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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
|  | RODRIGUEZ ESPINO CLAUDIA |
| *Asignatura:* | FUNDAMENTOS DE PROGRAMACIÓN |
| *Grupo:* | 1104 |
| *No de Práctica(s):* | PRÁCTICA 9 |
| *Integrante(s):* | VALENCIA MEDINA ISAC |
|  |  |
|  |  |
| *Semestre:* | PRIMER SEMESTRE 2019-1 |
| *Fecha de entrega:* | 08/OCTUBRE/2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

GUÍA PRÁCTICA DE ESTUDIO 09: Estructuras de repetición

**OBJETIVO:**

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.

**INTRODUCCIÓN:**

Las estructuras de repetición son las llamadas estructuras cíclicas, iterativas o de bucles. Permiten ejecutar un conjunto de instrucciones de manera repetida (o cíclica) mientras que la expresión lógica a evaluar se cumpla (sea verdadera).

En lenguaje C existen tres estructuras de repetición: while, do-while y for. Las estructuras while y do-while son estructuras repetitivas de propósito general.

**Problema 1: Factorial**

**#include <conio.h>**

**#include <stdio.h>**

**#define pr printf**

**#define s scanf**

**double n,p;**

**main(){**

**pr("Problema 5: Factorial de un n%cmero\n",163);**

**pr("Dame un n%cmero del 1 al 20: ",163);**

**s("%lf",&n);**

**if(n<=20){**

**p=n\*(n-1);**

**n=n-2;**

**while(n>0){**

**p=p\*n;**

**n--;**

**}**

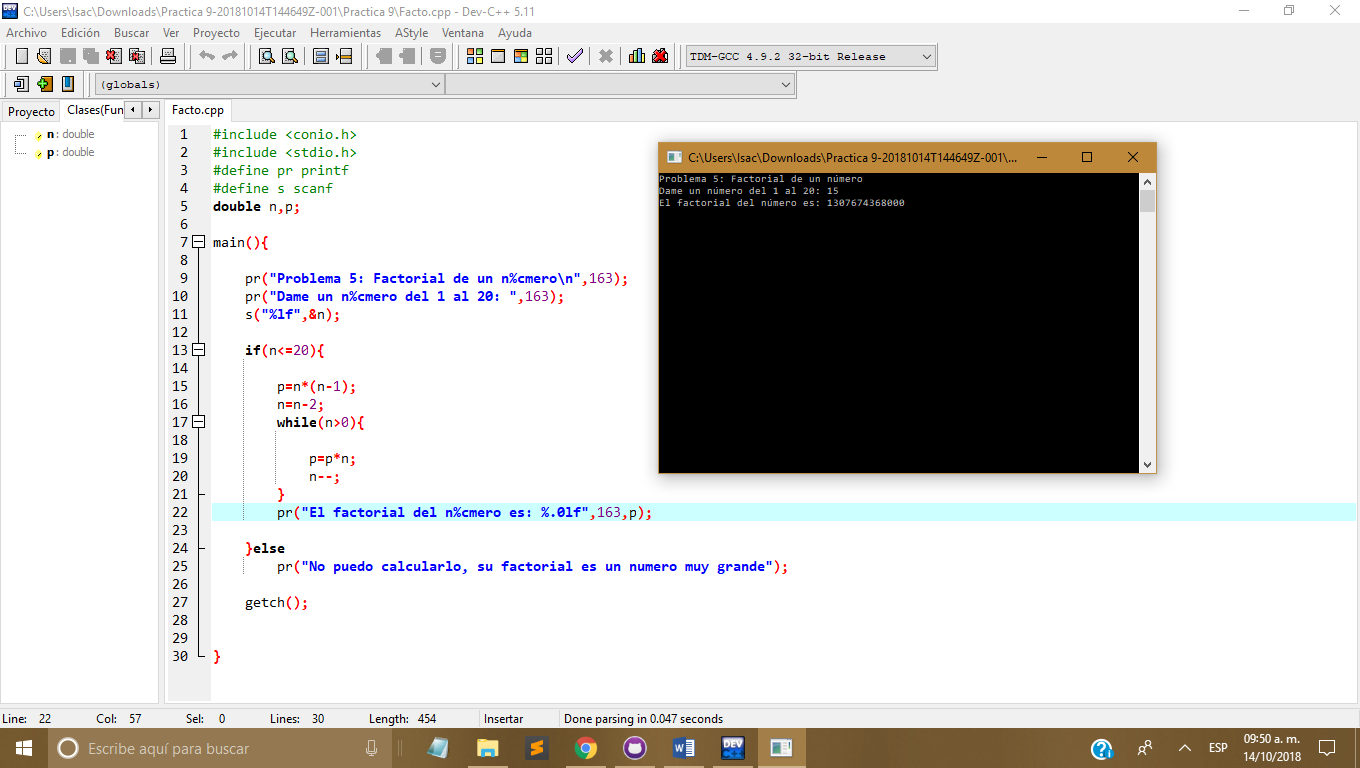
**pr("El factorial del n%cmero es: %.0lf",163,p);**

