|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre de la práctica** | **REPORTE EJERCICOS LAMINAS: 8-11** | | | **No.** | **1** |
| **Asignatura:** | **MÉTODOS NUMÉRICOS** | **Carrera:** | **Ingeniería en sistemas computacionales** | **Duración de la práctica (Hrs)** |  |

**YOEL HERNANDEZ DE JESUS**

**Competencia(s) específica(s):**

Aplicar el lenguaje de programación c para la solución de problemas.

**II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):**

* Casa
* Salón de clases

**III. Material empleado:**

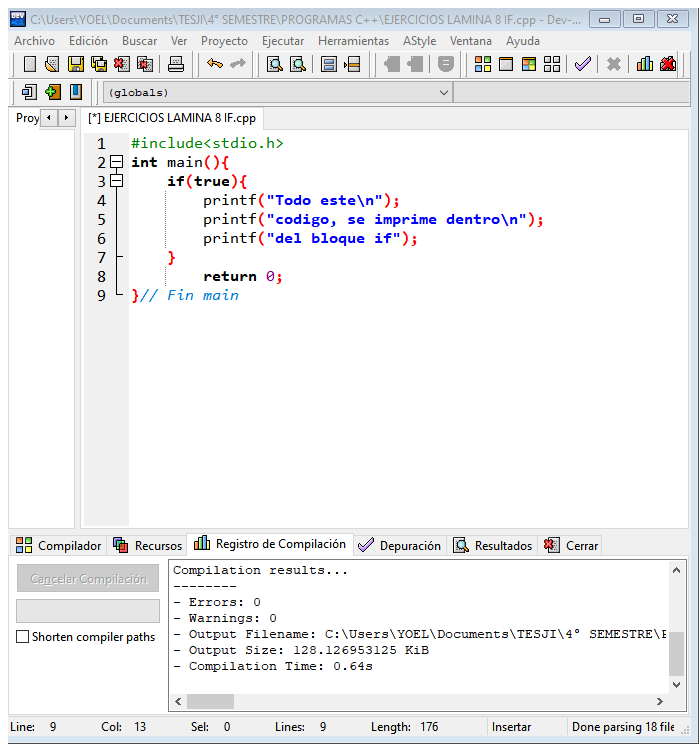
* Computadora

**IV. Desarrollo de la práctica:**

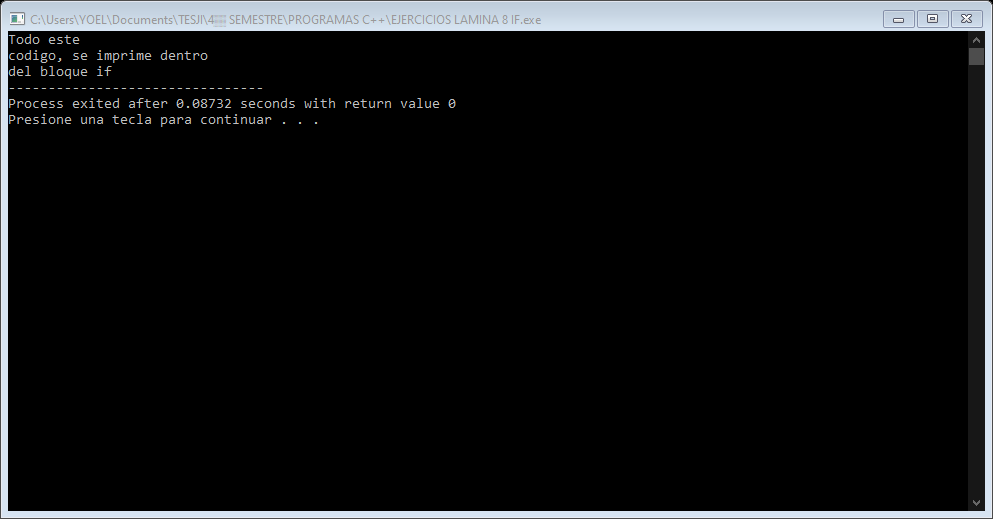
**EJERCICIO DE LAMINA 8 IF:**

**Ejercicio 1:**

Se dice que cuando el código condicionado consta de mas de una instrucción, las sentencias se agrupasn dentro de un par de llaves:

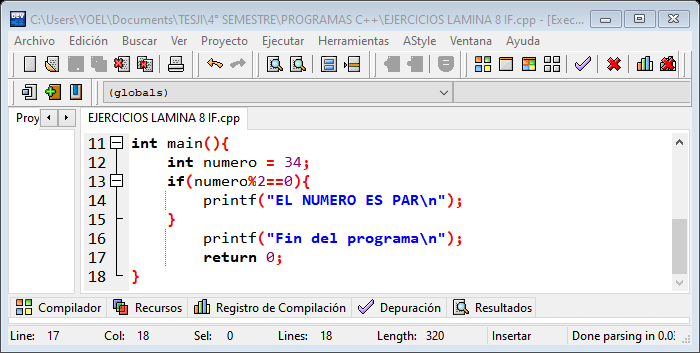


**Resultado**

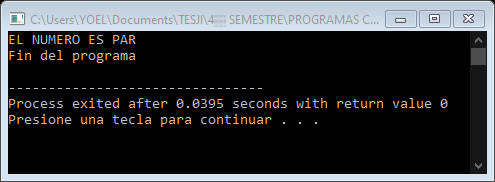
****

**Ejercicio 2:**

Se usa una variable llamada numero con un valor de 34, seguido de un ciclo if donde se evalúa la condición si se cumple prosigue de lo contario corre la siguiente línea:

