|  |  |  |
| --- | --- | --- |
|  | **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 12 |
| *Integrante(s):* | VALENCIA MEDINA ISAC |
|  |  |
|  |  |
| *Semestre:* | PRIMER SEMESTRE 2019-1 |
| *Fecha de entrega:* | 29/OCTUBRE/2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

GUÍA PRÁCTICA DE ESTUDIO 12: FUNCIONES

**OBJETIVO:**

Elaborar programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

**ACTIVIDADES:**

Implementar en un programa en C la solución de un problema dividido en funciones.

Elaborar un programa en C que maneje argumentos en la función principal.

En un programa en C, manejar variables y funciones estáticas.

**INTRODUCCIÓN:**

Como ya se mencionó, un programa en lenguaje C consiste en una o más funciones. C permite tener dentro de un archivo fuente varias funciones, esto con el fin de dividir las tareas y que sea más fácil la depuración, la mejora y el entendimiento del código.

En lenguaje C la función principal se llama main. Cuando se ordena la ejecución del programa, se inicia con la ejecución de las instrucciones que se encuentran dentro de la función main, y ésta puede llamar a ejecutar otras funciones, que a su vez éstas pueden llamar a ejecutar a otras funciones, y así sucesivamente.

PROBLEMA 1. Cubo de cualquier número (con parámetro)

#include <stdio.h>

#include <conio.h>

//Prototipo de Función Cubo

int cubo(int x);

int n,cub;

main(){

printf("\tFunci%cn Cubo\n",162);

printf("Dame un n%cmero: ",163);

scanf("%d",&n);

cub=cubo(n);

printf("El cubo del n%cmero %d es : %d",163,n,cub);

getch();

return 0;

}

int cubo(int x){

static result;

int i;

result=1;

for(i=1;i<=3;i++)

