



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Alejandro Pimentel

Profesor:

Fundamentos de programación

Asignatura:

3

Grupo:

7

No de Práctica(s):

Colonia Montero Sonia

Integrante(s):

*No. de Equipo de
cómputo empleado:*

8

No. de Lista o Brigada:

2020-1

Semestre:

03/10/2019

Fecha de entrega:

Observaciones: Muy bien
Pero tarde entrega

CALIFICACIÓN: 8

Introducción

Para programar en lenguaje C es necesario conocer la manera correcta de declarar los diferentes tipos de variables que existen de acuerdo a lo que se necesite. Dentro de las variables reales, se tienen diferentes tipos de variables que asignan más bits para tener mayor rango y mayor precisión en caso de ser necesario. Así mismo, se pueden usar especificadores de formato para dar a conocer al programa el tipo de dato con el que se está trabajando. Por otro lado, para poder programar también se deben conocer los comandos para realizar operaciones básicas como suma, resta, multiplicación, división o módulo. Además, las comparaciones y los operadores lógicos también son muy importantes para poder programar. Todas estas funciones se exploran a lo largo del presente documento, así como la manera correcta de ejecutarlas en 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

En primer lugar, se escribieron los diferentes tipos de variables que se pueden utilizar.

```
1 int main() {
2
3     // Variables enteras
4     short numeroEntero1;
5     signed int numeroEntero2;
6     unsigned long numeroEntero3;
7
8     // Caracter
9     char caracter;
10
11    // Variables reales
12    float puntoFlotante1;
13    double puntoFlotante2;
14
15    return 0;
16 }
17
```

Posteriormente, se mostraron y leyeron diferentes tipos de variables con un especificador de formato.

```
#include <stdio.h>

int main() {

    // Declaramos variables a leer
    int numeroEntrada;
    double realEntrada;

    // Asignamos variables
    int numeroEntero = 32768;
    char caracter = 'B';
    float numeroReal = 89.8

    // Mostramos texto y valores
    printf("Primero texto solo\n");
    printf("Luego podemos poner un entero: %i\n", numeroEntero);
    printf("También podemos poner un caracter: %c\n", caracter);
    printf("Y un numero real: %.2f\n", numeroReal);

    // Leemos valores
    scanf("%i", &numeroEntrada);
    scanf("%lf", &realEntrada);

    // Y ahora podemos mostrarlos también
    printf("Tu entero: %i\n", numeroEntrada);
    printf("Tu real: %.3lf\n", realEntrada);

    return 0;
}
```

Después se usaron operadores (en este caso de división entre dos números).

```
#include <stdio.h>

int main() {

    int dos, tres, cuatro, cinco;
    double resultado;

    dos = 2;
    tres = 3;
    cuatro = 4;
    cinco = 5;

    resultado = cinco/dos;
    printf("5 / 2 = %.11f\n");

    resultado = (double)cinco/dos;
    printf("5 / 2 = %.11f\n", resultado);

    return 0;
}
```

Por último, se usaron algunas comparaciones y operadores lógicos.

```
#include <stdio.h>

int main() {

    int num1, num2, res;
    char c1, c2;

    num1 = 7;
    num2 = 15;
    c1 = 'h';
    c2 = 'H';

    printf("& num1 es menor a num2 ? -> %td\n", num1<num2);
    printf("& c1 es igual a c2 ? -> %td\n", c1==c2);
    printf("& c1 es diferente a c2 ? -> %td\n", c1!=c2);

    res = num1 < num2 && c1 == 'h';
    printf("& num1 < num2 Y c1 es igual a 'h'? -> %td\n", res);

    return 0;
}
```

Resultados

Todos los programas se compilaron y se leyeron.

Mostrar y leer

```
Primero texto solo
Luego podemos poner un entero: 32768
Tambien podemos poner un caracter: B
Y un numero real: 89.88
12
2
Tu entero: 12
Tu real: 2.000
```

Operadores

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

Operadores lógicos.

```
bash: ./main.c: Permisos denegados
alumno@pch:~/Documentos$ gcc main.c -o main
alumno@pch:~/Documentos$ ./main
¿ 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
alumno@pch:~/Documentos$
```

A continuación se explica lo que hacen las instrucciones dadas para el programa.

Tipos de variables

```
int main() { %función principal
```

```
    // Variables enteras %comentarios (las tres son enteros pero con diferentes características)
    short numeroEntero1; %se declara una variable de tipo short int llamada numeroEntero1
    signed int numeroEntero2; %se declara una variable de tipo signed int llamada numeroEntero2
    unsigned long numeroEntero3; %se declara una variable de tipo unsigned int llamada
    numeroEntero3
```

```
    // Caracter %comentarios
```

```
    char caracter; %se declara una variable de tipo char llamada caracter
```

```
    //Variables reales %comentarios (ambas variables reales pero double tiene más precisión que
    float)
```

```
    float puntoFlotante1; %se declara una variable de tipo float llamada puntoFlotante1
```

```
    double puntoFlotante2; %se declara una variable de tipo double llamada puntoFlotante2
```

```
    return 0; %se usa para cerrar el main
```

```
}
```

