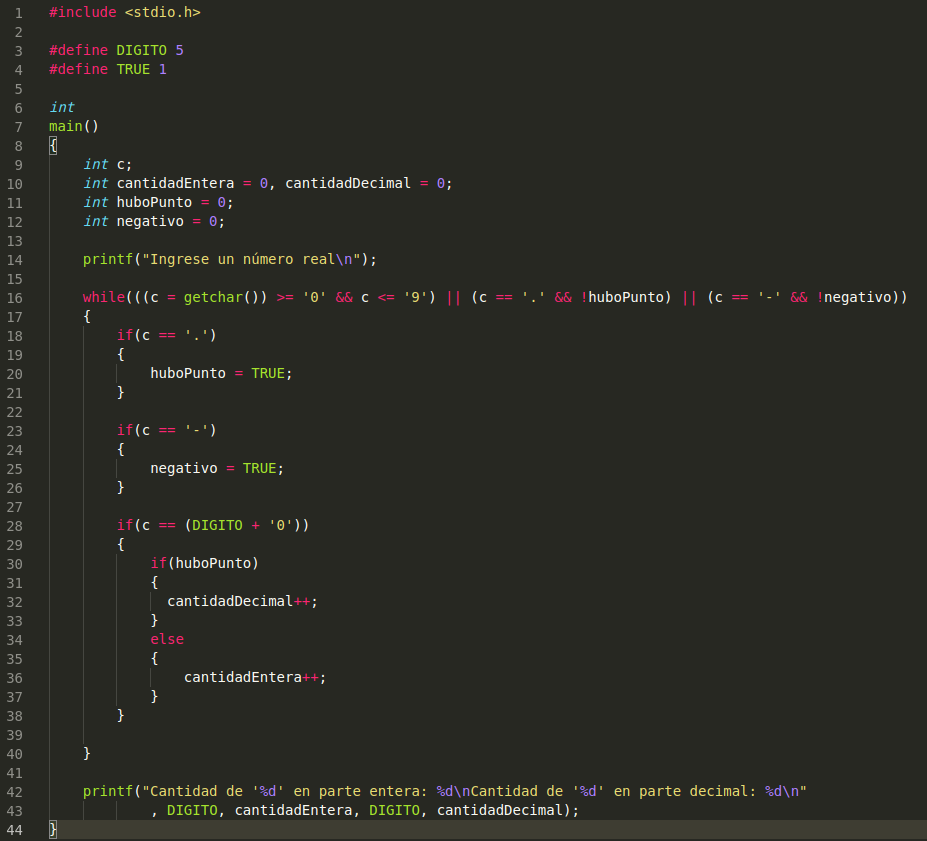
### Ejercicio 19



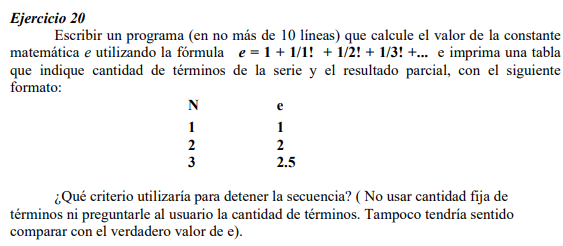
Parte a. Para número enteros.

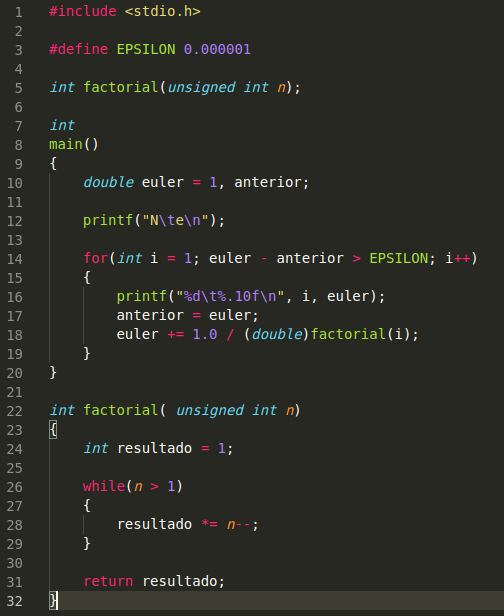


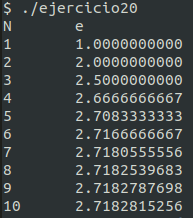
Parte b. Para número reales.



