



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Esteban Alejandro Pimentel Alarcon

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): #7

Integrante(s): Franco Inglés Carolina

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 317332836

Semestre: 1°

Fecha de entrega: 3 de octubre de 2019

Observaciones: Muy bien

CALIFICACIÓN: 10

PRÁCTICA #7

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.

LOS TIPOS DE VARIABLES:

Existen varios tipos de variables, y cada uno corresponde a un tamaño máximo de un número, un carácter o incluso una verdad. Cuanto mayor sea el número que pueda admitir, más espacio en memoria ocupará.

BOOL

Por lo general utiliza **1** byte de memoria, valores: **true** o **false**.

CHAR

Utiliza generalmente **1** byte de memoria, permite almacenar un carácter, valores; **256** caracteres.

UNSIGNED SHORT INT

Utiliza generalmente **2** bytes de memoria, valores: de **0** a **65 535**

SHORT INT

Utiliza generalmente **2** bytes de memoria, valores: de **-32768** a **32767**.

UNSIGNED LONG INT

Utiliza generalmente **4** bytes de memoria, valores: de **0** a **4 294 967 295**.

LONG INT

Utiliza generalmente **4** bytes de memoria, valores: de **-2 147 483 648** a **2 147 483 647**.

INT (16 bits)

Utiliza generalmente **2** bytes de memoria, valores: de **-32 768** a **32 767**.

INT (32 bits)

Utiliza generalmente **4** bytes de memoria, valores: de **-2 147 483 648** a **2 147 483 647**.

UNSIGNED INT (16 bits)

Utiliza generalmente **2** bytes de memoria, valores: de **0** a **65 535**.

UNSIGNED INT (32 bits)

Utiliza generalmente **2** bytes de memoria, valores: de **0** a **4 294 967 295**.

DOUBLE

Utiliza generalmente **8** bytes de memoria, valores: de **2.2e-308** a **3.4e-38**.

FLOAT

Utiliza generalmente **4** bytes de memoria, valores: de **1.2e-308** a **3.4e-38**.

NOTA:

El tamaño de las variables en memoria puede variara de un PC a otro.

OPERADORES:

Los operadores de C son un subconjunto de los operadores integrados de C++.

Hay tres tipos de operadores. Una expresión unaria consta de un operador unario antepuesto a un operando, o la palabra clave `sizeof` seguida de una expresión. La expresión puede ser el nombre de una variable o una expresión de conversión. Si la expresión es una expresión de conversión, se debe incluir entre paréntesis. Una expresión binaria consta de dos operandos unidos por un operador binario. Una expresión ternaria consta de tres operandos unidos por el operador de expresión condicional.

C incluye los operadores unarios siguientes:

Símbolo	Nombre
- ~ !	Operadores de negación y complemento
* &	Operadores de direccionamiento indirecto y address-of
sizeof	Operador de tamaño
+	Operador unario más
++ --	Operadores unarios de incremento y decremento

Los operadores binarios se asocian de izquierda a derecha. C proporciona los operadores binarios siguientes:

Símbolo	Nombre
* / %	Operadores de multiplicación
+ -	Operadores aditivos
<< >>	Operadores de desplazamiento
< > <= >= == !=	Operadores relacionales
& ^	Operadores bit a bit
&&	Operadores lógicos
,	Operador de evaluación secuencial

TABLAS DE AYUDA:

