



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de programación

Grupo:

No de Práctica(s): 7

Integrante(s): Crail Ávila Regina 8973

*No. de Equipo de
cómputo empleado:* 29

No. de Lista o Brigada: 9

Semestre: 2020-1

Fecha de entrega: 03/10/19

Observaciones:

CALIFICACIÓN: _____

FUNDAMENTOS DEL 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.

¿QUÉ ES EL LENGUAJE C?

Se trata de un **lenguaje** de tipos de **datos estáticos**, débilmente tipificado, de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los **lenguajes de alto nivel** pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los **compiladores** suelen ofrecer extensiones al lenguaje que posibilitan **mezclar código** en ensamblador **con código C** o **acceder** directamente a memoria o **dispositivos periféricos**.

TIPOS DE VARIABLES

El lenguaje C **obliga a declarar una variable** antes de ser usada. Declarar una variable no significa que se le asigne contenido, sino **simplemente se indica que la variable existe**.

Para los reales, se tienen también diferentes tipos de variables que asignan más bits para tener mayor rango y mayor precisión. Las variables reales siempre poseen signo.

TIPO DE DATOS	SE ESCRIBE	MEMORIA REQUERIDA*	RANGO ORIENTATIVO*	EQUIVALENCIA EN PSEUDOCÓDIGO	OBSERVACIONES
Entero	int	2 bytes	- 32768 a 32767	Entero	Uso en contadores, control de bucles etc.
Entero largo	long	4 bytes	- 2147483648 a 2147483647	Entero	Igual que int pero admite un rango más amplio
Decimal simple	float	4 bytes	- $3,4 \cdot 10^{38}$ a $3,4 \cdot 10^{38}$	Real	Hasta 6 decimales. También admite enteros
Decimal doble	double	8 bytes	- $1,79 \cdot 10^{308}$ a $1,79 \cdot 10^{308}$	Real	Hasta 14 decimales. También admite enteros
Carácter	char	1 bytes	0 a 255	Alfanumérica	Carácter, independiente o parte de una cadena

TIPOS DE VARIABLES

```
C:\Users\Usuario\Documents\int main.cpp - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
[Iconos de herramientas] TDM-GCC 4.9.2 64-bit Release
(global)
Proyecto Clases(Fun) int main.cpp
1 int main() {
2     //Variables enteras
3     short numeroEntero1;
4     signed int numeroEntero2;
5     unsigned long numeroEntero3;
6
7     //Caracter
8     char caracter;
9
10    //Variables reales
11    float puntoFlotante1;
12    double puntoFlotante2;
13
14    return 0;
15 }
```

```
C:\Users\Usuario\Documents\int main.exe

-----
Process exited after 0.05857 seconds with return value 0
Presione una tecla para continuar . . .
```

MOSTRAR Y LEER

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

```
C:\Users\Usuario\Documents\include.cpp - [Executing] - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
(globals)
Proyecto Clases(Fun) int main.cpp include.cpp
1 #include <stdio.h>
2 int main() {
3
4 //Declaramos variables a Leer
5 int numeroEntrada;
6 double realEntrada;
7
8 //Asignamos variables
9 int numeroEntero = 32768;
10 char caracter = 'B';
11 float numeroReal = 89.8;
12
13 //Mostramos texto y valores
14 printf("Primero texto solo\n");
15 printf("Luego podemos poner un entero: %i\n", numeroEntero);
16 printf("También podemos poner un caracter: %c\n", caracter);
17 printf("Y un número real: %.2f\n", numeroReal);
18
19 //Leemos valores
20 scanf("%i", &numeroEntrada);
21 scanf("%lf", &realEntrada);
22
23 //Y ahora podemos mostrarlos también
24 printf("Tu entero: %i\n", numeroEntrada);
25 printf("Tu real: %.3lf\n", realEntrada);
26
27 return 0;
28 }
29
```

```
C:\Users\Usuario\Documents\include.exe
Primero texto solo
Luego podemos poner un entero:
Tambi n podemos poner un caracter: B
Y un n mero real: 89.80
```

OPERADORES

- Operadores de asignaci n

La operaci n m s importante en cualquier lenguaje de programaci n es la asignaci n:

En C hay muchos operadores de asignaci n que ser n en realidad simplificaciones de expresiones m s complejas, como las auto-operaciones (`+=`, `-=`, `*=`, `/=`, `&=`, ...) y los pre/post incrementos/decrementos (`++`, `--`).