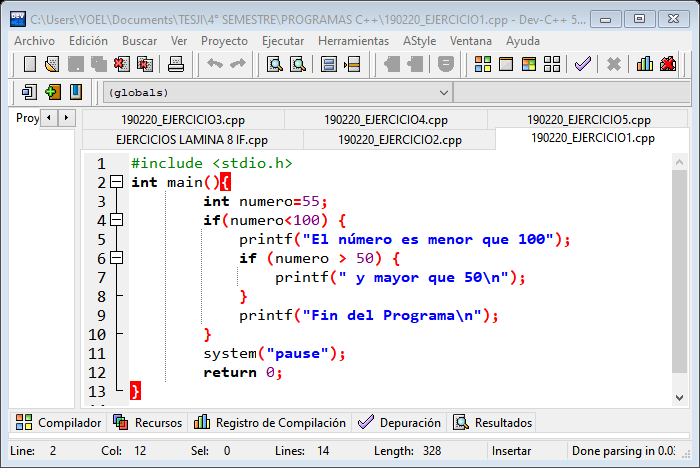


**Resultado:**

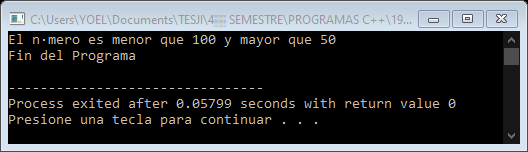
****

**Ejercicio 3:**

Una estructura es anidada cuando existe una estructura dentro de otra, ejemplo:



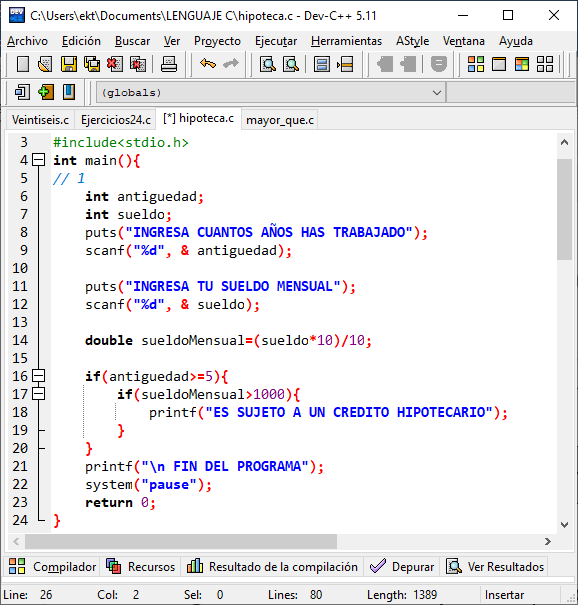
Se observa que dentro de un ciclo if se encuentra otro ciclo así que si se cumple la primera condición se debe cumplir la segunda condición:



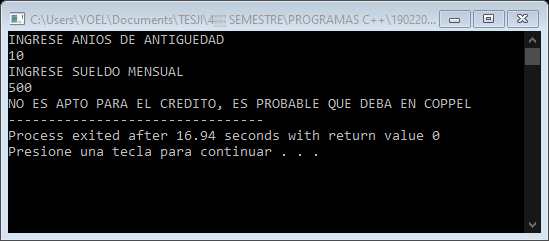
**Ejercicio 4:**

La condición utilizada en este programa fue IF pues se busca saber qué persona está sujeta a un crédito hipotecario pero se deben cumplir las siguientes condiciones:

* Que la persona tenga un mínimo de 5 años de antigüedad en su trabajo.
* Y su sueldo mensual sea mayor a 1000



**RESULTADOS:**

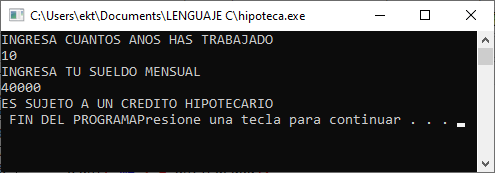
****

CASO 1: **NO SUJETO A CREDITO HIPOTECARIO**

* Una persona que solo cumple con el mínimo de años cumplidos en su trabajo.

Pero su sueldo es

* menor a los 1000

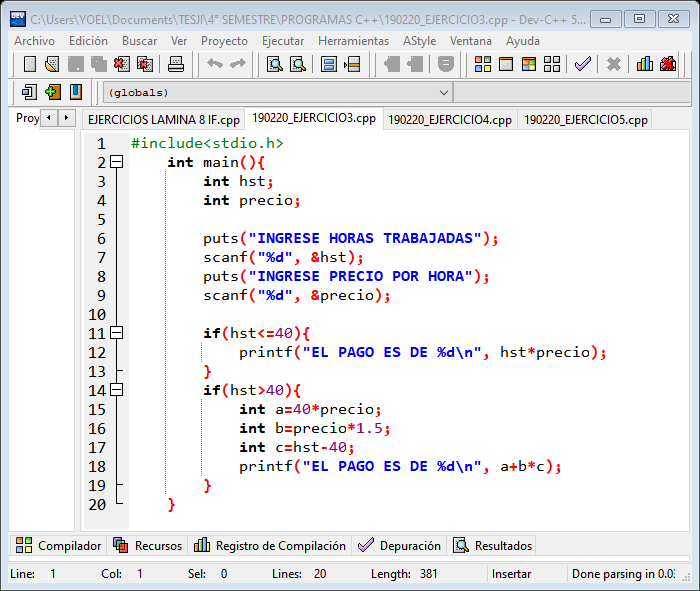


CASO 2: **SUJETO A CREDITO HIPOTECARIO**

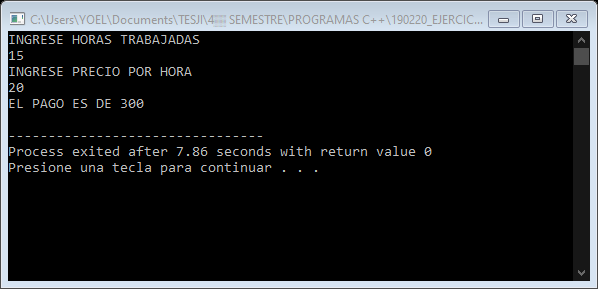
* La persona que tiene mas de 5 años de antigüedad
* También tiene un mínimo de 1000 mensuales

