 Bienvenid@s!! a la clase 6, ya estamos bastante avanzados en cuanto a lo que se refiere a programar, en la clase anterior aprendimos como utilizar la sentencia de condición " if ". Ahora estudiaremos la sentencia de iteración " for " que realiza bucles de forma precisamente definidas y que poseen un esquema sencillo de acción.

     Este esquema es así:

for( inicialización del bucle;  **condición**;  *incremento* ).

     Ahora vamos a proceder a definir cada una de las partes del esquema básico:

1) Inicialización (expresión 1): asignación que se utiliza para comenzar el bucle.

2) Condición (exp. 2): es una expresión relacional que hace que el bucle concluya cuando no se cumpla.

3) Incremento (exp. 3): es la parte que determina como cambia la variable hasta que el bucle concluye. ej. Suma 1 vuelta al bucle.

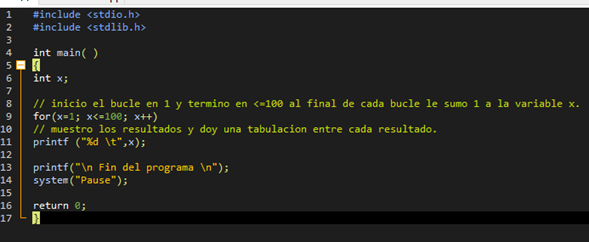
     Es importante señalar que cada parte ha de ser separado por " ; " .

     Para que deje de ejecutarse el bucle como antes hemos mencionado dependemos de la condición, que deberá de ser falsa para que el bucle se detenga.

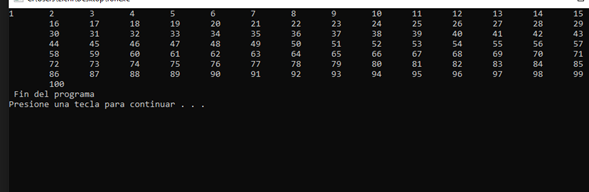
     Debemos tener en cuenta que a diferencia de otros bucles este no llegará a ejecutarse si la condición es falsa, ya que la condición se comprueba antes de entrar en el bucle.

     Esta sentencia de iteración resulta muy útil por sobre todo para controlar la cantidad de veces que se deben efectuar operaciones, ya sean estas aritméticas, de validación o de comprobación de estado, ya que un proceso se realizara tantas veces sea necesario hasta concretar una condición especifica (por ejemplo: for (a=1; a<=100; a++ ) ), es decir que inicia la función de la variable a = 1, mientras que sea menor o igual a 100, realiza el bucle y luego le suma 1 a la variable a, por lo que el proceso se realizara 100 veces.

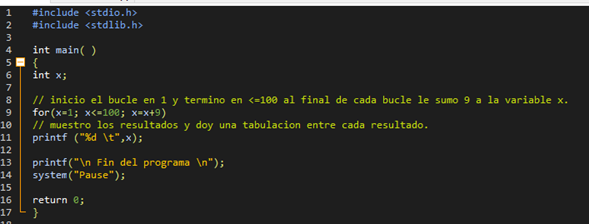
     Un *sencillo* ejemplo:



     *Obtenemos como resultado:*

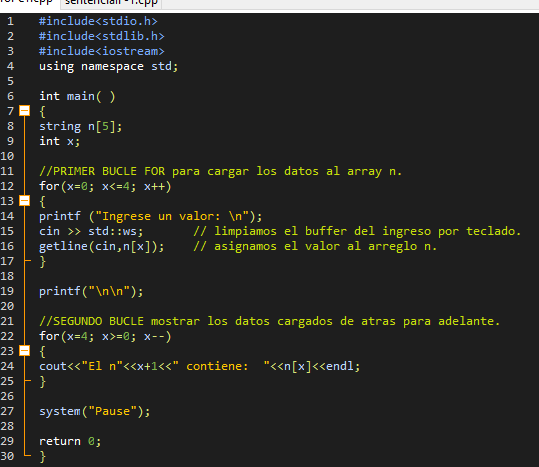


     Otro ejemplo utilizando el mismo formato del ejercicio 1:



*\*presten atención a la asignación de x. Prueben ahora como se ve por pantalla.*

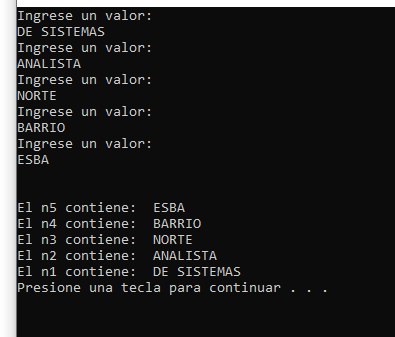
     Vamos a realizar otro ejercicio que nos muestra como la sentencia "for" nos permite ordenar de atrás para adelante, en este caso vamos a memorizar 5 cadenas de texto y luego ordenarlos desde el ultimo al primero:



     Si fuimos cuidadosos veremos que en este último ejemplo utilizamos algo nuevo al lado de la variable string "n" aparece " [5] ", esto es un array de tamaño 5, lo veremos específicamente más adelante, por ahora lo que necesitamos saber es que memoriza 5 espacios de asignación de memoria a los cuales yo puedo volver por ejemplo el valor que está en n[3].