- Operadores aritm ticos

Cabe destacar que se diferencian la divisi n entera y la divisi n real, seg n los operandos.

DdF	C	Descripci�n
+	+	Suma, adici�n
-	-	Resta, sustracci�n
.	*	Multiplicaci�n, producto
<u>Div</u>	/	Cociente divisi�n entera
<u>Mod</u>	%	Resto divisi�n entera
/	/	Divisi�n

OPERADORES

The image shows a screenshot of the Dev-C++ IDE. The top window displays a C++ source file named `include2.cpp`. The code defines a `main` function that declares integer variables `dos`, `tres`, `cuatro`, and `cinco`, and a `double` variable `resultado`. It assigns values to these variables and performs two calculations of `cinco / dos`, printing the results with different formatting. The bottom window shows the execution output of the program, displaying the results of the division and a message indicating the process has exited.

```
1 #include <stdio.h>
2 int main()
3 {
4     int dos, tres, cuatro, cinco;
5     double resultado;
6
7     dos = 2;
8     tres = 3;
9     cuatro = 4;
10    cinco = 5;
11
12    resultado = cinco/dos;
13    printf("5 / 2 = %.1f\n", resultado);
14
15    resultado = (double)cinco/dos;
16    printf("5 / 2 = %.1f\n", resultado);
17
18    return 0;
19 }
```

Execution Output:

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

-----
Process exited after 0.05104 seconds with return value 0
Presione una tecla para continuar . . .
```

COMPARACIONES

DdF	C	Descripción
>	>	Mayor
≥	>=	Mayor o igual
<	<	Menor
≤	<=	Menor o igual
=	==	Igual
≠	!=	Diferente

DdF	C	Descripción
^	&&	And, y, conjunción
∨		Or, o, disyunción
¬	!	Not, no, negación

OPERADORES LÓGICOS

```
C:\Users\Usuario\Documents\include3.cpp - [Executing] - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
TDM-GCC 4.9.2 64-bit Release
(globals)
Proyecto Clases(Fun) int main.cpp include.cpp include2.cpp include3.cpp
1 #include <stdio.h>
2
3 int main() {
4     int num1, num2, res;
5     char c1, c2;
6
7     num1 = 7;
8     num2 = 15;
9     c1 = 'h';
10    c2 = 'H';
11
12    printf("¿ num1 es menor a num2 ? -> %d\n", num1 < num2);
13    printf("¿ c1 es igual a c2 ? -> %d\n", c1 == c2);
14    printf("¿ c1 es diferente a c2 ? -> %d\n", c1 != c2);
15
16    res = num1 < num2 && c1 == 'h';
17    printf("¿ num1 < num2 Y c1 es igual a 'h' ? -> %d\n", res);
18
19    res = c1 == 's' || c2 == 'H';
20    printf("¿ c1 es igual a 's' O c2 a 'H' ? -> %d\n", res);
21
22    return 0;
23 }
```

C:\Users\Usuario\Documents\include3.exe

```
¿ 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 a 'H' ? ->      1

-----
Process exited after 0.0373 seconds with return value 0
Presione una tecla para continuar . . .
```

CONCLUSIÓN

En resumen ésta práctica número 7 me ha servido para ampliar mi conocimiento sobre qué es Lenguaje C, cuales son los tipos de variables así como sus operadores; con ésta actividad he aprendido a desarrollar mis conocimientos básicos en C así como compilar, etc... Por lo que creo que esto es muy necesario en nuestra vida diaria como ingenieros y no sólo quedarnos con lo que vimos en clase sino que investigar por nuestra cuenta y realizar ejercicios como estos en nuestra casa.