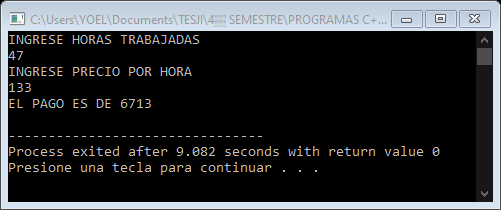
**EJERCICIO 5:**

Se desea calcular el salario semanal de un empleado, sabiendo que este se calcula en base a las horas semanales trabajadas y de acuerdo con un precio especifico por hora. Si se pasa de las 40 horas semanales, las horas extras se pagarán a razón de 1.5 veces la hora ordinaria.

****

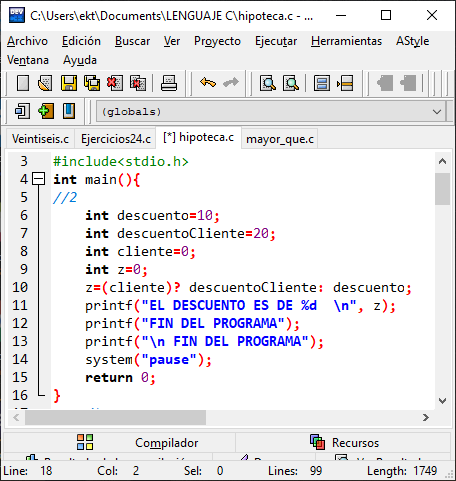
**Resultado:**

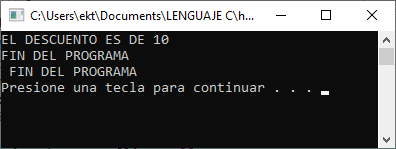
****

****

**EJERCICIO 6:**

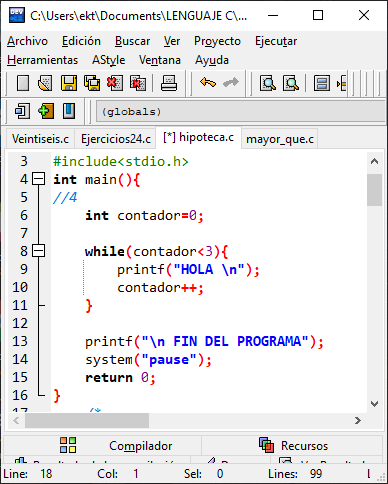
**Se usa el operador condicional:**



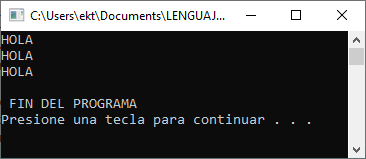