**}else**

**pr("No puedo calcularlo, su factorial es un numero muy grande");**

**getch();**

**}**



**Problema 2: Números del 1 al 100 (for,while,do while)**

**#include <stdio.h>**

**#include <conio.h>**

**//Biblioteca para utilizar la función Sleep**

**#include <windows.h>**

**#define p printf**

**int x;**

**main(){**

**for(x=1;x<=100;x++)**

**{**

**p("%d\t",x);**

**//Función Sleep que ayuda a mostrar**

**//el resultado de forma pausada**

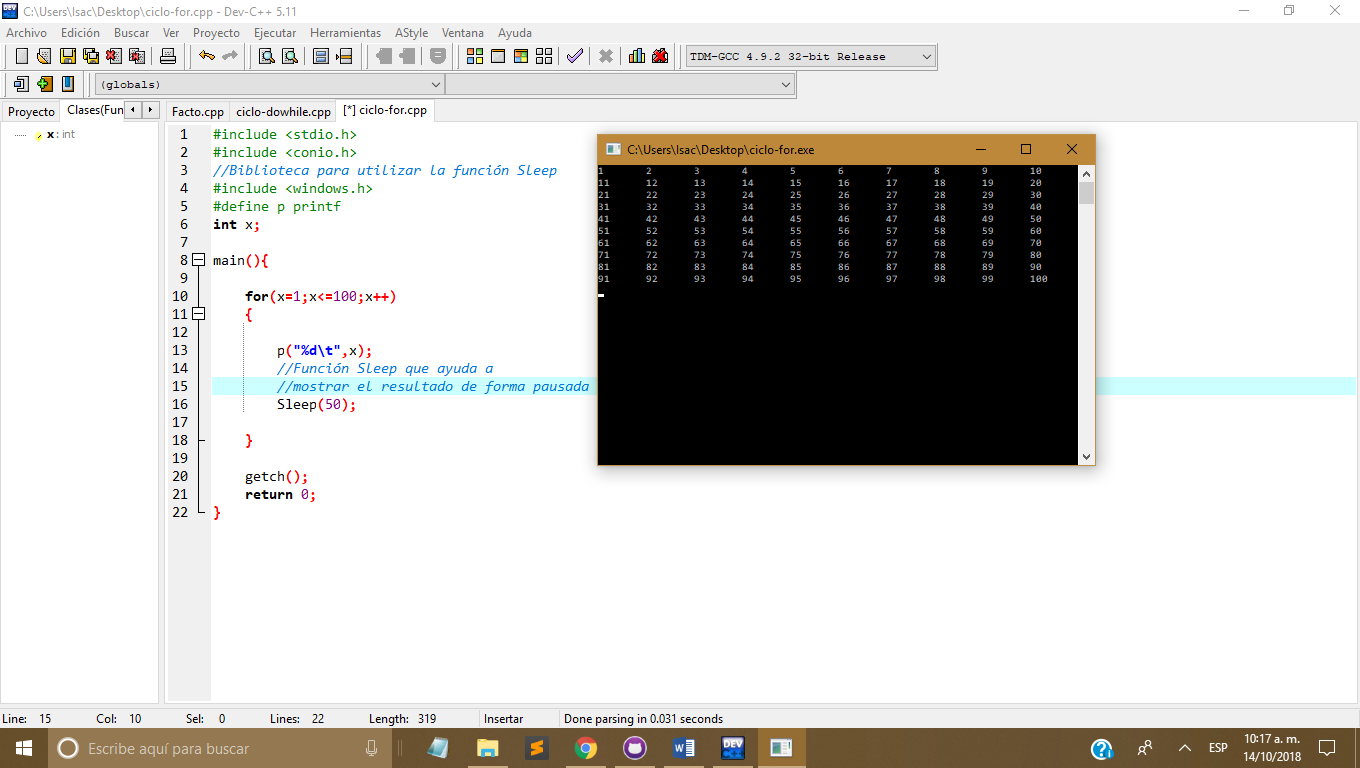
**Sleep(50);**

**}**

**getch();**

**return 0;**

**}**



**#include <stdio.h>**

**#include <conio.h>**

**//Biblioteca para utilizar la función Sleep**

**#include <windows.h>**

**#define p printf**

**int x;**

**main(){**

**x=1;**

**while(x<=100)**

**{**

**p("%d\t",x);**

**//Función