



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: CLAUDIA RODRIGUEZ ESPINO

Asignatura: Fundamentos de programación

Grupo: 1112

No de Práctica(s): Practica 7

Integrante(s): CARRASCO MENDOZA JENNIFER

Semestre: 2018-1

Fecha de entrega: 06/10/2017

Observaciones:

Esta practica nos ayudo a ver la importancia de agregar comentarios en los codigos de ciertas cosas para facilitar el arreglo de este.

CALIFICACIÓN: _____

Guía práctica de estudio 07

Fundamentos de 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.

Actividades:

- ♣ Crear un programa en lenguaje C que tenga definidas variables de varios tipos, se les asigne valores adecuados (por lectura o asignación directa) y muestre su valor en la salida estándar.
- ♣ En un programa en C, asignar valores a variables utilizando expresiones aritméticas; algunas con uso de cambio de tipo (cast)
- ♣ Elaborar expresiones relacionales/lógicas en un programa en C y mostrar el resultado de su evaluación.

Introducción

- **Declaración de variables}**

```
#include <stdio.h>
/*
```

Este programa muestra la manera en la que se declaran y asignan variables de diferentes tipos (numéricas enteras, numéricas reales, caracteres), así como la manera en la que se imprimen los diferentes tipos de datos.

```
*/
```

```
int main() {
```

/ Es recomendable que, al inicio, se declaren todas las variables que se van a utilizar en el programa */*

```
// variables enteras
```

```
int enteroNumero;
```

```
char caracterA = 65; // Convierte el entero a carácter (ASCII)
```

```
// Variable reales
```

```
double puntoFlotanteNumero;
```

```

// Asignar un valor del teclado a una variable
printf("Escriba un valor entero: ");
scanf("%d", &enteroNumero);
printf("Escriba un valor real: ");
scanf("%lf", &puntoFlotanteNumero);

// Imprimir los valores con formato
printf("\nImprimiendo las variables enteras:\n");
printf("\tValor de enteroNumero = %i\n", enteroNumero);
printf("\tValor de caracterA = %c\n", caracterA);
printf("\tValor de puntoFlotanteNumero = %lf\n", puntoFlotanteNumero);

printf("\nValor de enteroNumero en base 16 = %x\n", enteroNumero);
printf("\tValor de caracterA en código hexadecimal = %i\n", enteroNumero);
printf("\tValor de puntoFlotanteNumero en notación científica = %e\n",
puntoFlotanteNumero);

// La función getch() espera un carácter para continuar la ejecución
getch();
return 0;
}

```

- **Variables estáticas y dinámicas**

```

#include <stdio.h>
/*
Este programa muestra la manera en la que se declaran y asignan las variables
estáticas y las constantes.
*/
int main() {
    const int constante = 25;
    static char a = 'a';

    printf("Valor constante: %i\n", constante);
    printf("Valor estático: %c\n", a);

    // El valor de la variable declarada como constante no puede cambiar.
    // La siguiente línea genera un error al compilar si se quita el comentario:
    // constante = 30;
    // las variables estáticas sí pueden cambiar de valor
    a = 'b';
    printf("\nValor estático: %c\n", a);
    return 0;
}

```

- **Moldeo o cast.**

```
#include <stdio.h>

// Si se tienen 2 enteros
int cinco = 5, dos = 2, resultado;

// La operación de división entre dos enteros
// genera un valor real; en este caso, hay que
// moldear (cast) el resultado del lado derecho del
// igual para que corresponda con el lado izquierdo
// y se pueda asignar.

int main(){
    double res = (double)cinco/dos;
    // Si no se hiciese el cast, el resultado se truncaría.
    printf("%lf",res);

    resultado = cinco/dos;
    printf("\n%i",resultado);
    return 0;
}
```

- **Operadores.**

```
#include <stdio.h>
/*
Este programa muestra la manera en la que se realiza un moldeo o cast y cómo
manipular números a nivel de bits: Corrimiento de bits a la izquierda y a la derecha,
Operador AND a nivel de bits, Operador OR a nivel de bits
*/
int main(){
    short ocho, cinco, cuatro, tres, dos, uno;

    // 8 en binario: 0000 0000 0000 1000
    ocho = 8;
    // 5 en binario: 0000 0000 0000 0101
    cinco = 5;
    // 4 en binario: 0000 0000 0000 0100
    cuatro = 4;
    // 3 en binario: 0000 0000 0000 0011
    tres = 3;
```

```
// 2 en binario: 0000 0000 0000 0010
dos = 2;
// 1 en binario: 0000 0000 0000 0001
uno = 1;

printf("Operadores aritméticos\n");
double res = (double)cinco/dos; // Cast
printf("5 / 2 = %lf\n",res);
printf("5 modulo 2 = %d\n",cinco%dos);

printf("Operadores lógicos\n");
printf("8 >> 2 = %d\n",ocho>>dos);
printf("8 << 1 = %d\n",ocho<<1);
printf("5 & 4 = %d\n",cinco&cuatro);
printf("3 | 2 = %d\n",tres|dos);

printf("\n");
return 0;
}
```

- **Operador lógico OR.**

```
#include <stdio.h>

/*Uso del operador lógico OR a nivel bits*/

int main()
{
    int a=23;
    int b=90;
    int resultado;

    printf("a=%d b=%d ",a,b);

    resultado=a|b;

    printf(" Resultado=%d",resultado);

    return 0;
}
```

- **Operador lógico AND.**

```
#include <stdio.h>
```

*/*Uso del operador lógico AND a nivel bits*/*

```
int main()
{
    int a=23;
    int b=90;
    int resultado;

    printf("a=%d b=%d ",a,b);

    resultado=a&b;

    printf(" Resultado=%d",resultado);

    return 0;
}
```

Actividades

- ***Suma de dos números***

#include <stdio.h>

int a,b,c;

main()

```
{
    const int u = 163;
    printf("Introduzca un valor: ");
    scanf("%d",&a);
    printf("Introduzca el otro valor: ");
    scanf("%d",&b);

    c=(a+b);

    printf("La suma de los 2 n%cmeros es: %d",u,c);
}
```

```
}
```

- **Área de un círculo**

```
#include <stdio.h>
```

```
#include <math.h>
```

```
float r, e, A;
```

```
main ()
```

```
{
```

```
    const float PI = 3.1416;
```

```
    const int a = 160;
```

```
    const int i = 161;
```

```
    printf("Dame el valor del radio: ");
```

```
    scanf("%f",&r);
```

```
    e=(pow(r,2));
```

```
    A=(PI*e);
```

```
    printf("El %crea del c%crculo es: %.4f",a,i,A);
```

```
}
```

- **Promedio de 3 números**

```
#include <stdio.h>
```

```
int contador=0;
```

```
float numero, suma=0;
```

```
main()
```

```
{
```

```
    const int u = 163;
```

```
    while(contador<3)
```

```
    {
```

```
        contador++;
```

```
        printf("Dame el valor %d: ", contador);
```

```
        scanf("%f",&numero);
```

```
        suma=(suma+numero);
```

```
    }
```

```
    double promedio= (double)suma/contador;
```

```
    printf("El promedio de los %d n%cmernos es: %.4f",contador,u,promedio);
```

```
}
```

Conclusión:

Al agregar los comentarios al código nos ayuda a recordar procesos o poner notas que nos ayudaran más adelante para poder verlos y recordar que arreglos se habían hecho antes y que hacer en dado caso que algo pase.