**Lamina 9 ciclo while:**

**EJERCICIO 1: Es repetir un bloque de código mientras una condición se mantenga verdadera.**

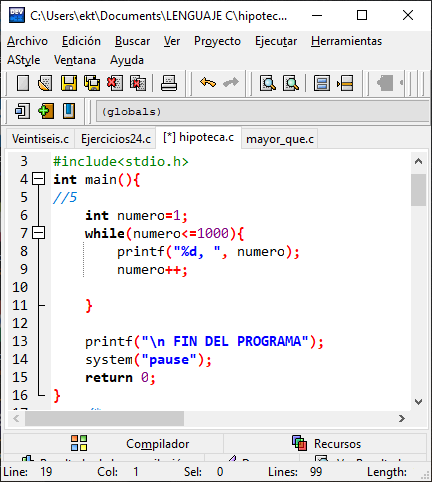


**Resultado1:**

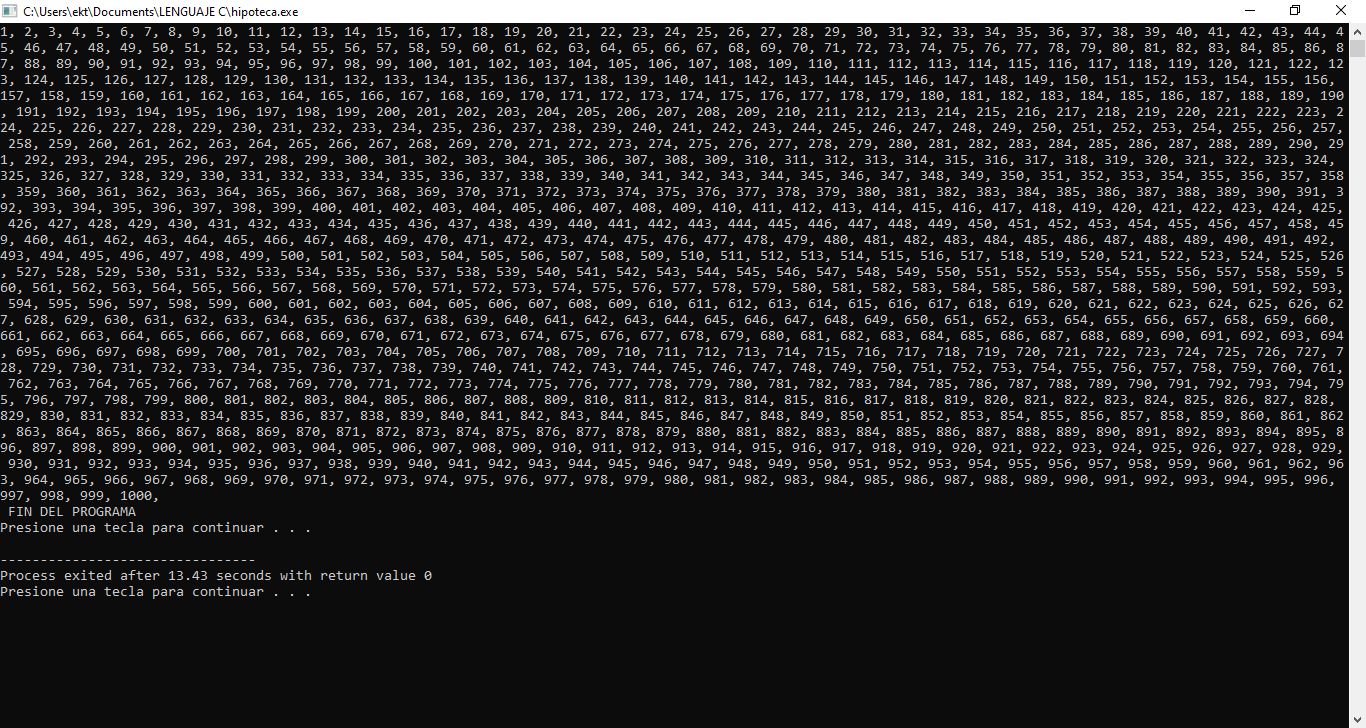


**EJERCICIO 2:**

Se hace uso de un ciclo while para que imprima los antecesores de 1000.

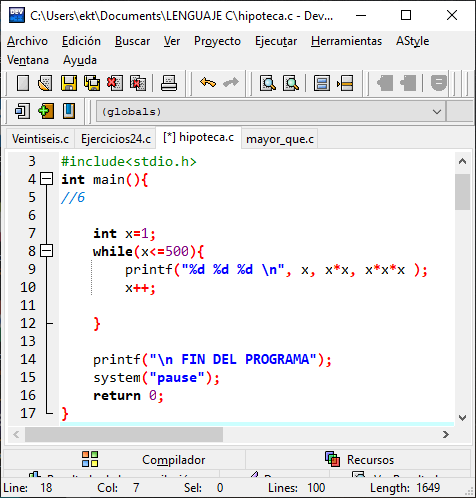


**Resultado:**

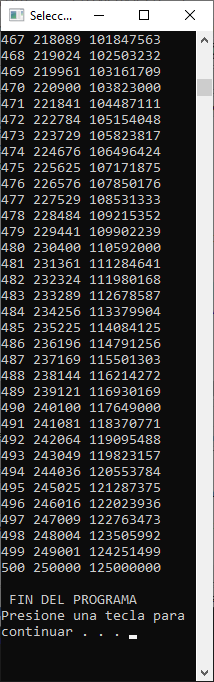
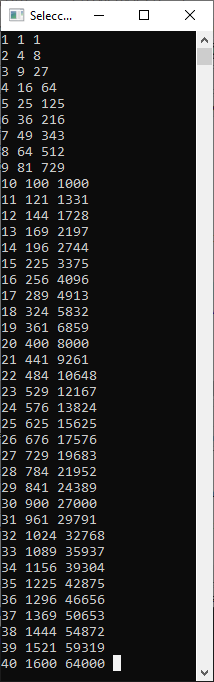


**EJERCICIO 3:**

Imprime los cuadrado y los cubos de los primeros quinientos números naturales:



