



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Claudia Rodriguez Espino

Asignatura: Fundamentos de Programación

Grupo: 1102

No de Práctica(s): Práctica número 7

Integrante(s): Alejandro Nava Cruz

Semestre: 2018-1

Fecha de entrega: 1 de Octubre del 2017

Observaciones:

CALIFICACIÓN: _____

Practica 7: Fundamentos de Lenguaje C

Objetivo:

Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

Desarrollo:

Como en todo desarrollo, antes de hacer un programa, se requiere plantear el problema y analizarlo, esto es, analizar los datos de la entrada y salida deseada, además de que se haya diseñado un algoritmo que resuelva el problema de la mejor manera (clara y precisa), y que se ha representado el diagrama de flujo y/o pseudocódigo, para después proceder con la codificación.

La codificación se puede realizar desde cualquier lenguaje de programación estructurada, como Pascal, Python, Fortran o PHP.

Antes de continuar, debemos de saber que C es un lenguaje de programación de propósito general que ofrece como ventajas economía de expresión, control de flujo y estructuras de datos con conjunto de operadores. Por otro lado, C es un lenguaje compilado, es decir, existe un programa llamado compilador que a partir de este lenguaje genera un código objeto mejor conocido como ejecutable.

Para crear un programa en C se siguen tres etapas principales: edición, compilación y ejecución.

-Edición:

-Compilación: A partir del código fuente (lenguaje C) se genera el archivo en lenguaje máquina (se crea el programa objeto o ejecutable).

-Ejecución: El archivo en lenguaje máquina se puede ejecutar en la arquitectura correspondiente.

Un programa en c consiste en una o más funciones, de las cuales una de ellas debe llamarse main() y que es la función principal.

Aquí entraron conceptos nuevos para nosotros, del cual uno de ellos trata sobre agregar comentarios, ya que es una buena práctica para documentar un programa. Existen dos tipos de comentarios, por línea y por bloque, representados por símbolos "//" los que solo representaran un renglón y con "/*" "*/", dando inicio y final respectivamente, este tipo de comentario puede abarcar más de una línea.

Este es un ejemplo de un programa con comentarios:

```
#include <stdio.h>
```

```
main() {  
  
    // Este código compila y ejecuta  
    /* pero no muestra salida alguna  
    debido a que un comentario  
    ya sea por línea o por bloque */  
    // no es tomado en cuenta al momento  
    // de compilar el programa,  
    /* sólo sirve como documentación en el */  
    /*  
    código fuente  
    */  
}
```

Otra de las nuevas definiciones son los llamados modificadores de alcance, los cuales son “const” y “static”.

El modificador de “const” impide que la variable cambie su valor durante toda la ejecución, es decir, crea constantes, mientras que el de “static” indica que la variable permanece en memoria desde su creación y durante toda la ejecución del programa, dicho de otra manera, permanece estática en la memoria.

Un ejemplo del uso de estos es:

CONST:

```
const int NUM_MAX = 1000;
```

```
const double PI = 3,141592653589793 3;
```

STATIC:

```
static int num;
```

```
static float resultado = 0;
```

Otra nueva función que se vio fue el uso del moldeo o cast, el cual se usa para que el resultado de una operación entre dos tipos de datos iguales dé un dato diferente.

```
#include<stdio.h>
```

```
Int cinco = 5, dos = 2;
```

```
main() {
```

```
    double r = (double)cinco/dos
```

```
printf("El resultado es: %f", r)
```

```
}
```

