|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | RODRIGUES ESPINO CLAUDIA |
| *Asignatura:* | FUNDAMENTOS DE PROGRAMACION |
| *Grupo:* | 4 |
| *No de Práctica(s):* | PRACTICA 9: Estructuras de repetición. |
| *Integrante(s):* | FLORES MENDOZA OLGA |
|  |  |
| *No. de Equipo de cómputo empleado* | 44 |
| *Semestre:* | 2019-2 |
| *Fecha de entrega:* | 10 DE ABRIL DE 2019 |
| *Obervaciones:* |  |
|  |  |

COMPILACION

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**PRACTICA 9**

**OBJETIVOS:**

Elaborar programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva define.

**ACTIVIDADES:**

* Elaborar un programa que utilice la estructura while en la solución de un problema.
* Elaborar un programa que requiera el uso de la estructura do-while para resolver un problema. Hacer la comparación con el programa anterior para distinguir las diferencias de operación entre while y do-while.
* Resolver un problema dado por el profesor que utilice la estructura for en lugar de la estructura while. Usar la directiva define para elaboración de código versátil.

**INTRODUCCION:**

Las estructuras de repetición permiten ejecutar un conjunto de instrucciones de manera repetida (o cíclica) mientras que la expresión lógica a evaluar se cumpla (sea verdadera). Estas son las conocidas como estructuras cíclicas, iterativas o de bucles. En lenguaje C existen tres estructuras de repetición:

* **while:** La estructura repetitiva while, primero valida la expresión lógica y si ésta se cumple (es verdadera) procede a ejecutar el bloque de instrucciones de la estructura, que está delimitado por las llaves {}. Si la condición no se cumple se continúa el flujo normal del programa sin ejecutar el bloque de la estructura. Su sintaxis es la siguiente:

**while (condicion)**

**{ Enunciados }**

* **do-while :** Es una estructura cíclica que ejecuta el bloque de código que se encuentra dentro de las llaves y después valida la condición, es decir, el bloque de código se ejecuta de una a ene veces. Su sintaxis es la siguiente:

**do { Bloque de código que se ejecuta }**

**while (condicion);**

Esta estructura de control siempre termina con el signo de puntuación ';'. Las estructuras while y do-while son estructuras repetitivas de propósito general.

* **for:** Esta estructura permite realizar repeticiones cuando se conoce el número de elementos que se quiere recorrer. La sintaxis que generalmente se usa es la siguiente:

**for (valor inicial ; condición; incremento o decremento)**

**{ Enunciados }**

La estructura for ejecuta 3 acciones básicas antes o después de ejecutar el bloque de código. La inicialización, en la cual se pueden definir variables e inicializar sus valores. Después la expresión lógica, la cual se evalúa y, si ésta es verdadera, ejecuta el bloque de código, si no se cumple se continúa la ejecución del programa. Por último el conjunto de operaciones que se realizan cada vez que termina de ejecutarse el bloque de código y antes de volver a validar la expresión lógica.

* **Define:** Conocida también como constantes simbólicas. Su sintaxis es la siguiente:

**#define <nombre> <valor>**

Al definir la constante simbólica cada vez que aparezca el nombre en el programa se cambiará por el valor definido. El valor puede ser numérico o puede ser texto.

**DESARROLLO DE ACTIVIDADES:**

Se realizaron 4 programas, de los cuales dos fueron creados en la terminal y dos en Dev-C++. Las estructuras principales en estos programas fueron las de repetición: while, do-whale y for. En todos los programas se definió P como printf y S como scanf.

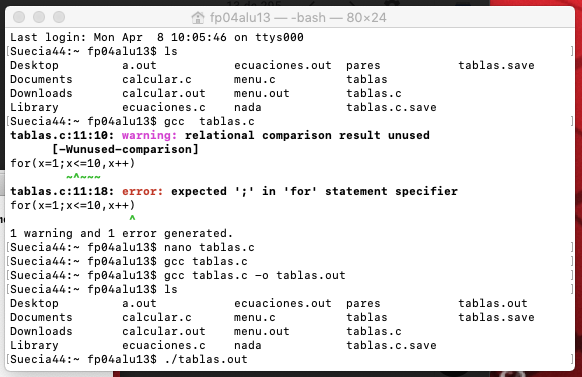
**PROGRAMA 1** (Terminal)

Este programa imprime las tablas de multiplicar del 1 al 10, utilizando la estructura for para realizar las repeticiones de y , x. No se solicita ninguna variable.