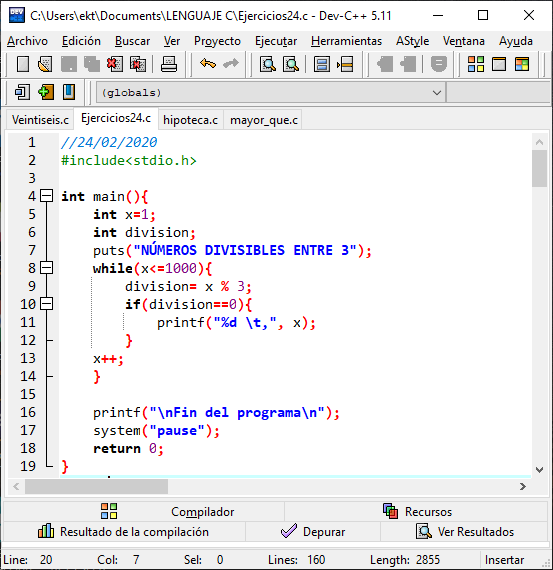
**Resultado**



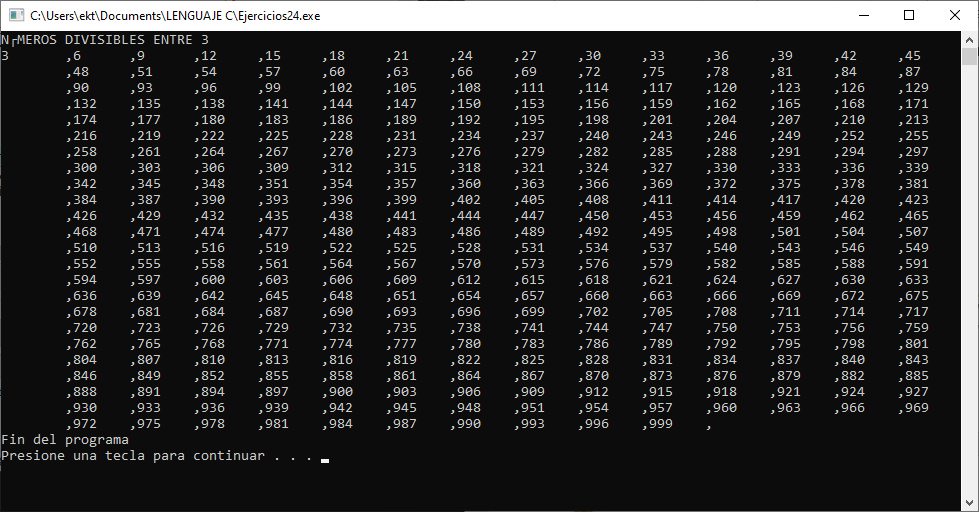
**EJERCICIO 4:**

Imprimir todos los números divisibles entre 3 mayores a 0 y menores a mil.

Ejemplo: 3, 6, 9, 12, 15, 18, 21, …. 999



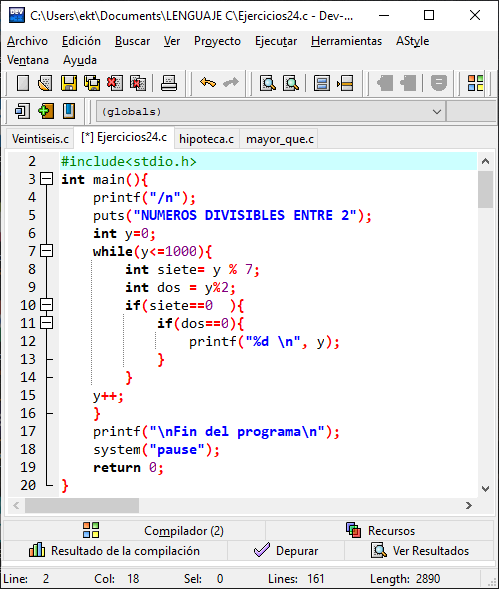
**Resultado:**



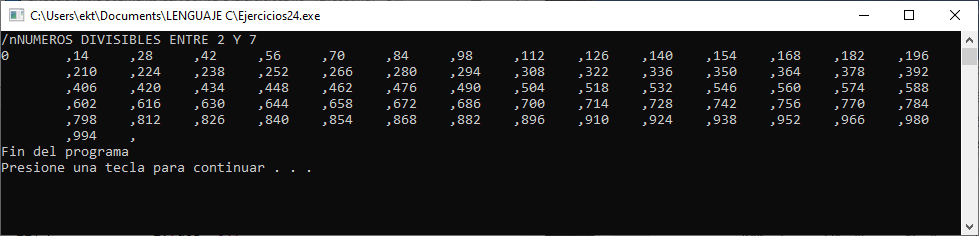
**EJERCICIO 5:**

Imprimir todos los números que son divisibles entre 2 y entre 7, mayores a 0 y menores a mil.

Ejemplo: 14, 28, 42, 56.

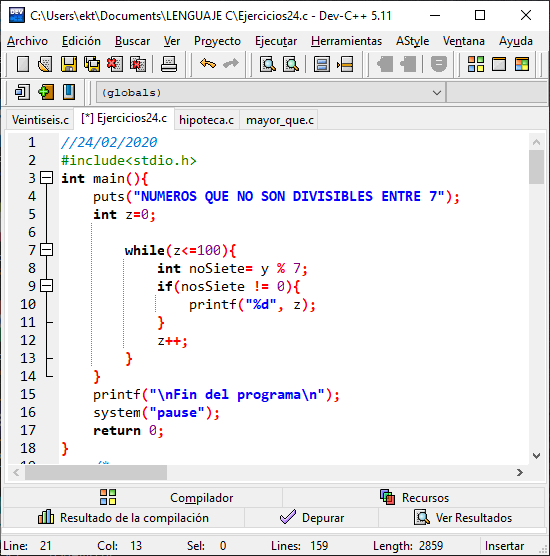


**Resultado:**

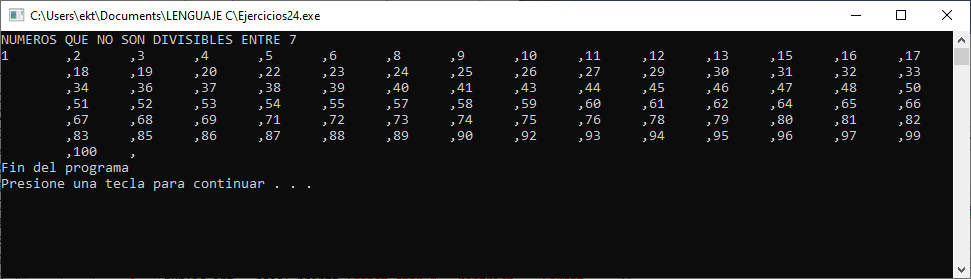


**EJERCICIO 6:**

Escribir todos los enteros positivos menores que 100 omitiendo aquellos que son divisibles por 7.