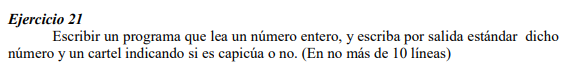
### Ejercicio 20

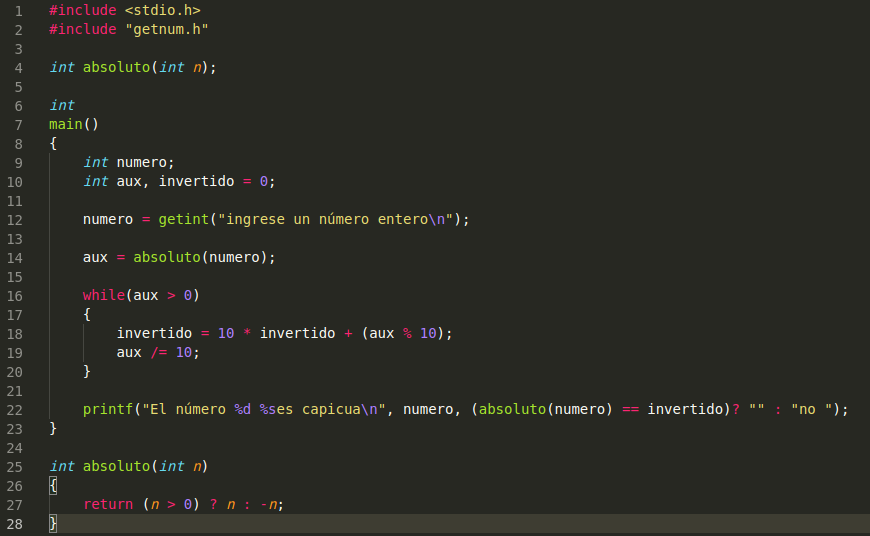




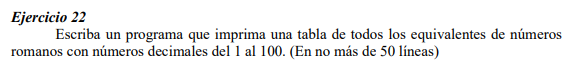


### Ejercicio 21

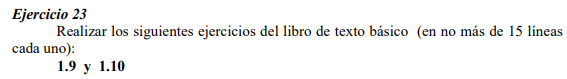




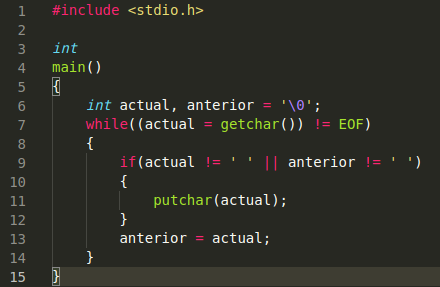
### Ejercicio 22

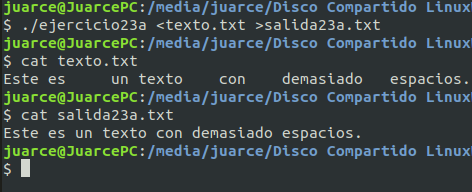


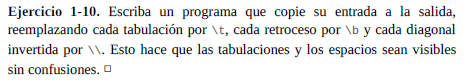
### Ejercicio 23



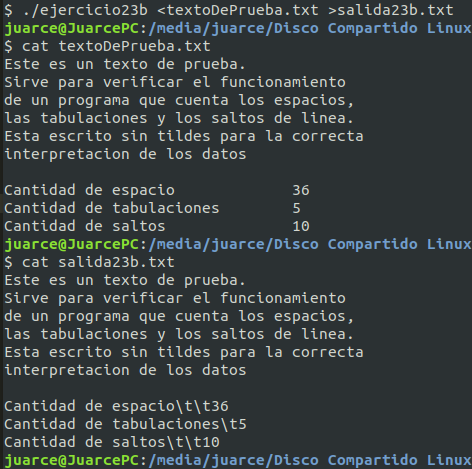




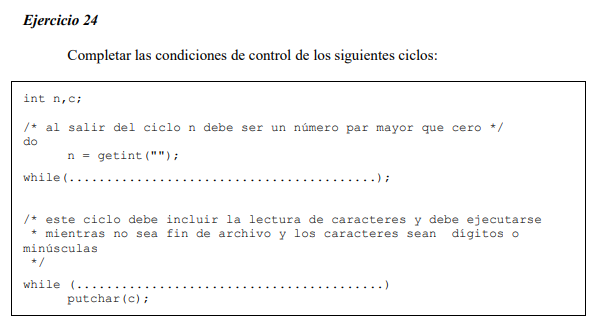


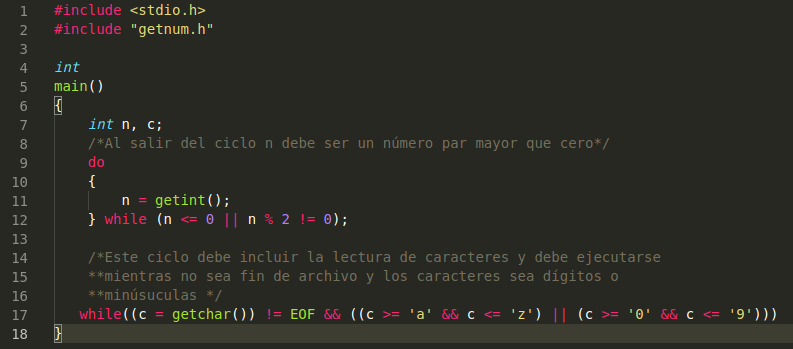




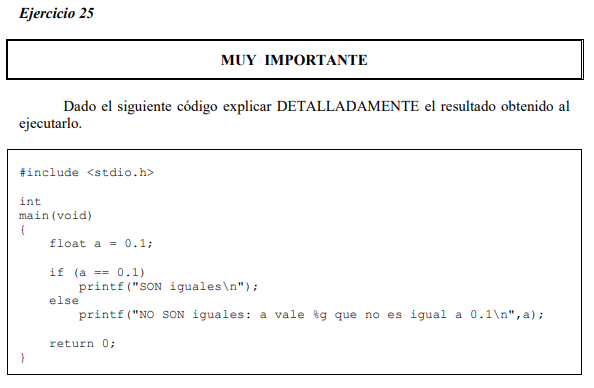


### Ejercicio 24





### Ejercicio 25



No son iguales debido a que ‘a’ no es exactamente igual a 0.1 por el error de representación propio de los números reales (IEEE 754).

### Ejercicio 26