Mostrar y leer

```
#include <stdio.h> %se dice en que librería se va a trabajar (dependiendo de la librería las funciones
que se pueden usar cambian).
```

```
int main() { %función principal
```

```
    // Declaramos variables a leer %comentarios
```

```
    int numeroEntrada; %se declara una variable de tipo entera de nombre numeroEntrada
```

```
    double realEntrada; %se declara una variable de tipo double de nombre realEntrada
```

```
    // Asignamos variables %comentarios
```

```
    int numeroEntero = 32768; %se asigna valor a la variable de tipo entera de nombre
numeroEntero
```

```
    char caracter = 'B' %se asigna valor a la variable de tipo carácter de nombre caracter
```

```
    float numeroReal 89.8 %se asigna valor a la variable de tipo float de nombre numeroReal
```

```
    // Mostramos texto y valores %comentarios
```

```
    printf("Primero texto solo\n"); %se muestra en pantalla el texto entre comillas
```

```
    printf("Luego podemos poner un entero: %i\n", numeroEntero); %se muestra en
pantalla el texto entre comillas y el valor de numeroEntero
```

```
    printf("También podemos poner un carácter; %c\n", caracter); %se muestra en
```

pantalla el texto entre comillas y el valor de carácter

```
printf("Y también un número real\n", numeroReal); %se muestra en pantalla el texto entre comillas y el valor de numeroReal
```

```
// Leemos valores %comentarios
scanf("%i", &numeroEntrada) %se lee el valor de numeroEntrada usando %i para aclarar que es una variable de tipo entero
scanf("%lf", &realEntrada) %se lee el valor de realEntrada usando %lf para aclarar que es una variable de tipo flotante

// Y ahora podemos mostrarlos también %comentarios
printf("Tu entero: %i\n", numeroEntrada); %se muestra en pantalla el texto entre comillas y el valor de numeroEntrada
printf("Tu real: %.3lf\n", realEntrada); %se muestra en pantalla el texto entre comillas y el valor de realEntrada

return 0; %se usa para cerrar el main
}
```

Operadores

#include <stdio.h> %se dice en que librería se va a trabajar (dependiendo de la librería las funciones que se pueden usar cambian).

```
int main() { %función principal
```

```
int dos, tres, cuatro, cinco; %se declaran las variables dos, tres, cuatro, cinco de tipo enteras
double resultado; %se declara una variable de tipo double de nombre resultado
```

```
dos = 2; %se asigna valor de 2 a la variable dos
tres = 3; %se asigna valor de 3 a la variable tres
cuatro = 4; %se asigna valor de 4 a la variable cuatro
cinco = 5; %se asigna valor de 5 a la variable cinco
```

```
resultado = cinco/dos; %se realiza la operación de la variable cinco entre la variable dos y se guarda en la variable resultado
```

```
printf("5/2 = %.1lf\n", resultado); %se muestra en pantalla el texto entre comillas y el valor de resultado
```

%el resultado obtenido va a ser 2 ya que, aunque la variable resultado sea un double, las variables cinco y dos son enteros, por lo que su resultado nunca será un doble; a menos que se haga lo que se muestra a continuación.

```
resultado = (double)cinco/dos; %se realiza la operación de la variable cinco entre la variable dos declarando a esta operación como un double para que el resultado sea exacto (2.5)
```

```
printf("5/2 = %.1lf\n", resultado); %se muestra en pantalla el texto entre comillas y el valor de resultado
```

```
return 0; %se usa para cerrar el main
```

```
}
```

Operadores lógicos

#include <stdio.h> %se dice en que librería se va a trabajar (dependiendo de la librería las funciones que se pueden usar cambian).

```
int main() { %función principal
```

```

int num1, num2, res; %se declaran las variables num1, num2, res de tipo enteras
char c1, c2; %se declaran las variables c1, c2 de tipo carácter

num1 = 7 %se asigna valor de 7 a la variable num1
num2 = 15; %se asigna valor de 15 a la variable num2
c1 = 'h'; %se asigna valor de h a la variable c1
c2 = 'H'; %se asigna valor de H a la variable c1

printf("\n num1 es menor a num2 ? -> \t%d\n", num1<num2); %se muestra en pantalla el
texto entre comillas (se usa %d para señalar que es un carácter) y si num1 es menor a num2
printf("\n c1 es igual a c2 ? -> \t%d\n", c1==c2); %se muestra en pantalla el texto entre
comillas (se usa %d para señalar que es un carácter) y si c1 es igual a c2
printf("\n c1 es diferente a c2 ? -> \t%d\n", c1!=c2); %se muestra en pantalla el texto
entre comillas (se usa %d para señalar que es un carácter) y si c1 es diferente a c2

res = num1 < num2 && c1 == 'h'; %a res se le asigna valor si num1 es menor a num2 y c1 es
igual a h
printf("\n num1 < num2 Y c1 es igual a 'h' ? -> \t%d\n", res); %se muestra en pantalla
el texto entre comillas (se usa %d para señalar que es un carácter) y el valor de res

res = c1 == 's' || c2 == 'H'; %a res se le asigna valor si c1 es igual a s o si c2 es igual a H
printf("\n c1 es igual a 's' O c2 a 'H' ? -> \t%d\n", res); %se muestra en pantalla el
texto entre comillas (se usa %d para señalar que es un carácter) y el valor de res

return 0; %se usa para cerrar el main
}

```

Conclusión

Si bien ya se conoce como se usan los operadores lógicos, comparaciones y tipos de variables en diagrama de flujo y pseudocódigo; la manera en la que se usan este tipo de comandos dentro del lenguaje C cambia. Es por ello, que es de gran importancia realizar pequeños programas, como los que se realizaron a lo largo del presente documento, para poder adaptarse a la manera en la que se trabaja dentro de C.