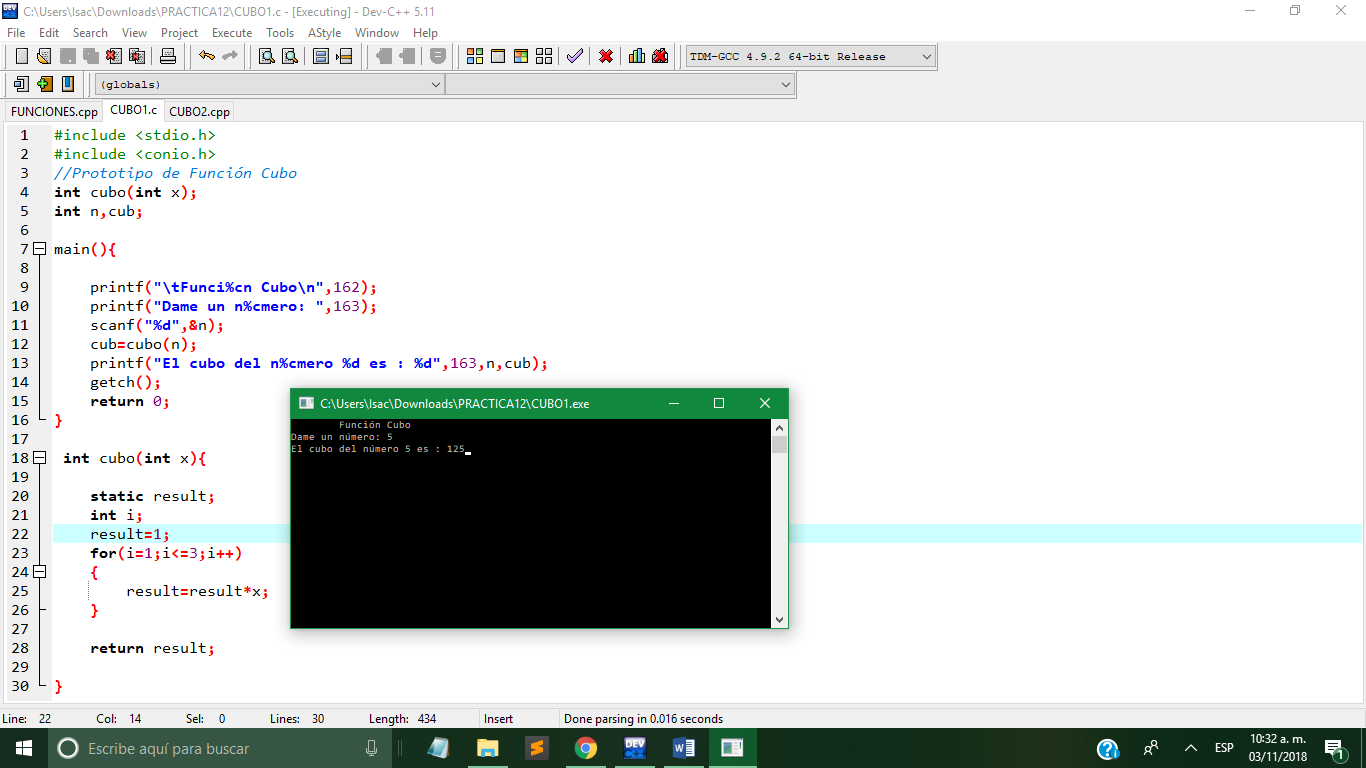
{

result=result\*x;

}

return result;

}



PROBLEMA 2. Cubo de cualquier número (sin parámetro)

#include <stdio.h>

#include <conio.h>

void cubo();

main(){

cubo();

getch();

return 0;

}

void cubo(){

int n;

int i;

int result;

result=1;

printf("Dame un n%cmero: ",163);

scanf("%d",&n);

for(i=1;i<=3;i++)