     El ejercicio en si consta de una variable simple que queda almacenada en una posición que va desde 0 a 4, en el primer " for " almacenamos valores en distintas posiciones n[0], n[1]…,n[4].  En el segundo for recorremos y mostramos por pantalla lo que se encuentra almacenado en n[4], n[3]......, n[0].  Es por ello que la variable x que es el valor de asignación de n[x] en el primer for suma desde 0 a 4 y en el segundo for resta desde 4 a 0.

         El ejemplo nos quedaría asi:



*\*Uds. Prueben asignándole valores de texto al ejemplo.*

*\*Notaron que para mostrar los resultados utilice la sintaxis:*

**cout<<"El n"<<x+1<<" contiene:  "<<n[x]<<endl**;

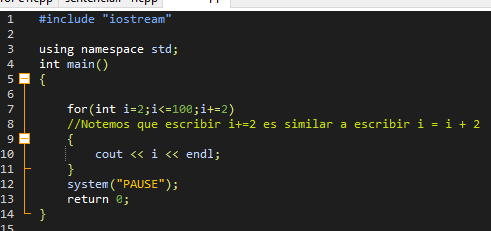
    esto seria como decir:

**cout<<TEXTO<<variable+1<<TEXTO<<valor contenido en n[x]<<salto de línea.**

    Utilizo el valor de la variable x+1 es para que en la salida se vea mas prolijo desde n[1] a n[5] sino mostraría de 0 a

**\*Ejemplo 1: Ver en pantalla los números pares.**

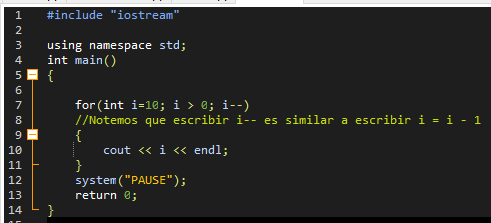
Tenemos que el valor inicial para nuestro ciclo el cual inicia en 2 y el valor final es el 100, además, dado que necesitamos los números pares vamos a ir de dos en dos, así que el tamaño del paso va a ser 2, teniendo estos 3 componentes identificados, estamos listos para crear nuestro ciclo for así:



**\*Ejemplo 2: Cuenta regresiva en un ciclo for.**

     Ahora veremos otro ejemplo sencillo en cual veremos que el ciclo for también puede iterar disminuyendo el valor del contador, para comprender esto, lo que haremos será imprimir por pantalla una cuenta regresiva desde el número diez hasta el cero, veamos:

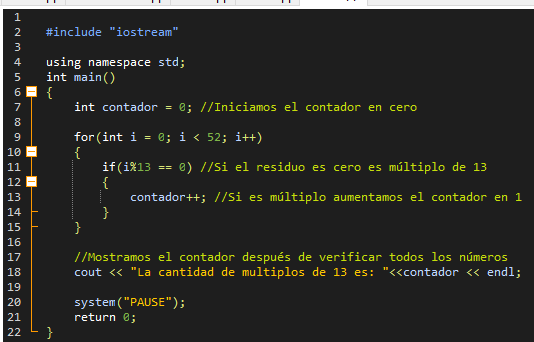
     Para este caso, debido a que queremos ir de un número mayor a uno más pequeño, por lo tanto, para este ejemplo el valor inicial será 10 y el valor final será cero. Adicional, el tamaño de paso será de 1 negativo, es decir, -1, así:



**Ejemplo 3: Contador con un ciclo for.**

     Para este ejemplo haremos algo un poco más complejo. El ejemplo consiste en contar al interior de un ciclo for, cuántos números entre el 0 y el 52 son múltiplos del 13. Para ello haremos uso del operador % (modulo) que obtiene el residuo de una división y también usaremos un pequeño condicional “if” para verificar que el modulo sea cero al dividir por 13.

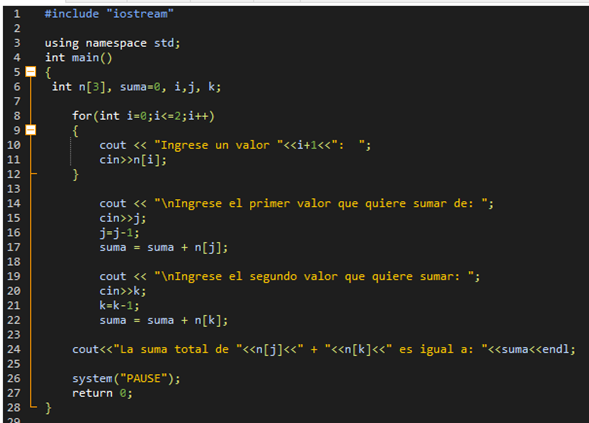
Para este caso el valor inicial será 0 y el valor final será 52. Adicional, el tamaño de paso será de 1. Al interior del ciclo, en cada iteración verificaremos si el número en el que estamos es divisible por trece o no y en caso afirmativo aumentaremos el contador en una unidad así:



*\*Este ciclo for nos permitirá saber que existen 4 múltiplos del 13 en los números del 0 al 52.*

**Ejemplo 4: for Suma dirigida.**

         En el siguiente ejemplo vamos a utilizar el for para asignarle valores a la variable n[3], es decir un vector con 3 espacios. El primer ciclo se encarga de permitirnos cargar los valores que queremos, siempre números enteros, tal como lo indica la variable. Luego hacemos dos salidas con cout, donde nos preguntara que valores de n queremos sumar. Por ultimo nos queda mostrar el resultado de la suma y se ha finalizado el ejercicio.



***\*\*Utilizaremos estos ejercicios para el próximo trabajo de EQUIPO así que por favor cópienlos y téngalos preparados***.