



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Rodríguez Espino Claudia Ing.

Asignatura: Fundamentos de Programación

Grupo: 1104

No de Práctica(s): 7

Integrante(s): Santa Rosa Ortiz Thelma Jazmín.

*No. de Equipo de
cómputo empleado* 50

Semestre: 1°

Fecha de entrega: 29 de septiembre del 2018

Observaciones:

CALIFICACIÓN: _____

Práctica 7. Fundamentos del Lenguaje C.

Objetivos:

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:

Una vez que un problema dado ha sido analizado, que se ha diseñado un algoritmo que lo resuelva de manera eficiente, y que se ha representado el algoritmo de manera gráfica o escrita se puede proceder a la etapa de codificación.

La codificación se puede realizar en cualquier lenguaje de programación estructurada, como lo son Pascal, Python, Fortran o PHP. En este curso se aprenderá el uso del lenguaje de programación C.




Lenguaje de Programación C: C es un lenguaje de programación de propósito general que ofrece como ventajas economía de expresión, control de flujo y estructuras de datos y un conjunto de operadores.

C es un lenguaje de propósito general basado en el paradigma estructurado. El teorema del programa estructurado, demostrado por Böhm-Jacopini, dicta que todo programa puede desarrollarse utilizando únicamente 3 instrucciones de control:

-  Secuencia
-  Selección
-  Iteración

Por otro lado, C es un lenguaje compilado, es decir, existe un programa (llamado compilador) que, a partir de un código en lenguaje C, genera un código objeto (ejecutable).

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

-  Edición: Se escribe el código fuente en lenguaje C desde algún editor de textos.
-  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 es la principal.

Al momento de ejecutar un programa objeto (código binario), se ejecutarán únicamente las instrucciones que estén definidas dentro de la función principal. La función principal puede contener sentencias, estructuras de control y comentarios. Dentro de las sentencias se encuentran la declaración y/o asignación de variables, la realización de operaciones básicas, y las llamadas a funciones.

Al iniciar el programa se deben agregar todas las bibliotecas que se van a utilizar en el mismo, es decir, funciones externas necesarias para ejecutar el programa. En lenguaje C la biblioteca estándar de entrada y salida está definida en 'stdio.h' (standard in out) y provee, entre otras, funciones para lectura y escritura de datos que se verán a continuación.

Comentarios.

- ✚ // → Para comentarios por línea.
- ✚ /* → Para comentarios por bloque.

Declaración de Variables.

Para declarar variables en C se sigue la siguiente sintaxis:

[modificadores] tipoDeDato identificador [= valor];

ó

tipoDeDato identificador1[= valor], identificador2[= valor];

Tipos de datos:

- ✚ Caracteres: codificación definida por la máquina.
- ✚ Enteros: números sin punto decimal.
- ✚ Flotantes: números reales de precisión normal.
- ✚ Dobles: números reales de doble precisión.

Variables enteras.

<i>Tipo</i>	<i>Bits</i>	<i>Valor Mínimo</i>	<i>Valor Máximo</i>
<i>signed char</i>	8	-128	127
<i>unsigned char</i>	8	0	255
<i>signed short</i>	16	-32 768	32 767
<i>unsigned short</i>	16	0	65 535
<i>signed int</i>	32	-2,147,483,648	2 147 483 647
<i>unsigned int</i>	32	0	4 294 967 295
<i>signed long</i>	64	9 223 372 036 854 775 808	9 223 372 036 854 775 807
<i>unsigned long</i>	64	0	18 446 744 073 709 551 615
<i>enum</i>	16	-32 768	32 767

Variables Reales.

<i>Tipo</i>	<i>Bits</i>	<i>Valor Mínimo</i>	<i>Valor Máximo</i>
<i>float</i>	32	3.4 E-38	3.4 E38
<i>double</i>	64	1.7 E-308	1.7 E308
<i>long double</i>	80	3.4 E-4932	3.4 E4932

Las variables reales poseen signo.: Especificadores para C

<i>Tipo de dato</i>	<i>Especificador de formato</i>
<i>Entero</i>	%d, %i, %ld, %li, %o, %x
<i>Flotante</i>	%f, %lf, %e, %g
<i>Carácter</i>	%c, %d, %i, %o, %x
<i>Cadena de caracteres</i>	%s

El especificador de dato se usa para guardar o imprimir el valor de una variable.

Para imprimir un valor entero en base 10 se pueden usar los especificadores %d o %i (%ld ó %li para enteros largos), para imprimir un valor entero en base 8 se utiliza el especificador %o y para imprimir un valor entero en base 16 se utiliza el especificador %x.

Un valor real se puede imprimir con el especificador %f para reales de precisión simple, %lf para reales de doble precisión, %e (notación científica) y %g (redondea la parte fraccionaria a 3 dígitos significativos).

