Estimad@s alumn@s tengan uds. muy buenos dias, para esta clase vamos a ver 2 sentencias que, si bien son simples, cumplen un rol que resultara de gran utilidad en la completitud de los programas.

**CONTINUE**

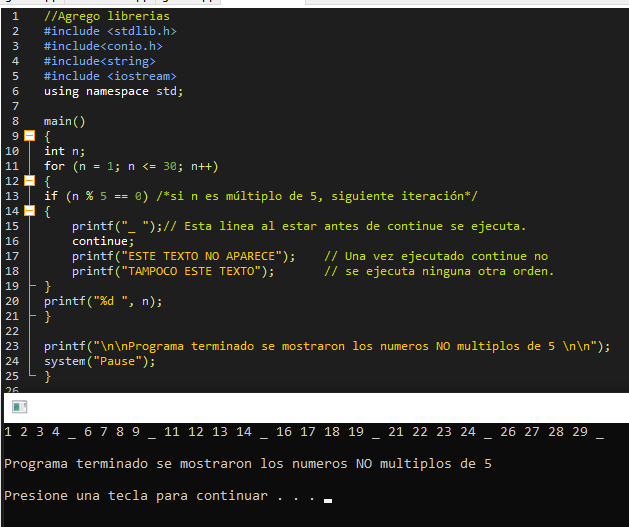
     La sentencia continue obliga al programa a continuar, esto significa que omitirá cualquier acción una vez puesta en funcionamiento. Para poder funcionar continue tiene que estar incluida dentro de una iteración del tipo **for**, **while** o **do while**.

     Su sintaxis es:

For(), while, do  
{  
continue;  
}  
while

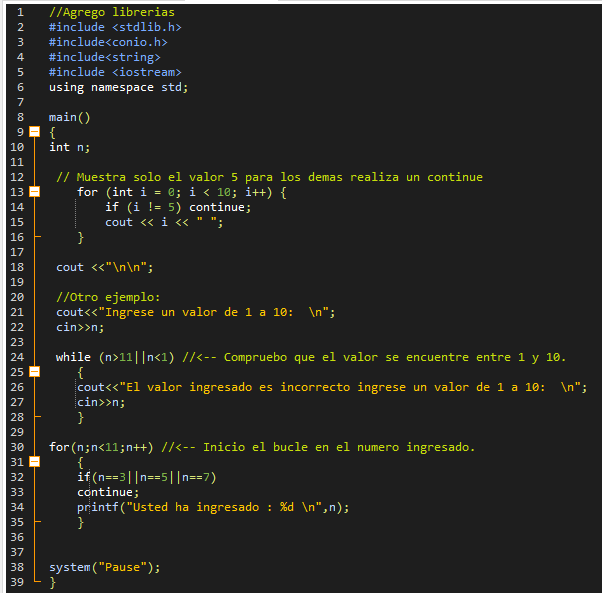
     Como ejemplo, vea el siguiente programa que imprime todos los números entre 1 y 30 y le imponemos una condición de que estos no sean múltiplos de 5.

**\*Ejemplo 1 continue:**



     Ejecute este programa y observe que cada vez que se ejecuta la sentencia continue, todas las ordenes posteriores quedan nulas, de esta forma podemos notar que la primer línea de código printf(“\_ “); sera visible pero no asi los printf siguientes.

**\*Ejemplo 2 continue:**

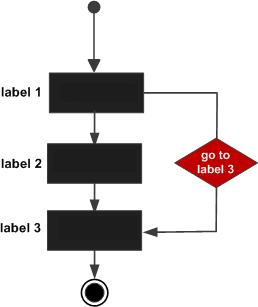


         En este ejemplo vamos a ver un mini programa donde un for mostrara solo el valor 5 y descartará gracias a continue todos los demás valores, y otro programa donde realizaremos el ingreso a una variable que vaya del 1 al 10, una vez ingresado un valor a partir de ese número arrancará nuestro for para realizar un bucle que omitirá si encuentra los valores 3/5/7 (nótese como el **for** está esquematizado “for (n; n<11; n++)”). Por favor copien el código y prueben este programa para poder comprenderlo en la práctica.