Sleep que ayuda a mostrar**

**//el resultado de forma pausada**

**Sleep(50);**

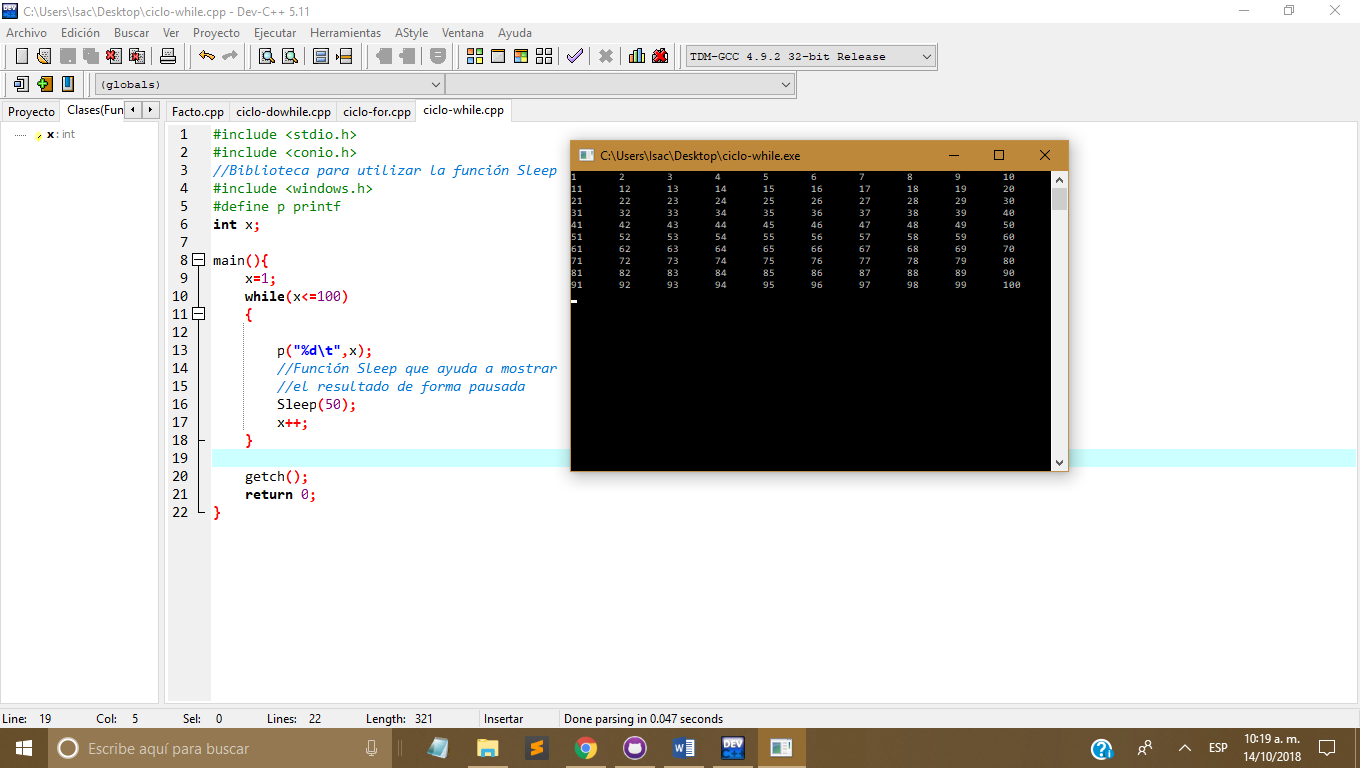
**x++;**

**}**

**getch();**

**return 0;**

**}**



**#include <stdio.h>**

**#include <conio.h>**

**//Biblioteca para utilizar la función Sleep**

**#include <windows.h>**

**#define p printf**

**int x;**

**main(){**

**x=1;**

**do**

**{**

**p("%d\t",x);**

**//Función Sleep que ayuda a mostrar**

**//el resultado de forma pausada**

**Sleep(50);**

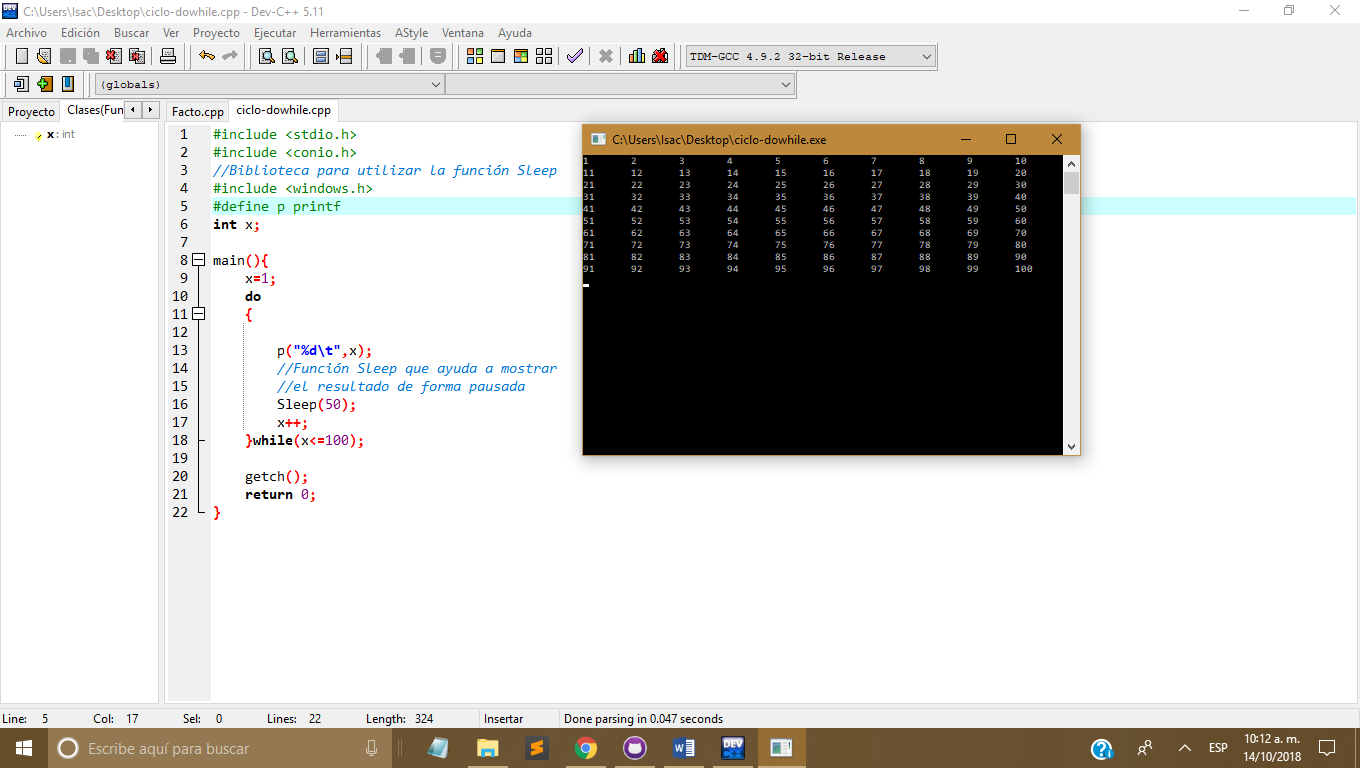
**x++;**

**}while(x<=100);**

**getch();**

**return 0;**

**}**



**Problema 3: Potencia de un número**

**#include <stdio.h>**

**#include <conio.h>**

**#define p printf**

**#define s scanf**

**double n,ptc,x;**

**main(){**

**p("Problema: Potencias de un