Una variable de tipo carácter se puede imprimir con el especificador %c, con los especificadores %i o %d para imprimir el valor del código ASCII del carácter en base 10, con el especificador %o para imprimir el valor del código ASCII del carácter en base 8 o con el especificador %x para imprimir el valor del código ASCII del carácter en base 16.

El lenguaje C también permite manejar cadenas de caracteres y éstas se pueden imprimir con el especificador %s.









Identificador.

Un identificador es el nombre con el que se va a almacenar en memoria un tipo de dato. Los identificadores siguen las siguientes reglas:

- 🚧 Debe iniciar con una letra [a-z].
- 🚧 Puede contener letras [A-Z, a-z], números [0-9] y el carácter guión bajo (_).

Para imprimir con formato también se utilizan algunas secuencias de caracteres de escape,

C maneja los siguientes:

-  \a carácter de alarma
-  \b retroceso
-  \f avance de hoja
-  \n salto de línea
-  \r regreso de carro
-  \t tabulador horizontal
-  \v tabulador vertical
-  '\0' carácter nulo

Modificaciones de enlace.

Los modificadores que se pueden agregar al inicio de la declaración de variables son const y static.

El modificador const impide que una variable cambie su valor durante la ejecución del programa, es decir, permite para crear constantes. Por convención, las constantes se escriben con mayúsculas y se deben inicializar al momento de declararse.

El modificador static indica que la variable permanece en memoria desde su creación y durante toda la ejecución del programa, es decir, permanece estática en la memoria.

Operadores.

Los operadores aritméticos que maneja lenguaje C se describen en la siguiente tabla:

<i>Operador</i>	<i>Operación</i>	<i>Uso</i>	<i>Resultado</i>
+	Suma	125.78 + 62.5	188.28
-	Resta	65.3 - 32.33	32.97
*	Multiplicación	8.27 * 7	57.75
/	División	15 / 4	3.75
%	Módulo	4 % 2	0

Los operadores lógicos a nivel de bits que maneja el lenguaje C se describen en la siguiente tabla:

<i>Operador</i>	<i>Operación</i>	<i>Uso</i>	<i>Resultado</i>
>>	Corrimiento a la derecha	8 >> 2	2
<<	Corrimiento a la izquierda	8 << 1	16
&	Operador AND	5 & 4	4
	Operador OR	3 2	3
~	Complemento ar-1	~2	1

Moldeo o cast:

El resultado de una operación entre dos tipos de datos iguales puede dar como resultado un tipo de dato diferente, en esos casos es necesario moldear el resultado. A este proceso se le conoce como cast.

Expresiones Lógicas.

Las expresiones lógicas están constituidas por números, caracteres, constantes o variables que están relacionados entre sí por operadores lógicos. Una expresión lógica puede tomar únicamente los valores verdadero o falso.

Los operadores de relación permiten comparar elementos numéricos, alfanuméricos, constantes o variables.

<i>Operador</i>	<i>Operación</i>	<i>Uso</i>	<i>Resultado</i>
==	Igual que	'h' == 'H'	Falso
!=	Diferente a	'a' != 'b'	Verdadero
<	Menor que	7 < 15	Verdadero
>	Mayor que	11 > 22	Falso
<=	Menor o igual	15 <= 22	Verdadero
>=	Mayor o igual	20 >= 35	Falso

Los operadores lógicos permiten formular condiciones complejas a partir de condiciones simples.

<i>Operador</i>	<i>Operación</i>	<i>Uso</i>
!	No	! p
&&	Y	a > 0 && a < 11
 	O	opc == 1 salir != 0

Después de ver la teoría y aprender mas de lo que ya nos había enseñado la profesora del lenguaje de programación C.

Comenzamos a hacer la práctica, aquí tuvimos que modificar unos programas que ya teníamos hechos. La declaración de variables la cambiamos y solo pusimos tipo entero, y en el código donde se realizan las operaciones agregamos el tipo de variable double.

Y eso hicimos con tres programas, el del área del círculo, el del promedio de tres números, y el de el Banco.

AREA DEL CIRCULO.

```
areadouble.cpp
1  #include<stdio.h>
2  #include<conio.h>
3  int r,a;
4  const float pi=3.1416;
5
6  main()
7  {
8      printf("Este programa calcula el area de un circulo.\n");
9      printf("Dame el valor de r: ");
10     scanf("%d",&r);
11     double a=(pi*(double)(r*r));
12     printf("El area del circulo es: %lf",a);
13     getch();
14 }
```

I:\Programacion\areadouble.exe

Este programa calcula el area de un circulo.
Dame el valor de r: 1
El area del circulo es: 3.141600