#define

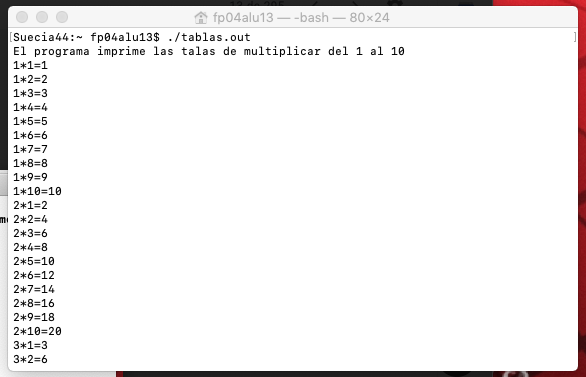
for



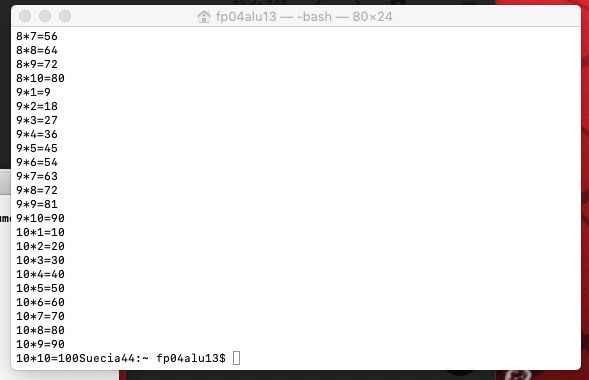
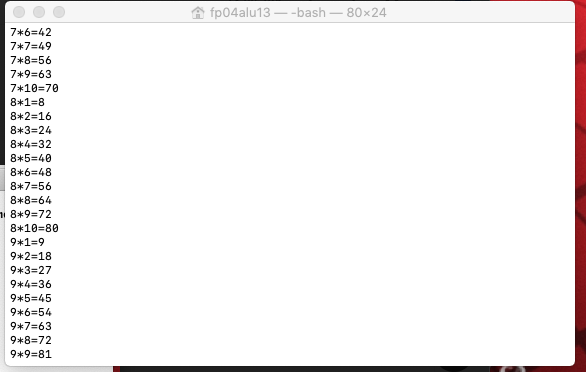
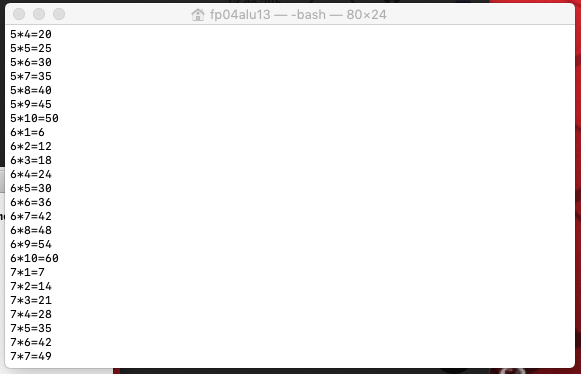
Compilar

Cambiar archivo

Se ejeecuto el programa para comprovar si funcionaba correctamente.