n%cmero\n",163);**

**p("Dame un n%cmero del 1 al 10: ",163);**

**s("%lf",&n);**

**if((n>0) && (n<=10))**

**{**

**p("%.0lf^0=0\n",n);**

**x=1;**

**ptc=n\*x;**

**p("%.0lf^%.0lf=%.0lf\n",n,x,ptc);**

**x++;**

**while(x<=10){**

**ptc=ptc\*n;**

**p("%.0lf^%.0lf=%.0lf\n",n,x,ptc);**

**x++;**

**}**

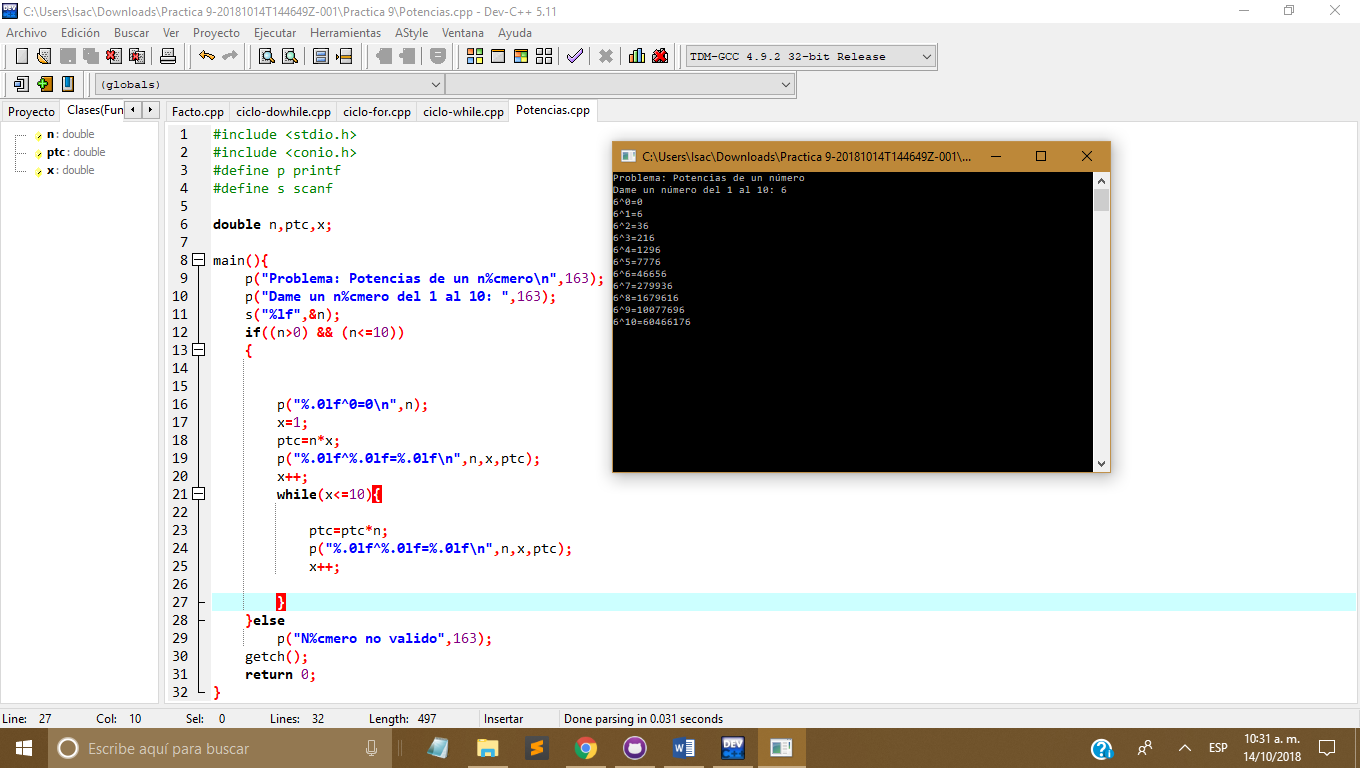
**}else**

**p("N%cmero no valido",163);**

**getch();**

**return 0;**

**}**



**Problema 4: Tablas del 1 al 10**

**#include <stdio.h>**

**#include <conio.h>**

**//Biblioteca para la función Sleep**

**#include <windows.h>**

**int main(){**

**int x,y;**

**printf("\a----- Tablas de multiplicar del 1 al 10 -----\n");**

**for(x=1;x<=10;x++){**

**printf("La tabla de multiplicar del %d es:\n",x);**

**for(y=1;y<=10;y++){**

**printf("%d x %d = %d\n", x, y, x\*y);**

**//Función Sleep para mostrar pusadamente**

**//la ejecución del programa**

**Sleep(100);**

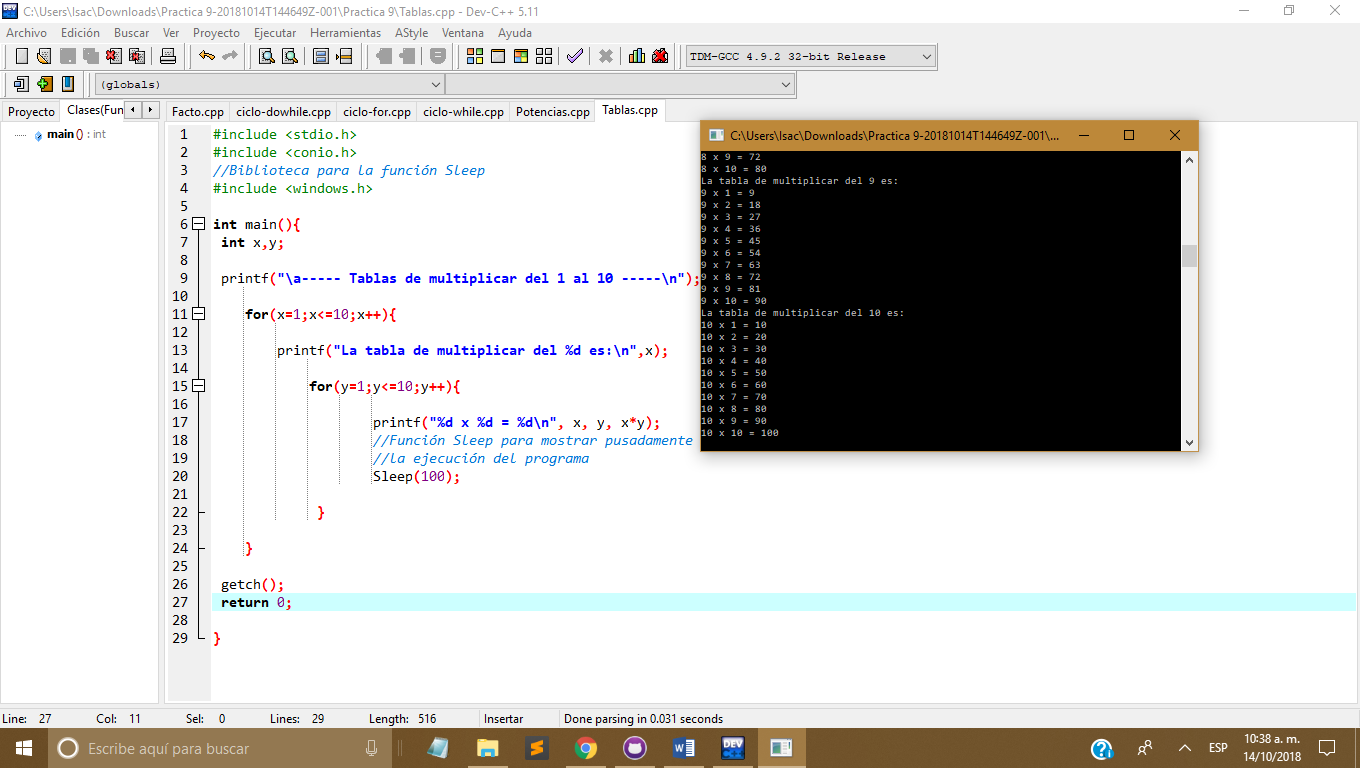
**}**

**}**

**getch();**

**return 0;**

**}**



**CONCLUSIÓN:**

En la práctica realizada, nuevamente aprendimos la importancia de utilizar ciclos en nuestro programa con el objetivo de mejorar el rendimiento y reducir el espacio de código empleado. Lo ciclos repetitivos son una herramienta muy poderosa si queremos procesar varios datos sin la necesidad de hacerlo uno por uno, hay que destacar que tenemos tres opciones para poder implementar un ciclo repetitivo, está el for, el while y el do while, por lo general los tres funcionarían para lo mismo, pero dependiendo del problema a veces es más práctico utilizar uno u otro o incluso una combinación entre ellos. Como había mencionado los ciclos repetitivos ayudan a generar programas cortos, pero con mucha información a mostrar, en lo personal creo que son la herramienta más potente de un lenguaje de programación, en la actualidad la mayoría de los programas se encargan de procesar muchos datos y siempre será más sencillo procesarlo todo de forma implícita mediante el ciclo repetitivo evitando generar un código muy largo y poco eficiente. Finalmente, con esta práctica se logró la implementación eficiente de los tres tipos de ciclo repetitivo: for, while y do while, resolviendo problemas desde imprimir la serie de números del 1 al 100 hasta obtener el factorial de un número problema que indudablemente sin un ciclo repetitivo sería muy complicado resolver.