PROMEDIO DE TRES NÚMEROS.

```
areadouble.cpp Promedio 3 numeros double.cpp
1  #include<stdio.h>
2  #include<conio.h>
3  int a,b,c,p;
4
5  main()
6  {
7      printf("Este programa calcula el promedio de tres numeros\n\n\tDame el primer numero: ");
8      scanf("%d",&a);
9      printf("\tDame el segundo numero: ");
10     scanf("%d",&b);
11     printf("\tDame el tercer numero: ");
12     scanf("%d",&c);
13     double p=(double)((a+b+c)/3);
14     printf("\n\tEl promedio es: %lf",p);
15     getch();
16 }
```

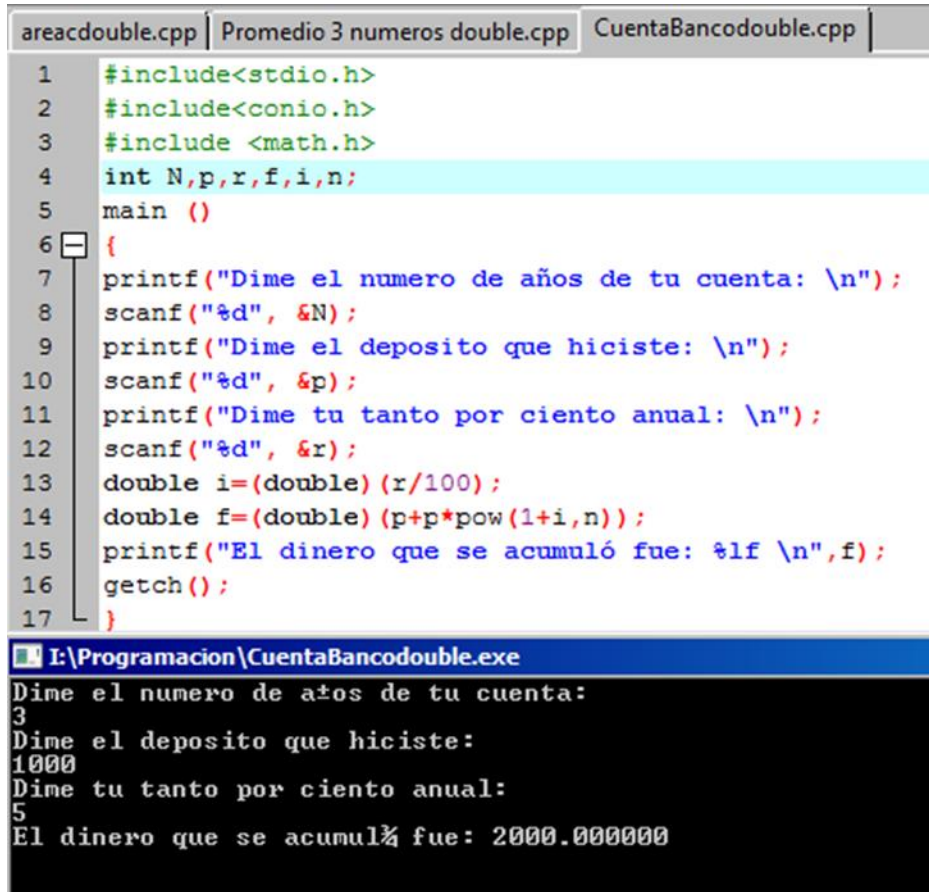
I:\Programacion\Promedio 3 numeros double.exe

Este programa calcula el promedio de tres numeros

Dame el primer numero: 10
Dame el segundo numero: 9
Dame el tercer numero: 8

El promedio es: 9.000000

BANCO.



The image shows a code editor with three tabs: 'areadouble.cpp', 'Promedio 3 numeros double.cpp', and 'CuentaBancodouble.cpp'. The 'CuentaBancodouble.cpp' tab is active, displaying the following C++ code:

```
1  #include<stdio.h>
2  #include<conio.h>
3  #include <math.h>
4  int N,p,r,f,i,n;
5  main ()
6  {
7      printf("Dime el numero de años de tu cuenta: \n");
8      scanf("%d", &N);
9      printf("Dime el deposito que hiciste: \n");
10     scanf("%d", &p);
11     printf("Dime tu tanto por ciento anual: \n");
12     scanf("%d", &r);
13     double i=(double) (r/100);
14     double f=(double) (p+p*pow(1+i,n));
15     printf("El dinero que se acumuló fue: %lf \n",f);
16     getch();
17 }
```

Below the code editor is a terminal window titled 'I:\Programacion\CuentaBancodouble.exe'. It shows the program's execution with the following input and output:

```
Dime el numero de años de tu cuenta:
3
Dime el deposito que hiciste:
1000
Dime tu tanto por ciento anual:
5
El dinero que se acumuló fue: 2000.000000
```

Conclusiones:

La practica me gusto demasiado ya que reforcé aprendizajes que ya tenía del lenguaje de programación C, aprendí a elaborar programas en este lenguaje, utilizando las instrucciones de control de tipo secuencia, para llevar a cabo la declaración de variables de diferentes tipos de datos, y efectué llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.