Ejercicios y actividades hechas en casa

Actividad 1. Suma de 2 números

Código fuente:

```
#include<stdio.h>

int a,b,c=0;

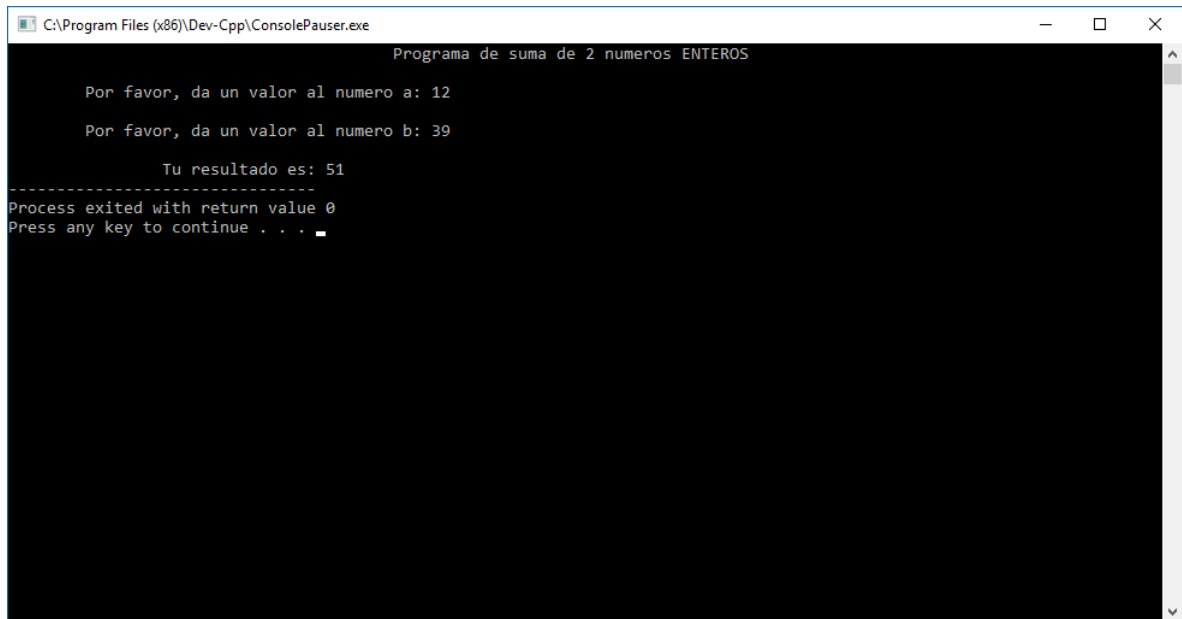
main()
{
    printf("\t\t\t\t\tPrograma de suma de 2 numeros ENTEROS\n");

    printf("\n\t\tPor favor, da un valor al numero a: ");
    scanf("%d",&a);

    printf("\n\t\tPor favor, da un valor al numero b: ");
    scanf("%d",&b);

    c=(a+b);

    printf("\n\t\t\tTu resultado es: %d", c);
}
```



Actividad 2. Área del círculo (Modificado para uso del modificador “const”)

Código fuente:

```
#include<stdio.h>
#include<math.h>
const double pi = 3.1415926536;

float r,b,a;

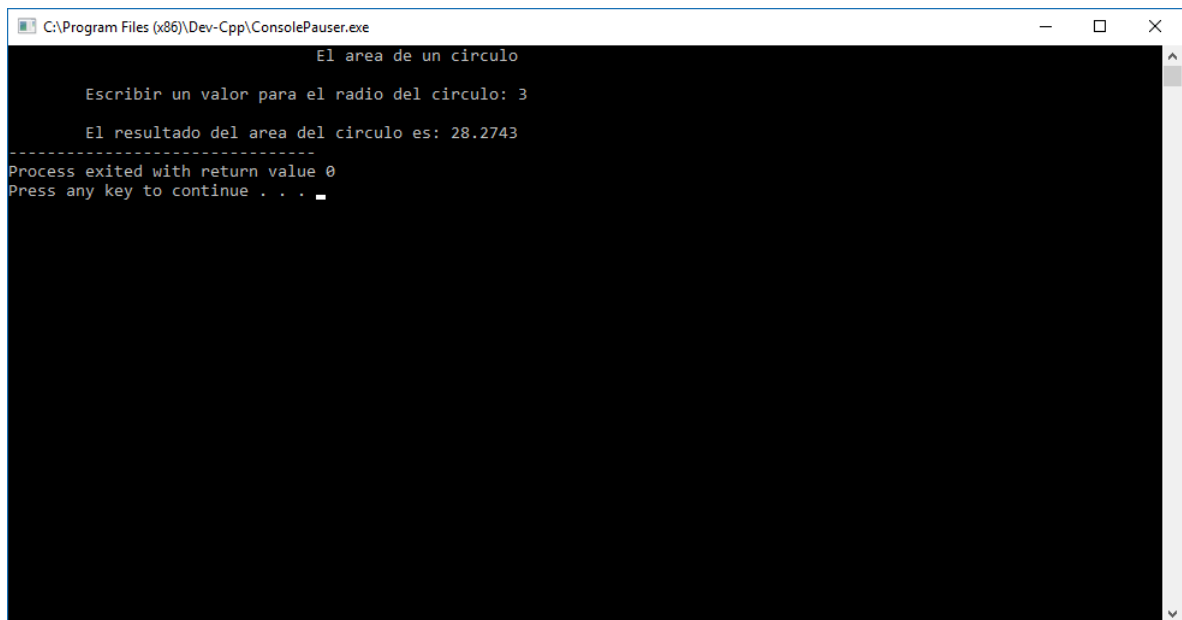
main()
{
    printf("\t\t\t\tEl area de un circulo\n\n");

    printf("\tEscribir un valor para el radio del circulo: ");
    scanf("%f",&r);

    b=pow(r,2);

    a=(pi*b);

    printf("\n\t\tEl resultado del area del circulo es: %.4f",a);
}
```



