



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de programación

Grupo: 3

No de Práctica(s):

Integrante(s): Gómez Mendoza Jonan

*No. de Equipo de
cómputo empleado:* 34 Perú

No. de Lista o Brigada: 18

Semestre: 2020-1

Fecha de entrega: 3/10/19

Observaciones: Muy bien

CALIFICACIÓN: 10

Introducción

Para poder programar en c se debe ser mas estricto para usar operadores, declarar, e imprimir para eso se debe de usar una sintaxis correcta.

Estos son los operadores

<i>Operador</i>	<i>Operación</i>
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo

Su resultado es un numero

<i>Operador</i>	<i>Operación</i>
==	Igual que
!=	Diferente a
<	Menor que
>	Mayor que
<=	Menor o igual
>=	Mayor o igual

<i>Operador</i>	<i>Operación</i>
!	No
&&	Y
	O

Y el resultado

de estos es 1 o 0, 1 para verdadero y 0 para falso

TIPO DE DATOS	SE ESCRIBE	MEMORIA REQUERIDA*	RANGO ORIENTATIVO*
Entero	int	2 bytes	- 32768 a 32767
Entero largo	long	4 bytes	- 2147483648 a 2147483647
Decimal simple	float	4 bytes	- $3,4 \cdot 10^{38}$ a $3,4 \cdot 10^{38}$
Decimal doble	double	8 bytes	- $1,79 \cdot 10^{308}$ a $1,79 \cdot 10^{308}$
Carácter	char	1 bytes	0 a 255

Aquí vemos los tipos de variables que hay estos son los mas básicos

<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

Para mostrar y leer se pone esto en el lugar donde se quiera que aparezca la variable luego se especifica que variable toma ese lugar

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.

Actividades

Poner en practica lo aprendido en la introducción

```

1 //mostrar y leer
2 #include <stdio.h>
3 int main(){
4     //Declaracion de palabras a leer
5     int numeroEntrada;
6     double realEntrada;
7     //aquí se declara y asigna valor a otras variables
8     int numeroEntero = 32768;
9     char caracter = 'B';
10    float numeroReal=89.8;
11    //para mostrar en pantalla se usa printf
12    printf("Primero texto solo\n");
13    printf("Luego ponemos poner un entero : %i\n",numeroEntero);
14    printf("Tambien podemos poner un caracter: %c\n",caracter);
15    printf("y un numero real :%.2f\n",numeroReal);
16    //para leer valores se usa scanf
17    scanf("%i",&numeroEntrada);
18    scanf("%lf",&realEntrada);
19    // y ahora podemos mostrarlos aquí también
20    printf("Tu entero: %i\n",numeroEntrada);
21    printf("Tu real: %.3lf\n",realEntrada);
22    return 0;
23 }
```

Aquí tenemos el código de un programa donde imprimimos escaneamos y imprimimos los valores escaneados es importante escribir & antes de la variable a escanear

```
Primero texto solo
Luego ponemos poner un entero : 32768
Tambien podemos poner un caracter: B
y un numero real :89.80
132
12421.5325
Tu entero: 132
Tu real: 12421.532
iMac-J:Practica 7 jonan$
```

Y así aparece en la terminal como se observar en el real que metimos solo se mostraron 3 decimales esto se debe a esta parte del código

`printf("Tu real: %.3lf\n", realEntrada);` en la parte .3 es donde se especifican los decimales

Operadores:

```
1  #include <stdio.h>
2  int main(){
3      //aqui se declaran varias variables en una sola linea
4      int dos, tres, cuatro, cinco;
5      double resultado;
6      //se le asigna valores
7      dos=2;
8      tres=3;
9      cuatro=4;
10     cinco=5;
11     resultado=cinco/dos;
12     printf("5/2=%.1lf\n", resultado);
13     resultado=(double)cinco/dos;
14     printf("5/2=%.1lf\n", resultado);
15     return 0;
16 }
```

Cuando le asignamos un valor a **resultado** se están dividiendo enteros y la división de enteros dara otro entero por eso en el segundo valor de **resultado** se pone (**double**) para que de un numero real

```
iMac-J:Practica 7 jonan$ ./main
5/2=2.0
5/2=2.5
```

Aquí se aprecia los valores que ve la computadora

Operadores lógicos

```
1  #include <stdio.h>
2  int main(){
3      int num1, num2, res;
4      char c1, c2;
5      num1=7;
6      num2=15;
7      c1='h';
8      c2='H';
9      printf("1 es para verdadero y 0 es para falso\n");
10     printf("¿num1 es menor a num2? ->\t%d\n",num1<num2);
11     printf("¿c1 es igual a c2? -> \t\t%d\n",c1==c2);
12     printf("¿c1 es diferente a c2? ->\t%d\n",c1!=c2);
13     res=num1<num2 && c1=='h';
14     printf("¿num1<num2 y c1 es igual a 'h'? -> \t\t%d\n",res);
15     res= c1=='s' || c2=='H';
16     printf("¿c1 es igual a 's' o c2 es igual a 'H' -> \t%d\n",res);
17     return 0;
18 }
```

Adentro de los print se esta usando \t para que sea como un tab

```
1 es para verdadero y 0 es para falso
¿num1 es menor a num2? ->          1
¿c1 es igual a c2? ->              0
¿c1 es diferente a c2? ->          1
¿num1<num2 y c1 es igual a 'h'? ->          1
¿c1 es igual a 's' o c2 es igual a 'H' ->    1
```

Como los operadores usados son lógicos solo pueden tener dos valores 1 y 0, 1 para verdadero y 0 para falso

Conclusión

Para volver a aprender las acciones mas simples del pseudocódigo en lenguaje c se debe de ser mucho mas cuidadoso pero todo tiene su razón de ser con esto escalamos a hacer programas mas explícitos para la computadora que se alejan del lenguaje humano pero que son mas eficientes.