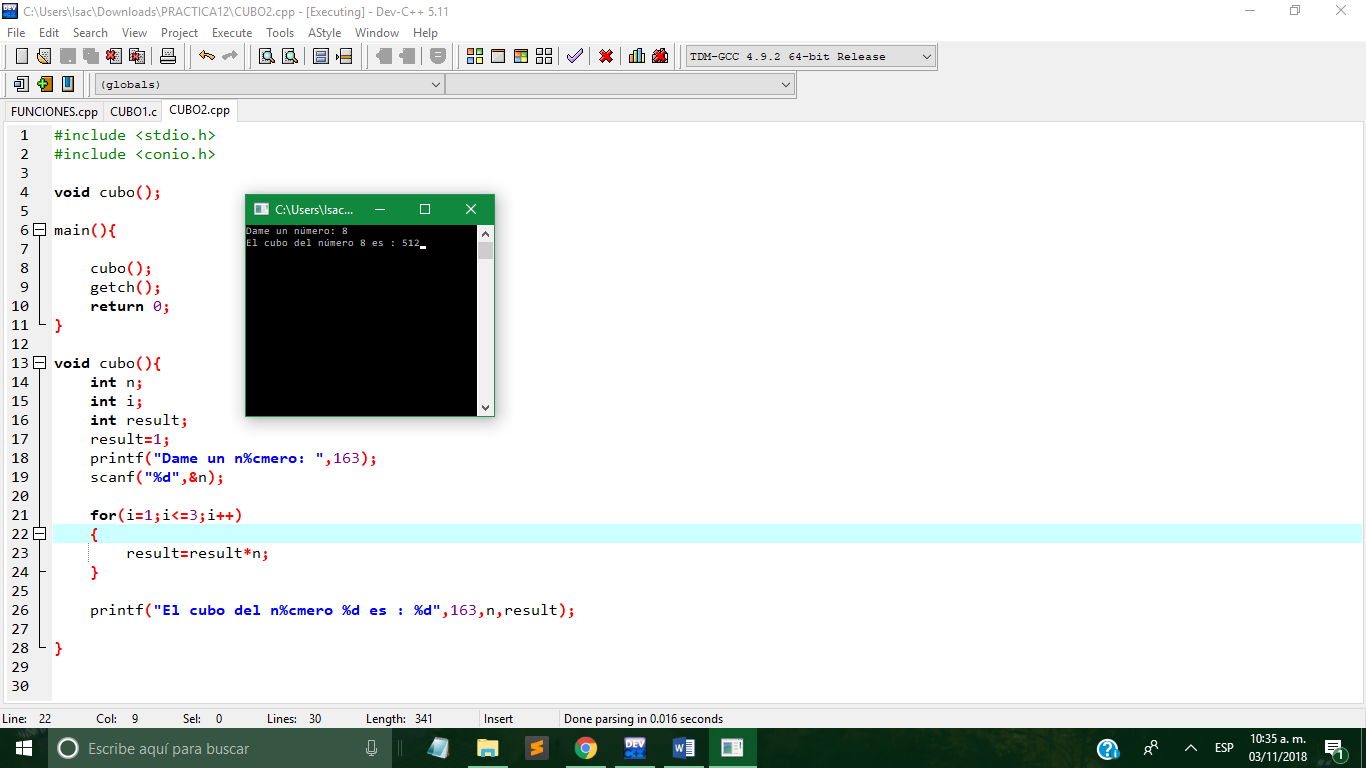
{

result=result\*n;

}

printf("El cubo del n%cmero %d es : %d",163,n,result);

}



PROBLEMA 3. Calculadora con Funciones

#include <stdio.h>

#include <conio.h>

#include <math.h>

#define p printf

#define s scanf

int suma(int n1,int n2);

int resta(int n1,int n2);

int multiplicacion(int n1,int n2);

int division(int n1,int n2);

int a,b,otc,result;

main(){

p("\tCalculadora con Funciones\n");

p("1.-SUMA\n");

p("2.-RESTA\n");

p("3.-MULTIPLICACION\n");

p("4.-DIVISION\n");

p("Selecciona una opci%cn: ",162);

s("%d",&otc);

switch(otc){

case 1:

p("Dame el 1er n%cmero: ",163);

s("%d",&a);

p("Dame el 2do n%cmero: ",163);

s("%d",&b);

result=suma(a,b);

p("La suma de %d + %d es: %d",a,b,result);

break;

case 2:

p("Dame el 1er n%cmero: ",163);

s("%d",&a);

p("Dame el 2do n%cmero: ",163);

s("%d",&b);

result=resta(a,b);

p("La resta de %d - %d es: %d",a,b,result);

break;

case 3:

p("Dame el 1er n%cmero: ",163);

s("%d",&a);

p("Dame el 2do n%cmero: ",163);

s("%d",&b);

result=multiplicacion(a,b);

p("La multiplicacion de %d x %d es; %d",a,b,result);

break;

case 4:

p("Dame el 1er n%cmero: ",163);

s("%d",&a);

p("Dame el 2do n%cmero: ",163);

s("%d",&b);

result=division(a,b);

p("La division de %d / %d es: %d",a,b,result);

break;

default:

p("Opci%cn no valida",162);

break;

}

getch();

return 0;

}

//Funcion de la suma

int suma(int n1,int n2){

return n1+n2;

}

//Funcion de la resta

int resta(int n1,int n2){

return n1-n2;

}

//Funcion de la multiplicación

int multiplicacion(int n1,int n2){

return n1\*n2;