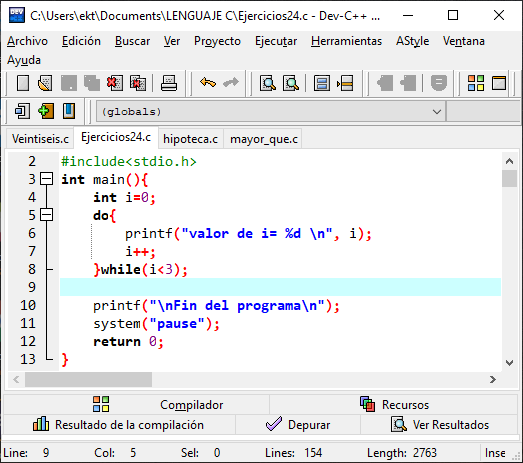
**Resultado:**



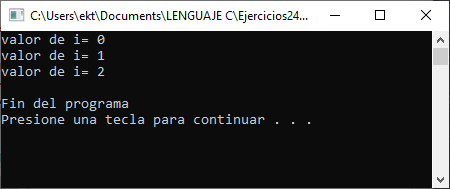
**CICLO DO WHILE**

**EJERCICIO 1:**

Primero ejecuta el conjunto de instrucciones y después verifica que la condición se cumpla:



**Resultado:**

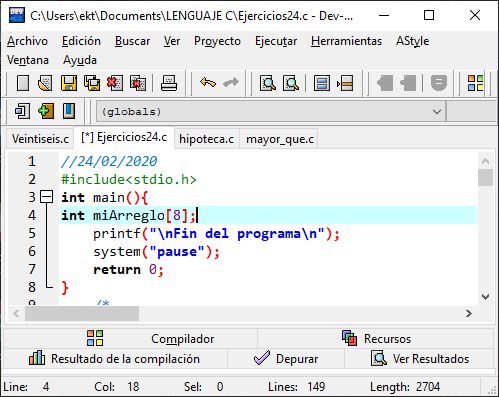


**Lamina 10: Arreglos**

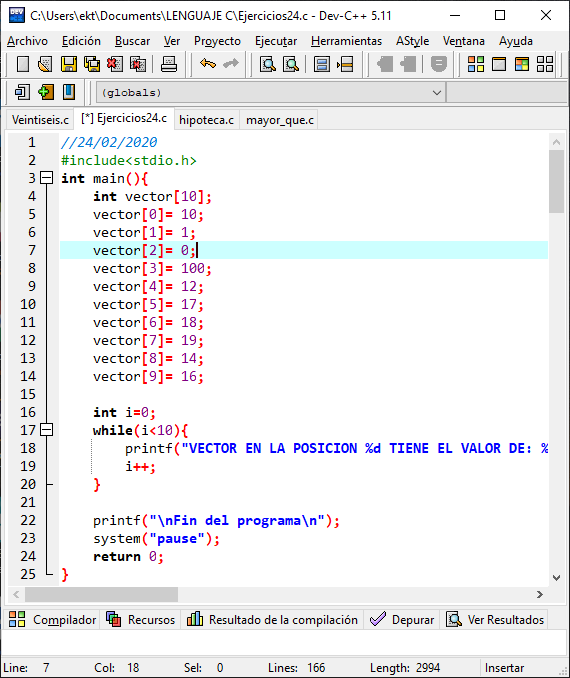
**EJERCICIO 1:**

Variable que hace referencias a varias posiciones de memoria, el índice comienza en 0.

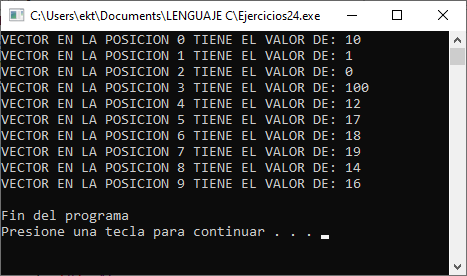
Declarando:



**EJERCICIO 2:** se establecieron 10 posiciones para crea un vector:

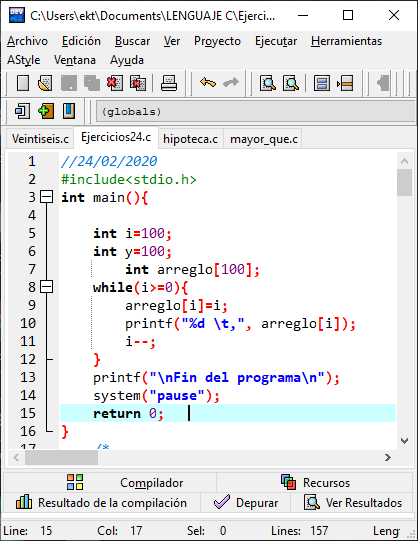


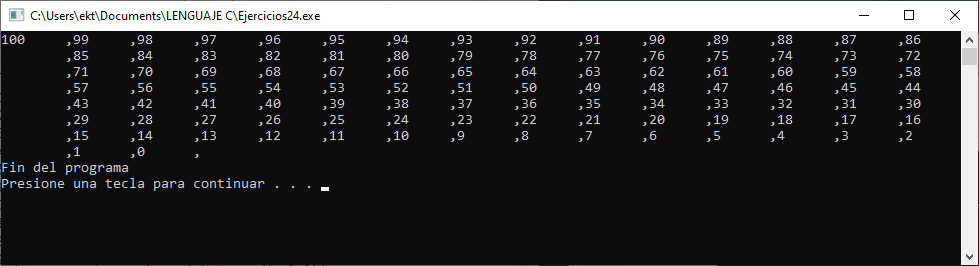
**Resultado:**



**EJERCICIO 3:**

Crear un arreglo de 100 posiciones, llenamos el arreglo con la tabla del 2 y luego mandamos a mostrar el resultado de forma normal e inversa:





