|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Cruz Carlón Juan Alfredo |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | Bloque 131 |
| *No de Práctica(s):* | Práctica 7 |
| *Integrante(s):* | Zavala Pérez Vianey Michelle |
|  |  |
|  |  |
| *Semestre:* | 2018-1 |
| *Fecha de entrega:* | Octubre 19, 2017 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Fundamentos de Programación Ejercicios de depuración.

Llena la tabla para cada uno de los ejercicios.

|  |  |  |  |
| --- | --- | --- | --- |
| **Programa** | **Lo que debería realizar** | **Lo que realiza** | **Reparación** |
| #include <stdio .h>  int main() {  int respuesta ; respuesta = 2 + 2; printf(”La respuesta es: %d\n”); return 0 ;  } | Imprimir La respuesta es: 4 | No realiza nada porque hay errores de sintaxis.  Hay un espacio que no debe de ir en el primer renglón en “<stdio.h>”  Hay varios espacios que no deben de estar en la declaración de variables, en la asignación de valores.  Las comillas están mal puestas en la impresión.  Se utilizó de forma incorrecta el %d, ya que le hacia falta decir que valor había que imprimir | Se quitaron todos los espacios que sobraban.  Se pusieron las comillas de forma correcta en la impresión.  Se agregó la variable “respuesta” después de las comillas que cerraban en la impresión del valor y así ya imprimió: La respuesta es 4. |
| #include <stdio .h>  int main() {  float respuesta ; respuesta = 7.0/22.0; printf (”La respuesta es : %d\n” , respuesta ); return 0 ;  } | Imprimir La respuesta es: 0.3181818 | No realiza nada porque hay errores de sintaxis.  Hay un espacio que no debe de ir en el primer renglón en “<stdio.h>”  Hay varios espacios que no deben de estar en la declaración de variables, en la asignación de valores.  Las comillas están mal puestas en la impresión.  Esta mal utilizado el %d porque no corresponde cuando se usa float. | Se quitaron todos los espacios que sobraban.  Se pusieron las comillas de forma correcta en la impresión.  Cambiamos %d por %f, que es el comando que corresponde a float ya que es un valor que no va a cambiar. |
| #include <stdio .h>  int main() {  char linea [80]; int balance ; printf(”Ingresa el balance (en pesos) de la  cuenta: ”); fgets(linea , sizeof(linea), stdin); sscanf(linea ,”%d”, &balance);  if(balance = 0){ printf(”No tienes dinero \n”);  }else{ printf (”Tienes : %d pesos .\n”, balance )  }  return 0 ; } | Imprimir el balance ingresado | No realiza nada porque hay errores de sintaxis.  Hay un espacio que no debe de ir en el primer renglón en “<stdio.h>”  Hay varios espacios que no deben de estar en la declaración de variables, en la asignación de valores.  Las comillas están mal puestas en la impresión.  Cuando se iguala el balance a cero, está mal colocado el símbolo de igualdad. | Se quitaron todos los espacios que sobraban.  Se pusieron las comillas de forma correcta en la impresión.  Se corrigió el símbolo de igualdad por dos == en vez de solo uno. |

**Determina que hace el siguiente programa. Explica en qué contribuye cada línea y haz una representación gráfica del mismo.**

#include <stdio.h> //Abrimos la biblioteca

int main() //Empezamos el proceso principal

{ //Abrimos una llave para indicar desde donde empezará la acción principal

int a; //Declaramos nuestra primera variable “a” del tipo entero.

int b; //Declaramos una segunda variable “b” del tipo entero.

char c; //Declaramos una tercera variable “c”

char d[100]; //Declaramos una cuarta variable.

a=0; //Le asignamos a la variable a el valor de 0.

while(1) //Iniciamos un ciclo

{ //Abrimos una llave para indicar que pasará en el ciclo

printf("A: %d\n", a); // Imprime A y su valor que será el valor de la variable “a”.

printf ("Operador ?: "); //Imprime la palabra Operador para pedir uno al usuario.

fgets(d, sizeof(d), stdin); //Imprime la cadena

sscanf(d, "%c", &c); //Lee los datos de d , los muestra y los guarda en c.

printf ("Numero?: "); // Pide un número al usuario.