<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

<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

<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

<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

Operador Operación

!	No
&&	Y
	O

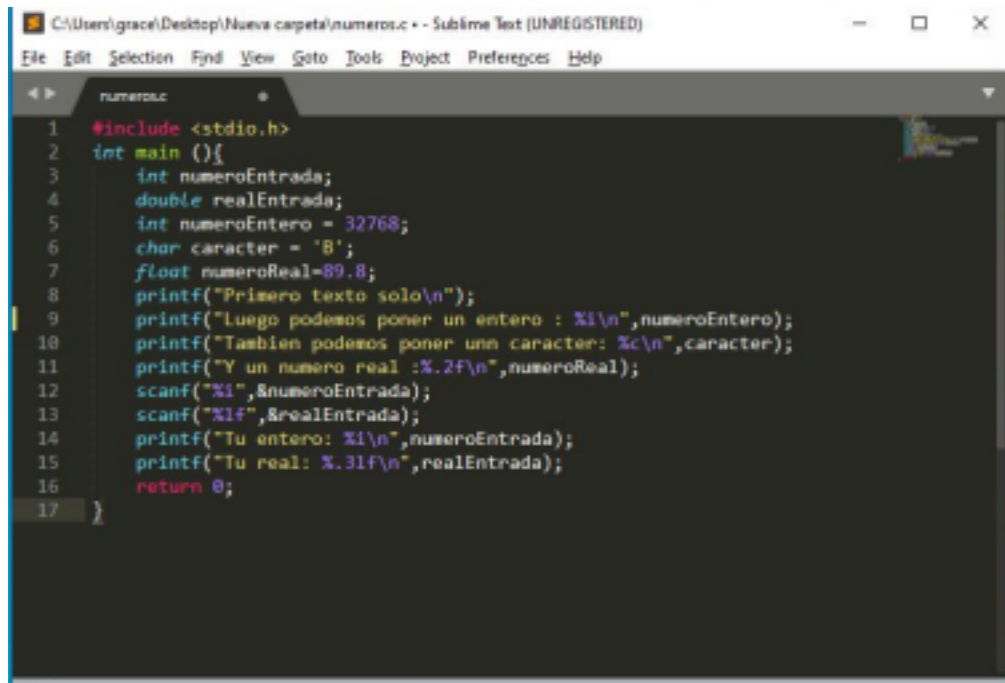
DECLARACIÓN Y ASIGNACIÓN DE VARIABLES

DECLARACIÓN

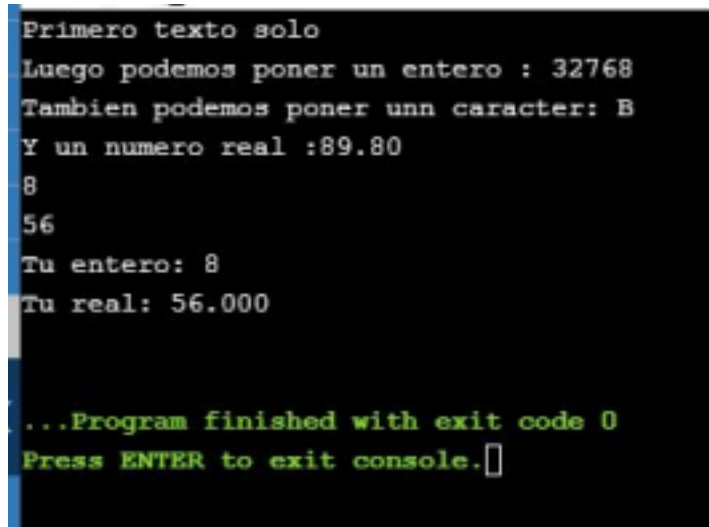
Para declarar una variable, basta con indicar su tipo y su nombre. Existen ciertas convenciones en cuanto al nombre de las variables. Algunos prefieren separar las partes de un nombre con '_', otros prefieren escribir una mayúscula para separarlas.

ACTIVIDADES

PRÁCTICA DE LA INTRODUCCIÓN:



```
1 #include <stdio.h>
2 int main () {
3     int numeroEntrada;
4     double realEntrada;
5     int numeroEntero = 32768;
6     char caracter = 'B';
7     float numeroReal = 89.8;
8     printf("Primero texto solo\n");
9     printf("Luego podemos poner un entero : %i\n", numeroEntero);
10    printf("Tambien podemos poner un caracter: %c\n", caracter);
11    printf("Y un numero real : %.2f\n", numeroReal);
12    scanf("%i", &numeroEntrada);
13    scanf("%lf", &realEntrada);
14    printf("Tu entero: %i\n", numeroEntrada);
15    printf("Tu real: %.3lf\n", realEntrada);
16    return 0;
17 }
```



```
Primero texto solo
Luego podemos poner un entero : 32768
Tambien podemos poner un caracter: B
Y un numero real : 89.80
8
56
Tu entero: 8
Tu real: 56.000

...Program finished with exit code 0
Press ENTER to exit console.
```


OPERADORES:

```
1 #include <stdio.h>
2 int main(){
3     int dos, tres, cuatro, cinco;
4     double resultado;
5     dos=2;
6     tres=3;
7     cuatro=4;
8     cinco=5;
9     resultado=cinco/dos;
10    printf("5/2=%.11f\n",resultado);
11    resultado=(double)cinco/dos;
12    printf("5/2=%.11f\n",resultado);
13    return 0;
14 }
```

```
5/2=2.0
5/2=2.5

...Program finished with exit code 0
Press ENTER to exit console.
```

OPERADORES LÓGICOS:

```
menorigual.c x
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 }
```

```
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

...Program finished with exit code 0
Press ENTER to exit console.
```

CONCLUSIÓN:

Al tener los conceptos definidos de lo que son las variables y operadores asimismo como de su función podemos comenzar a programar en C pero debemos de aprender las funciones de cada uno de ellos. Necesitamos también aprender muy bien a utilizar las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos y a llamar a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.