**EJERCICIO 4:**

Realizar las siguientes operaciones

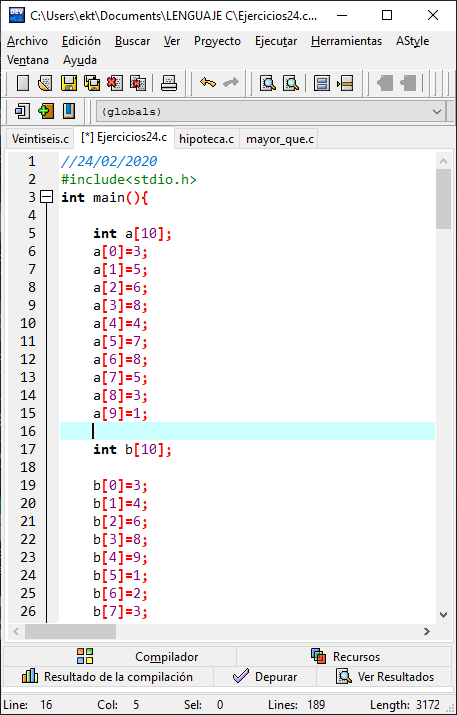
A[3] mod(B[2]/2)

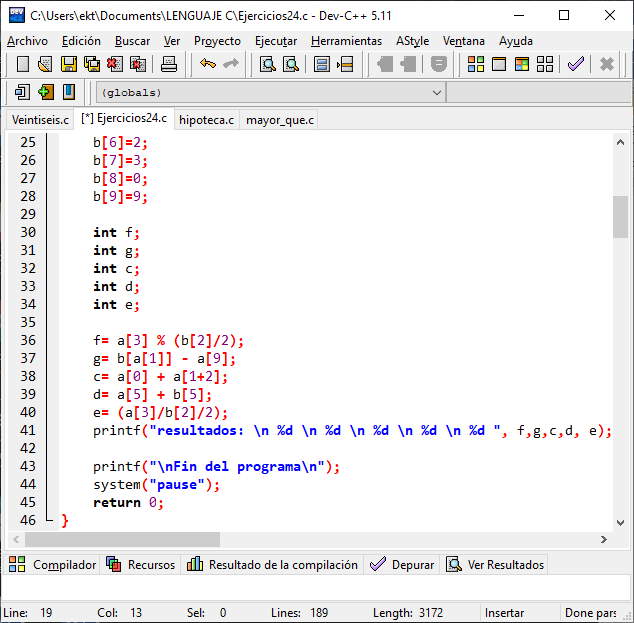
B[A[1]] –A[9]

A[0] + A[1+2]

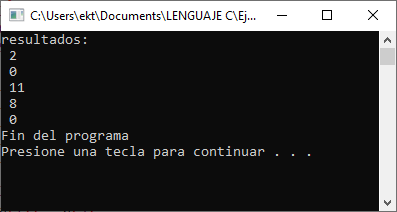
A[5] + B[5]

(A[3]/B[2])/2





**Resultado:**



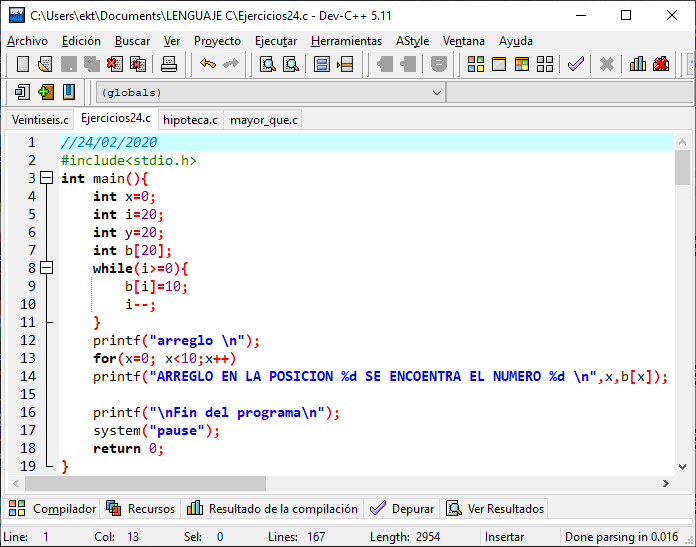
**EJERCICIO 5:**

Crea un arreglo de 20 posiciones.

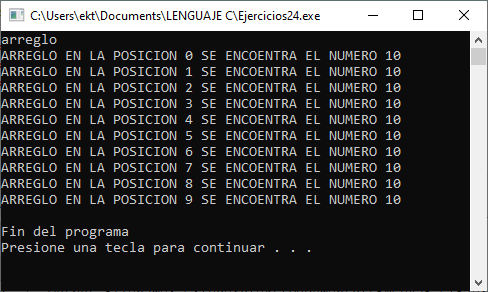
Asígnale a cada elemento un valor.

Calcula el promedio de todos los elementos.

Calcula la multiplicación de todos los elementos.