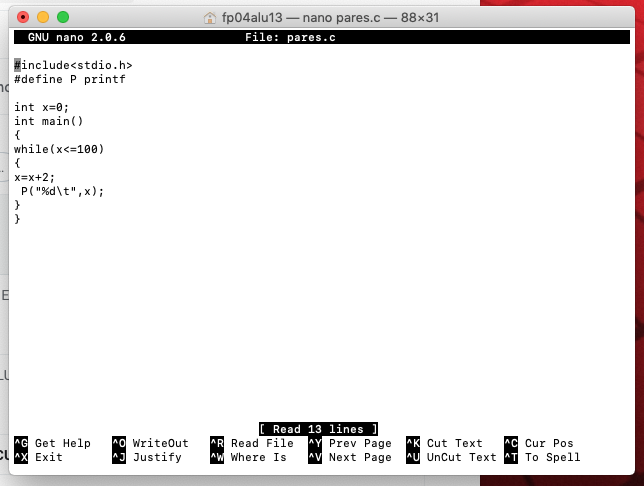


Ejecutar

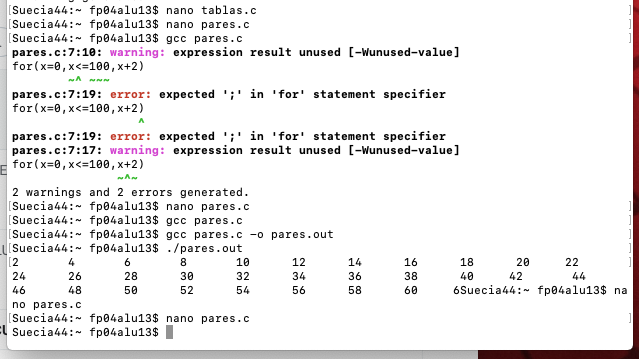


PROGRAMA 2: (Terminal)

Este programa imprime los números pares del 1 al 100, haciendo uso de while para repetir y validar la condición.



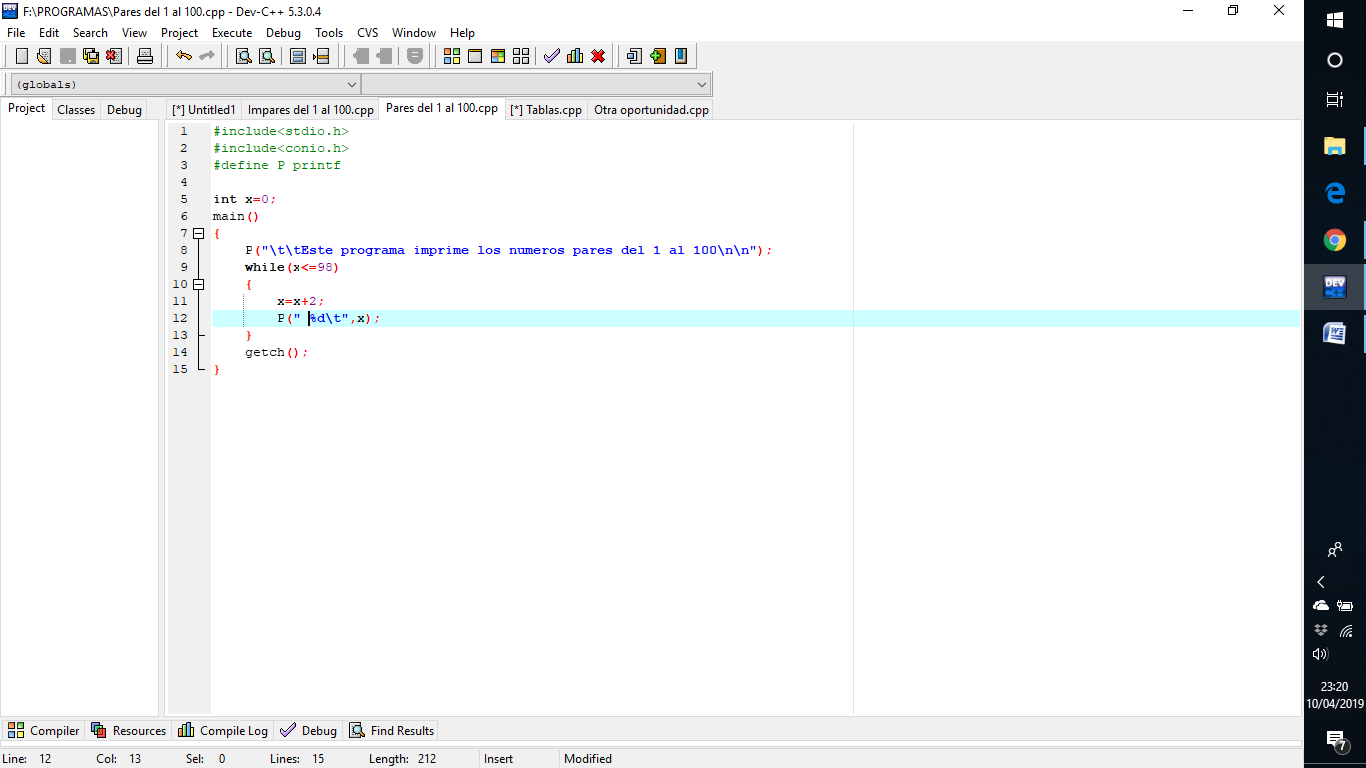
while(condición)