The screenshot shows a console window titled "C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe". The output of the program is as follows:

```
El area de un circulo

Escribir un valor para el radio del circulo: 3

El resultado del area del circulo es: 28.2743
-----
Process exited with return value 0
Press any key to continue . . .
```

Actividad 3. Promedio de 3 números (Modificado para uso del moldeo)

Código fuente:

```
#include<stdio.h>

int a,b,c,d=0;

main()
{
    printf("\t\t\t\tPrograma para dar el promedio de 3 numeros\n\n");

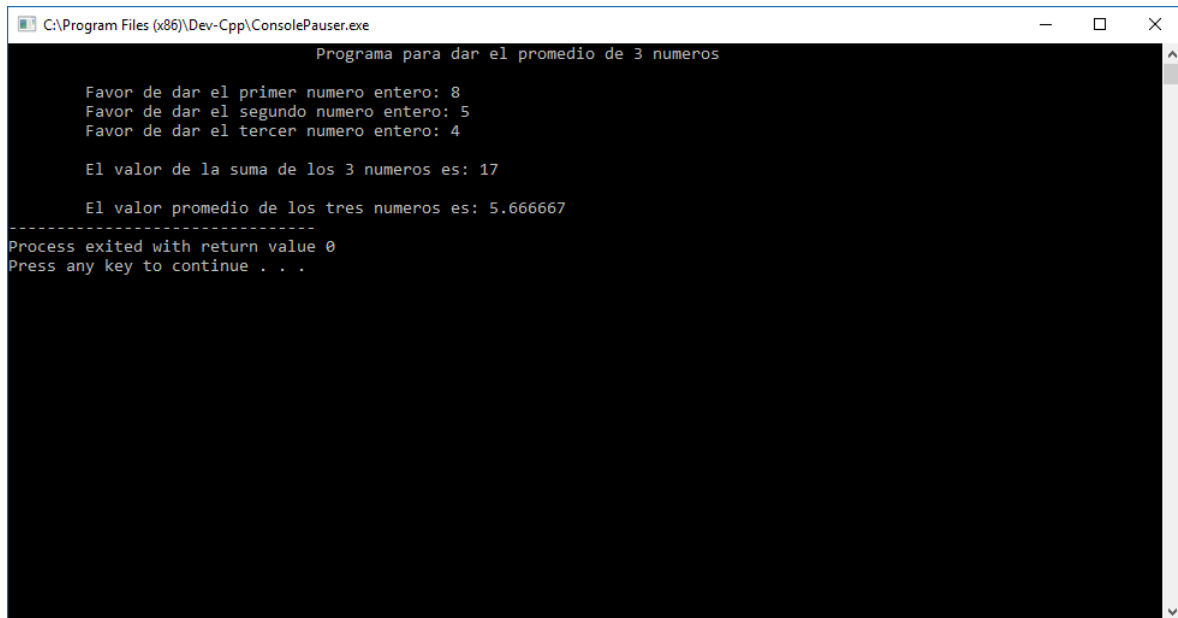
    printf("\tFavor de dar el primer numero entero: ");
    scanf("%d",&a);

    printf("\tFavor de dar el segundo numero entero: ");
    scanf("%d",&b);

    printf("\tFavor de dar el tercer numero entero: ");
    scanf("%d",&c);

    d=(a+b+c);
    printf("\n\tEl valor de la suma de los 3 numeros es: %d", d);

    double e = (double)d/3;
    printf("\n\n\tEl valor promedio de los tres numeros es: %f", e);
}
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Programa para dar el promedio de 3 numeros

Favor de dar el primer numero entero: 8
Favor de dar el segundo numero entero: 5
Favor de dar el tercer numero entero: 4

El valor de la suma de los 3 numeros es: 17

El valor promedio de los tres numeros es: 5.666667
-----
Process exited with return value 0
Press any key to continue . . .
```

Conclusiones:

Para la codificación del programa y como siempre se ha dicho, lo más conveniente para la realización del mismo es planear a lo que se quiere llegar, pasando por la creación del algoritmo, el diagrama o pseudocódigo para al final llegar al código fuente, y para esto se necesita especificar lo que se quiere que el programa realice, aquí entran las funciones, ya que nos permiten de una manera clara y precisa, especificar a lo que se quiere llegar. En el lenguaje C, se llevan a cabo distintos tipos de funciones, algunos que facilitan unas tareas respecto de otras, y otras que simplemente se tienen que hacer como se establecen para que pueda correr el programa.

Bibliografía:

Práctica número 7, Facultad de Ingeniería, en Laboratorios A y B (2017), Sitio Web:
<http://lcp02.fib.unam.mx/>