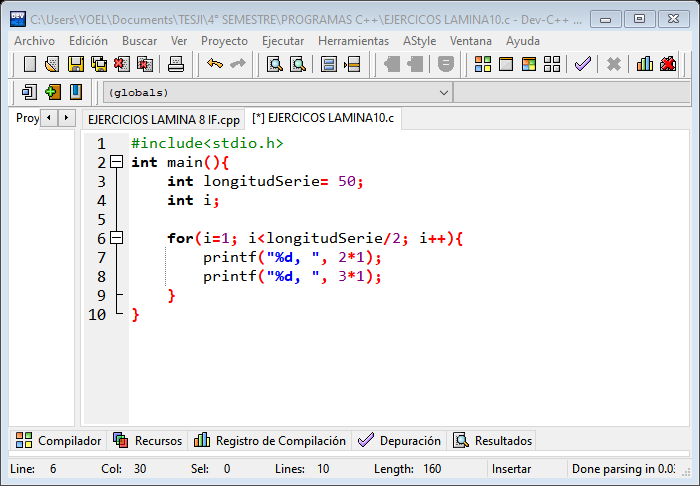
**Resultado:**



**LAMINA 11: CICLO FOR**

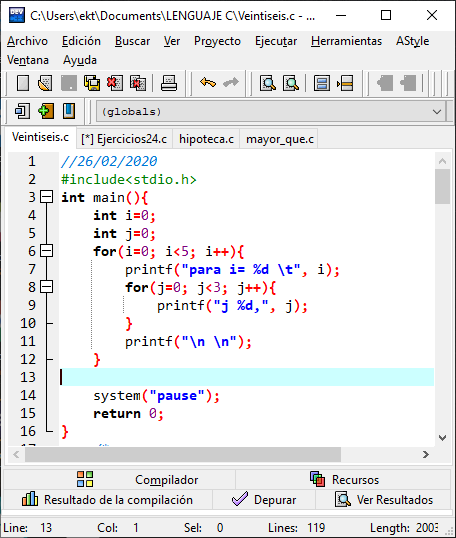
**EJERCICIO 1:**

Funciona similar al while, es muy útil para recorrer arreglos

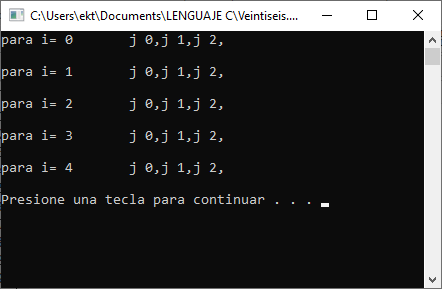
****

**EJERCICIO 2:**

For anidado es una opción donde se requiere realizar dos ciclos, y funcionan muy bien.

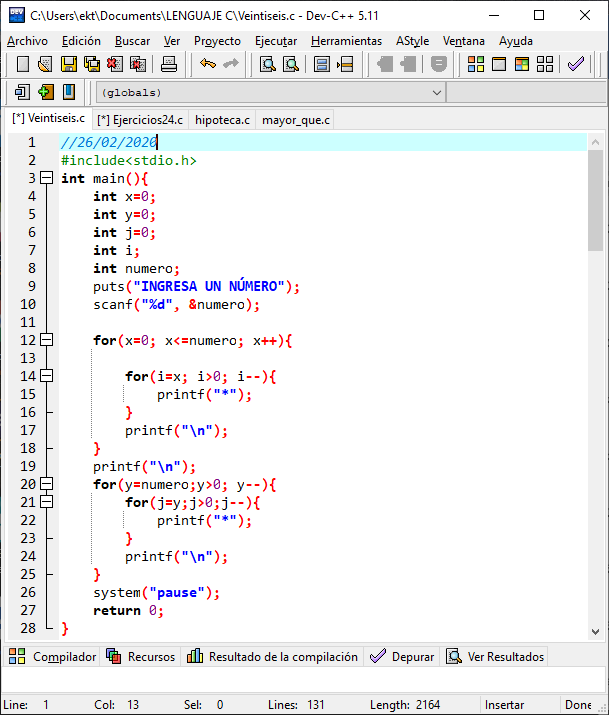


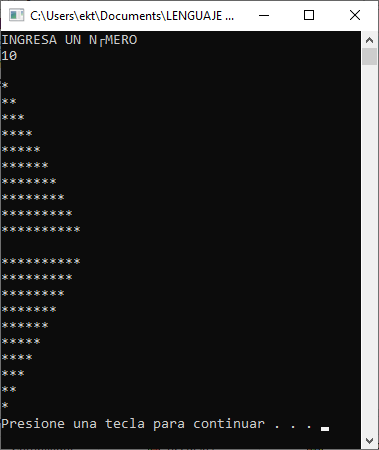
**Resultado:**



**EJERCICIO 3:**

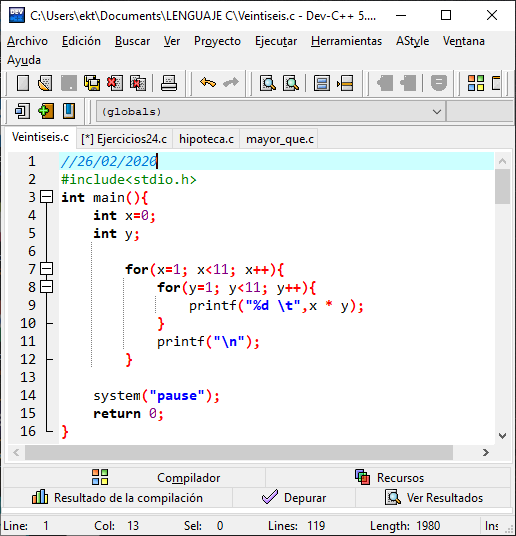
Escribe un programa que reciba un número entero N del usuario e imprima una escalinata de N pisos de asteriscos.



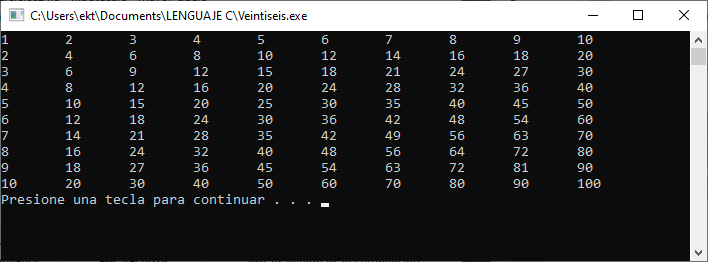


**EJERCICIO 4:**

**Haz un programa que muestre las tablas de multiplicar**



**Resultado:**



**V. Conclusiones:**

**Es bastante interesante la importancia del uso de los ciclos pues bien he aprendido a solucionar problemas algo complicados y con ayuda de los ciclos estos se vuelven muy fáciles.**