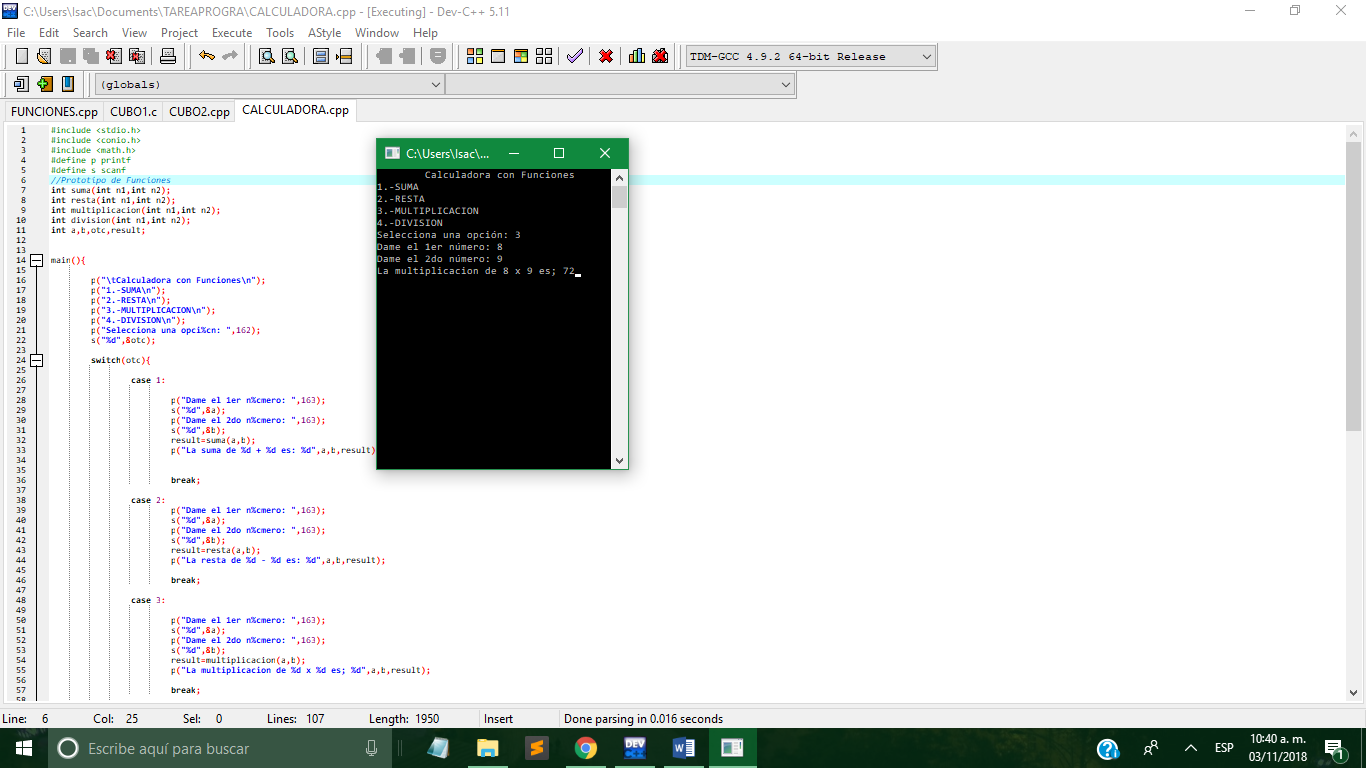
}

//Funcion de la division

int division(int n1,int n2){

return n1/n2;

}



PROBLEMA 4. Dos Funciones y main()

#include <stdio.h>

#include <conio.h>

//Prototipo de funciones

int leeNum();

int llamaNum();

int numero;

main(){

numero=llamaNum()\*100;

printf("\nEl n%cmero multiplicado por 100 es: %d",163,numero);

getch();

return 0;

}

int leeNum(){

int n1;

printf("\nEscribe un n%cmero: ",163);

scanf("%d",&n1);

return n1;