Cambio de archivo y ejecución

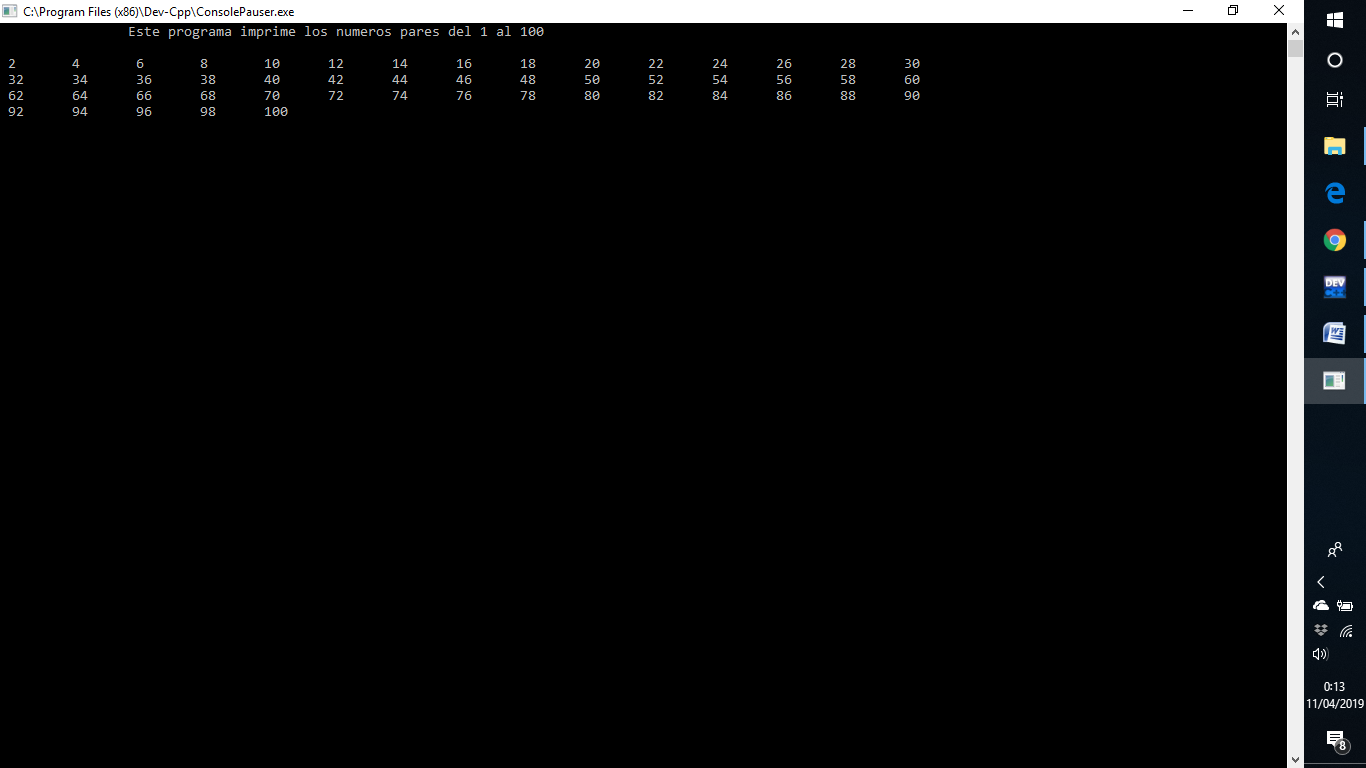
Compilar

En este programa ocurrió un inconveniente, se observo que los números solo se imprimían hasta el 60. por tal motivo se corroboroel programa en Dev-C++, con un cambio .