**GOTO**

     La sentencia **goto** ya la estuvimos utilizando en varios ejercicios anteriores y lo que permite realizar es una transferencia del control a una línea específica del programa, identificada por una etiqueta. Su sintaxis es la siguiente:

goto etiqueta;  
…  
…  
…  
etiqueta:

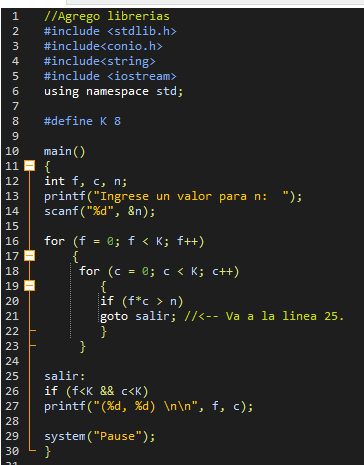


     Si la línea a la que se transfiere el control es una sentencia ejecutable, se ejecuta esa sentencia y las que le siguen. Si no es ejecutable, la ejecución se inicia en la primera sentencia ejecutable que se encuentre a continuación de dicha línea.

     No se puede transferir el control fuera del cuerpo de la función en la que nos encontramos. Un uso abusivo de esta sentencia da lugar a programas difíciles de interpretar y de mantener. Por ello, en programación estructurada, se utiliza solamente en ocasiones excepcionales. La función que desempeña una sentencia goto, puede suplirse utilizando las sentencias if...else, do, for, switch, while y do while.

     El uso más normal consiste en abandonar la ejecución de alguna estructura profundamente anidada, cosa que no puede hacerse mediante la sentencia break, ya que ésta se limita únicamente a un solo nivel de anidamiento.

**\* Ejemplo 1 goto:**

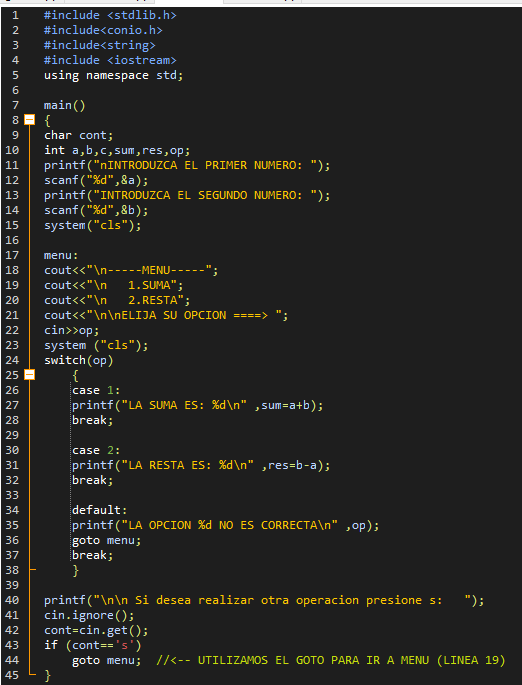


     El siguiente ejemplo muestra cómo se utiliza la sentencia goto. En este caso la estructura consta de dos bucles for anidados. En el bucle interior hay una sentencia goto que se ejecutará si se cumple la condición especificada es decir que “f\*c” sea mayor a “n”. Si se ejecuta la sentencia goto el control es transferido a la primera sentencia ejecutable que haya a continuación de la etiqueta salir: (que se encuentra en la línea 25 de código).

     Además en este ejercicio vamos a utilizar algo más, observemos en la línea 8 de código utilizamos un “**#define K 8**” esta regla global permite asignarle un valor universal a la variable **K = 8**, esto es muy útil ya que al ser global funcionara como constante en todo el programa, de esta forma cuando yo quisiera cambiar los parámetro del ejercicio solo cambio el valor del define K y resuelvo automáticamente en todos los lugares donde este utilizado K, en nuestro ejercicio lo utilizamos en los 2 for y hay una última instrucción de f<K y c<K (línea 26) en este caso si nuestro valor fuera menor a 49 veríamos los valores de f y c (ya que 7x7=49) por pantalla de otra forma como K es igual a 8, no nos mostraría nada.

Otro Ejemplo de uso **MUY UTIL**:

**\*Ejemplo 2 goto:**



            Este ejemplo es bastante explícito y muy similar al que ya vimos la clase pasada en switch, la única diferencia primordial es como utilizo el **goto** para el case default de switch y también lo utilizo como respuesta al if para realizar otra operación, líneas 36 y 44 respectivamente.

\*\*goto no se recomienda para programas extensos, siguiendo las normas de la buena programacion, dado que puede generar conflictos en cuanto al direccionamiento, sin embargo para programas que son en su diagramacion pequeños es un recurso muy util y por eso es necesario conocerlo.