}

int llamaNum(){

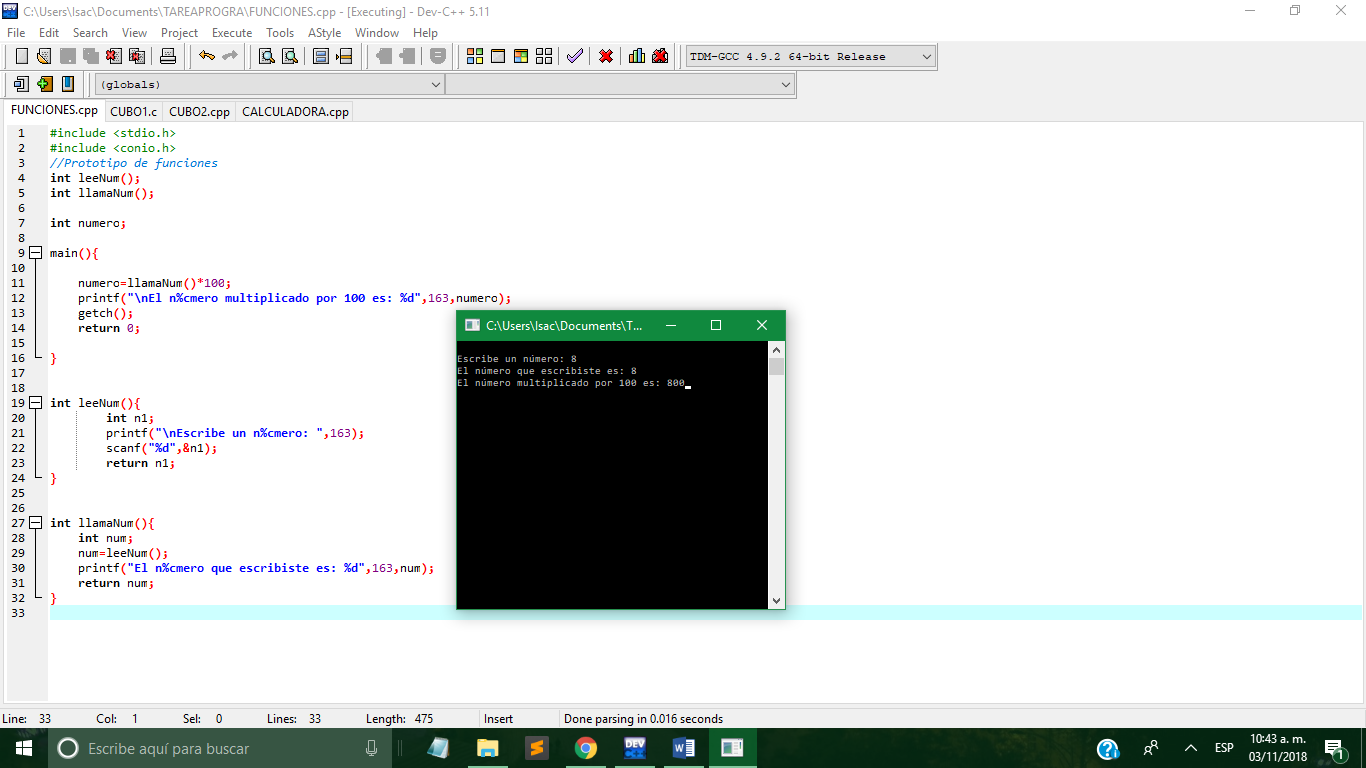
int num;

num=leeNum();

printf("El n%cmero que escribiste es: %d",163,num);

return num;

}



CONCLUSIÓN:

El uso de funciones en el desarrollo de un programa habla bien de quien lo desarrolla, considero que representa un conocimiento más firme sobre el lenguaje y permite realizar tareas más complejas teniendo un correcto orden en el programa. Las funciones nos permiten simplificar nuestro código fuente ya que ayudan a dividir nuestro programa en pequeñas partes reutilizables, esto último muy importante, las funciones las podemos usar en varias partes de nuestro programa lo que reduce significativamente las líneas de código al no tener que repetir toda la función una y otra vez, lo único que se necesita es mandar llamar a la función y listo. En la declaración de funciones como aprendimos en esta práctica es recomendable primero hacer el prototipo de la función, el cual representa el nombre de la función, el tipo de valor de retorno y los parámetros que requiere, posteriormente hacer todo el desarrollo del main() y al final o incluso antes, escribir nuestras funciones ya con el código que necesitaran, de esta forma nuestro programa estará más ordenado y fácil de entender. Finalmente es importante señalar que hay funciones que requieren de parámetros, es decir, valores de alguna otra variable que la función utilizara en su proceso, y que estos pueden ser de los tipos de datos que conocemos: int, char, float, e incluso arreglos, por otra parte, se pueden hacer funciones que no necesiten parámetros sino que dentro de ellas se realiza todo el proceso y solo hay que declararlas en la función main para que se ejecuten. Otro dato importante es que cuando declaras variables dentro de una función, estas se vuelven variables locales las cuales se utilizan solo dentro la función y una vez fuera dejan de existir, esto representa una ventaja al poder ahorrar memoria.