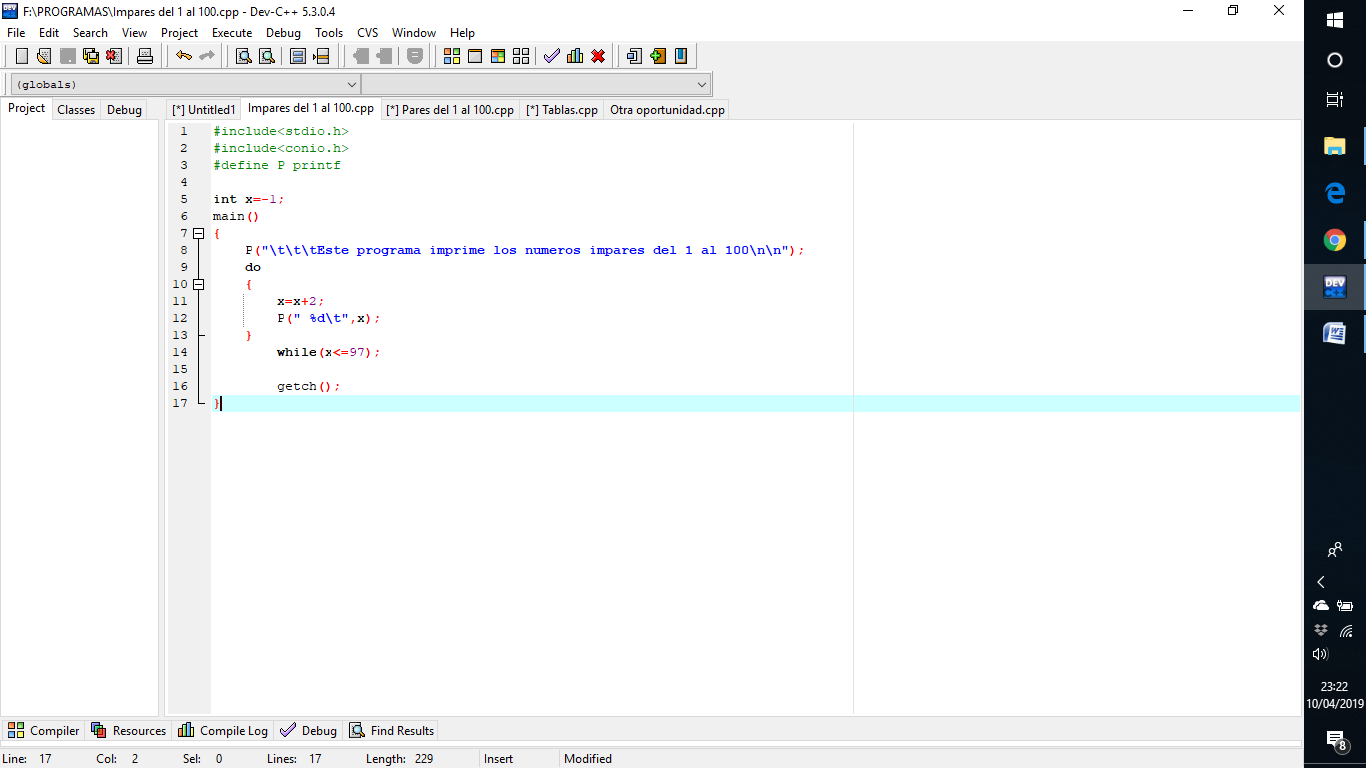
while(condición)

Ejecución.