fgets(d, sizeof(d), stdin); //Imprime la cadena

sscanf(d,"%d",&b); // Lee los datos de d , los muestra y los guarda en b.

if(c=='q') //Abrimos una condición.

break ; //Si cumple la condición se sale de ciclo.

if(c=='Q') // Abrimos una condición.

break ;// Si cumple la condición se sale de ciclo.

switch (c) // Empezamos una función de casos a partir de lo que sea c

{ //Abrimos una llave para indicar en donde empiezan los casos

case '+': //Pusimos en primer caso, si c es igual al signo +

a=a+b; //Establecimos que si c es igual al signo + entonces a “a” se le asignará el valor de a+b.

break ; // Con esto una vez de asignarle valor a “a” se saldrá de los casos.

case '-': //Iniciamos el caso de la resta.

a=a-b; //Asignamos a ‘a’ una resta.

break ; //Indicamos un break para que se termine un proceso.

case '\*': //Iniciamos el caso de la multipliación.

a=a\*b; //Asignamos una operación a la variable a.

break ; //Indicamos un break para que se termine un proceso.

case '/': //Iniciamos el caso de la división.

if (b==0) //Damos una condición para el if.

{ //Iniciamos el if.

printf("Error \n"); //Indicamos que imprima lo que está en los parétesis.

} //Cerramos la llave del if para indicar que terminó.

Else //Indicamos que se realizarán las siguientes acciones.

{ //Abrimos la llave del else.

a=a/b; //Damos un valor a la variable a.

} //Cerramos la llave del else.

break ; //Damos la orden de cerrar el proceso.

default : //Es el último caso del switch.

printf ("Op. desconcida %c\n",c); //Imprimimos lo que hay en paréntesis.

break ; //Damos la orden de cierrar el proceso.

} //Cerramos la llave del switch.

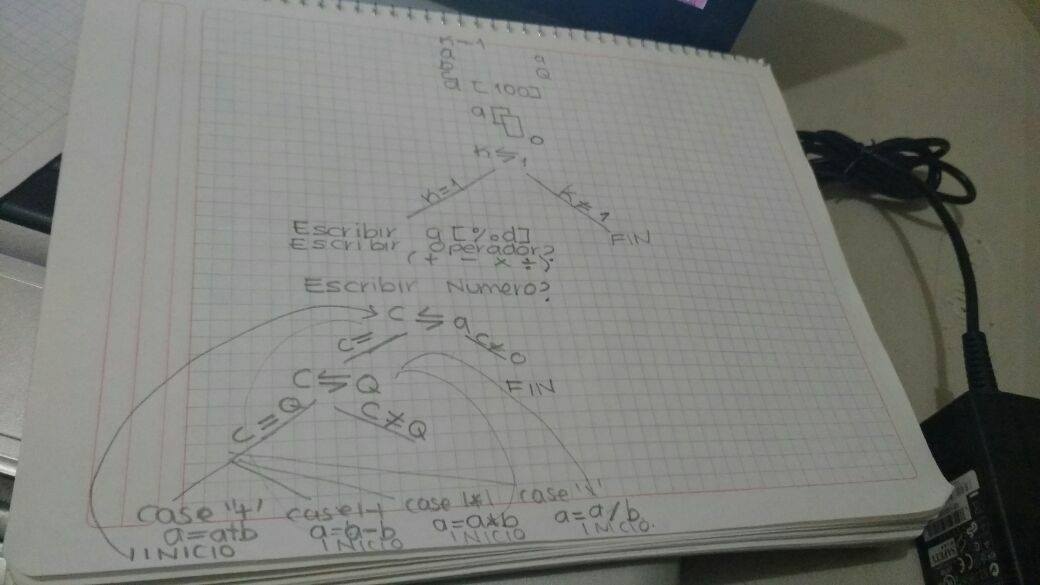
} //Cerramos la llave del while.

return 0; //Damos la indicación de return para que el programa continúe

} //Cerramos la llave de la función principal main.

El programa de aquí arriba te permite realizar las operaciones básicas (suma, resta, multiplicación y división) utilizando dos números, inicia con el cero, tú le pones el símbolo de la operación que quieres que haga y luego el otro número con cual hacerla y la realiza. Esto lo hace una infinidad de veces con el residuo de la última operación que realizó y un nuevo número que el usuario ponga.

**Diagrama**

****