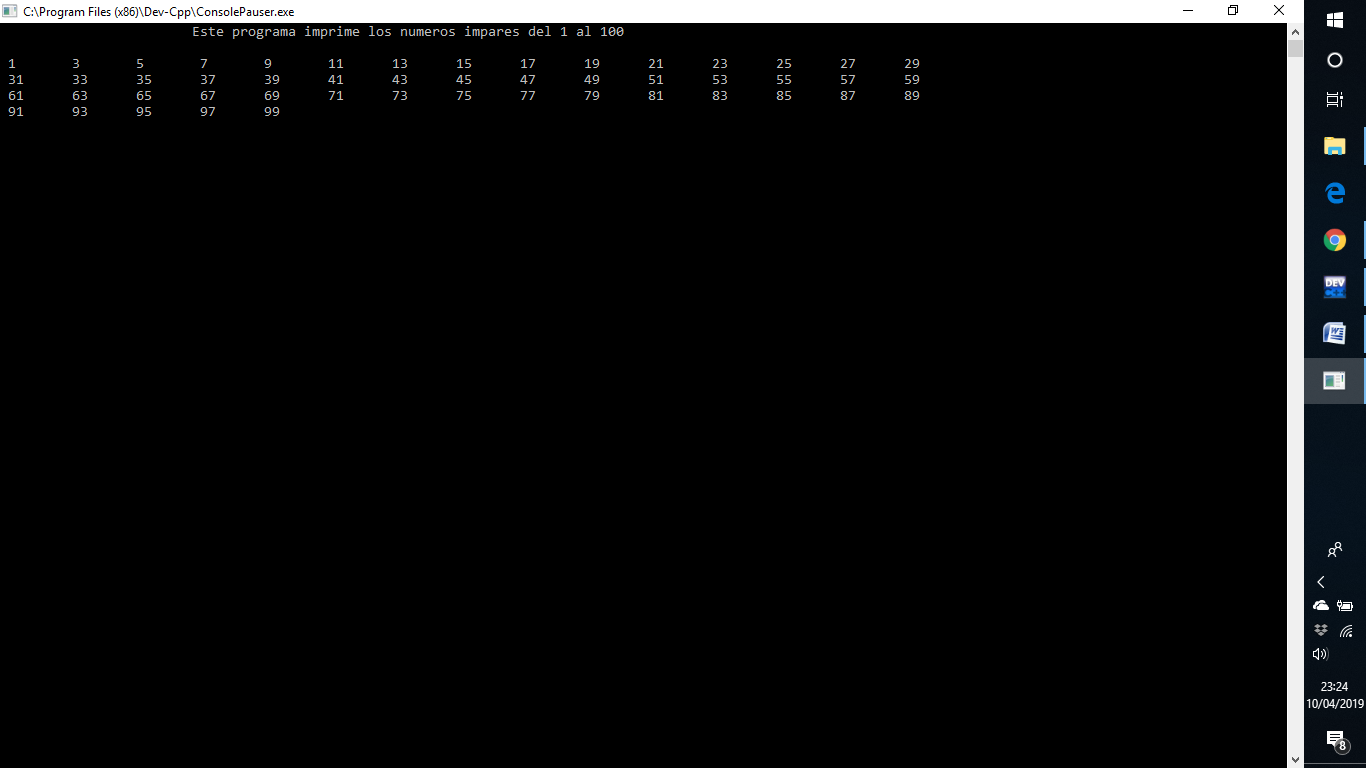
PROGRAMA 3: (Dev-C++)

Este programa imprime los números impares presentes en el rango de 1 a 100. En esta ocasión se efectuó el comando do-while para la condición



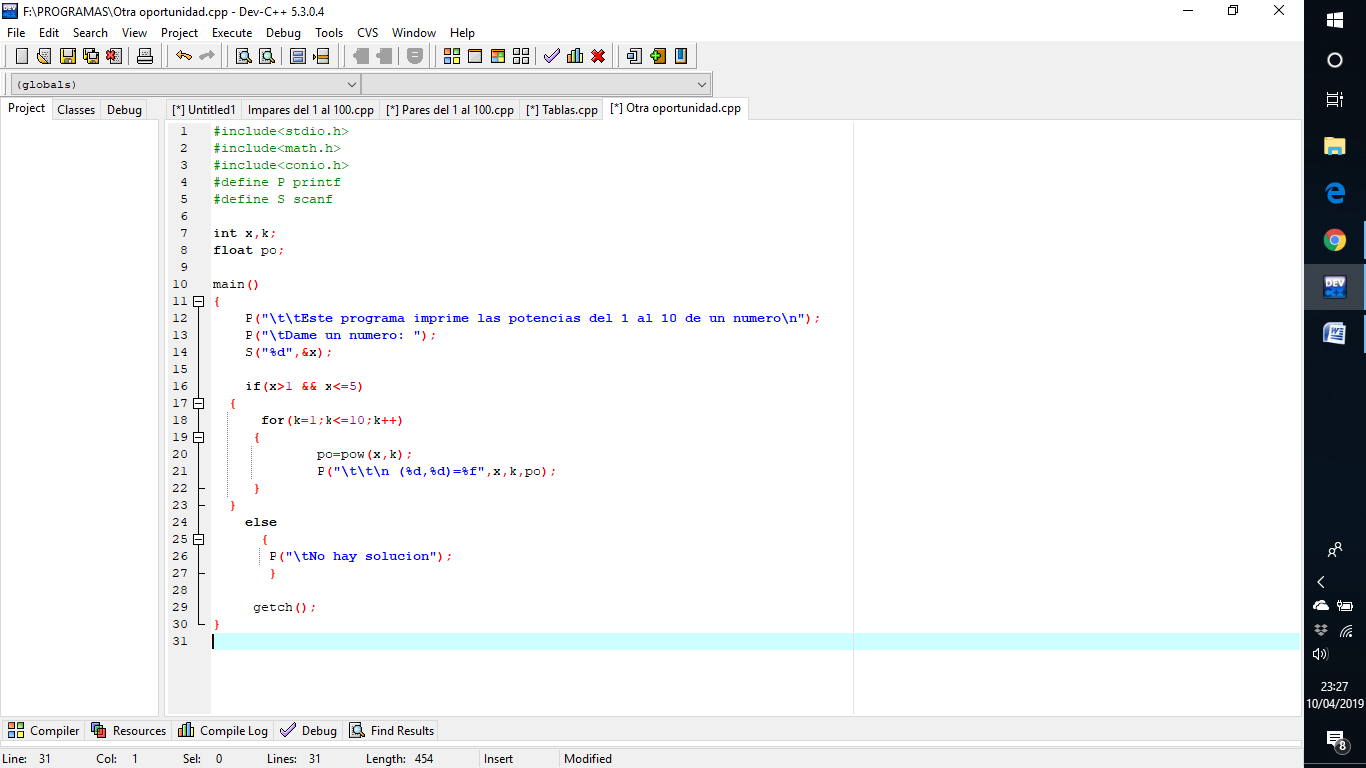
do-while

Se compilo y ejecuto el programa, comprobando que operaba correctamente.



PROGRAMA 4: (Dev-C++)

Este programa imprime las potencias 1 a 10 de un númer dado por el usuario, este deberá ser entre 1 y 5. En esta ocacuion se utilizo un for e if-else como estructura de selección. También se manejo float como tipo de dato y pow como operación.



float

for

If-else

Se compilo y ejecuto. Se produjeron dos ejecuciones para corroborar que la condición se respetaba.



La condición es falsa



La condición es verdadera

**CONCLUCIONES:**

En esta práctica se hizo uso de las estructuras de repetición, las cuales son de gran ayuda para la programación, ya que reducen el tiempo de trabajó empleado y la cantidad de contenido de texto, por ejemplo, en lugar de escribir 100 veces un numero o las misma operación, simplemente se utiliza un for, while o do-while y el programa entiende que es lo que se está solicitando. Es un de las más importantes herramientas que tiene el lenguaje en C por su rapidez y su manera cíclica que poseen. El desarrollo de los programas me ayudaron a comprender algo que aun no entendía, la interrogante de ¿Cuál es la diferencia entre while y do-whilw? Concluyo que while primero valida la condición y dependiendo si es verdadera o falsa realiza la operación y do-while ejecuta la condición y después la valida. Esta última siempre lleva “;” mientras que la primera no, si se hace lo contrario, al momento de compilar el programa marcara errores.