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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | M.C. Juan Alfredo Cruz Carlon. |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | 1107 |
| *No. De Práctica(s):* | 10– “Depuración de programas”. |
| *Integrante(s):* | Díaz Martínez Karla Lisset  Elvira López Biaani María  Hernández García María Esther  Vázquez León Pamela Estefanía |
| *Semestre:* | 2018-I |
| *Fecha de entrega:* | 13-Noviembre-2017 |
| *Observaciones:* |  |

**CALIFICACIÓN:\_\_\_\_\_\_\_\_\_\_\_\_\_**

Ejercicios de depuraci´on.

# Llena la tabla para cada uno de los ejercicios.

|  |  |  |  |
| --- | --- | --- | --- |
| Programa | Lo que deber´ıa realizar | Lo que realiza | Reparaci´on |
| **#include** *<* s t d i o . h*>*  **i n t** main ( )  {  **i n t** r e s p u e s t a ; r e s p u e s t a = 2 + 2 ;  p r i n t f ( ” La r e s p u e s t a e s : %d\n” ) ;  **return** 0 ;  } | Imprimir **La respuesta es: 4** | La respuesta es: 1052571112 | En la línea 7 después de cerrar las comillas del printf, falta una coma seguida de “respuesta” para que lo imprima. |
| **#include** *<* s t d i o . h*>*  **i n t** main ( )  {  **f l o a t** r e s p u e s t a ; r e s p u e s t a = 7 . 0 / 2 2 . 0 ;  p r i n t f ( ” La r e s p u e s t a e s : %d\n” , r e s p u e s t a ) ;  **return** 0 ;  } | Imprimir **La respuesta es: 0.3181818** | La respuesta es: 137964376 | Se cambia el %d por %f, ya que se usa este ultimo para numeros reales. |
| **#include** *<* s t d i o . h*>*  **i n t** main ( )  {  **char** l i n e a [ 8 0 ] ;  **i n t** b a l a n c e ;  p r i n t f ( ” I n g r e s a e l b a l a n c e ( en p e s o s ) de l a c u e n t a : ” ) ;  f g e t s ( l i n e a , **s i z e o f** ( l i n e a ) , s t d i n ) ; s s c a n f ( l i n e a , ” %d” , &b a l a n c e ) ;  **i f** ( b a l a n c e = 0 ) {  p r i n t f ( ”No t i e n e s d i n e r o \n” ) ;  } **e l s e** {  p r i n t f ( ” T i e n e s : %d p e s o s . \ n” , b a l a n c e ) ;  }  **return** 0 ;  } | Imprimir el balance ingresado | Pide al usuario: “Ingresa el balance (en pesos) de la cuenta:”, pero sea cual sea el balance ingresado te regresa: “Tienes: 0 pesos” | Falta agregar un doble = en la condición if. |

1

2

3

4

5

6

7

8

9

1

2

3

4

5

6

7

8

9

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

# Determina que *hace* el siguiente programa. Explica en qu´e contribuye cada l´ınea y haz una representaci´on gr´afica del mismo.

# //Este programa sirve para hacer operaciones (suma, resta, multiplicación y división), pidiendo al usuario la operación que desea hacer y luego el número que necesita.

1 **#include** *<* s t d i o . h*>* // Se agrega la biblioteca estándar de entrada y salida, que nos servirá para lectura y escritura de datos.

2

3 **i n t** main ( ) // Es la función principal

{

4 //Se abre la función principal.

5 **i n t** a ; // Se declara la variable a de valor entero

6 **i n t** b ; //Se declara la variable b de valor entero

7 **char** c ; // Se declara la variable c de tipo carácter

8 **char** d [ 1 0 0 ] ; //Se declara la variable d de tipo carácter con un máximo de 100 datos de tipo carácter.

9 a = 0 ; //Se le asigna el valor de 0 a la variable a

10

11 **while** ( 1 ) { //Se inicia una condición: Mientras sea 1

12 p r i n t f ( ”A: %d\n” , a ) ; //Imprime el valor de a

13

14 p r i n t f ( ” Operador ? : ” ) ; // Imprime la frase: Operador ?, para que el usuario proporcione alguna operación.

15 f g e t s ( d , **s i z e o f** ( d ) , s t d i n ) ; // Esta función se usa para que se puedan leer todos los caracteres, y se detendrá cuando en la variable d se copien el máximo de 100 caracteres que se estableció al inicio

16 s s c a n f ( d , ” %c ” , &c ) ; //Se usa para leer los datos de la variable que se obtuvo de la entrada

17

18 p r i n t f ( ”Numero ? : ” ) ; //Imprime la frase: Numero?, para que el usuario proporcione algun numero o valor para realizar las operaciones.

19 f g e t s ( d , **s i z e o f** ( d ) , s t d i n ) ; // Esta función se usa para que se puedan leer todos los caracteres, y se detendrá cuando en la variable d se copien el máximo de 100 caracteres que se estableció al inicio

20 s s c a n f ( d , ” %d” , &b ) ; //Se usa para leer los datos de la variable que se obtuvo de la entrada

21

22 **i f** ( c == ’ q ’ ) //Se abre una condición: Si c es igual a q

23 **break** ; // Si se cumple la condición, se termina el ciclo.

24 **i f** ( c == ’Q ’ ) //Se abre una condición: Si c es igual a Q

25 **break** ; //Si se cumple la condición se termina el ciclo.

26 **switch** ( c ) // Se va a evaluar la variable c, comparándola con los siguientes casos.

{

27 **case** ’+ ’ : // Se compara la variable c, que contiene la operación ingresada por el usuario, con una suma

28 a = a + b ; // Si la variable c es una suma, a la variable a se le asignará el resultado de la suma de a y b.

29 **break** ; //Se termina el ciclo

30 **case** ’ ’ : // Se compara la variable c, que contiene la operación ingresada por el usuario, con una resta

−

31 a = a b ; // Si la variable c es una resta, a la variable a se le asignará el resultado de la resta de a y b.

−

32 **break** ; //Se termina el ciclo.

33 **case** ’ ’ : // Se compara la variable c, que contiene la operación ingresada por el usuario, con una multiplicación

∗

34 a = a b ; // Si la variable c es una multiplicación a la variable a se le asignará el resultado de la multiplicación de a y b.

∗

35 **break** ; //Se termina el ciclo.

36 **case** ’ / ’ : // Se compara la variable c, que contiene la operación ingresada por el usuario, con una división

37 **i f** ( b == 0 ) //Si la variable c es una división, se abre una condición: si es igual a 0.

{

38 p r i n t f ( ” E r r o r n” ) ; //Si b es igual a 0 se imprime “Error”

\

39 **e l s e** // Si b no es igual a 0, se hace lo siguiente.

} {

40 a = a /b ; // A la variable a se le asigna el resultado de la division de a entre b.

}

41 // Se termina la condición else o de lo contrario

42 **break** ;//Se termina el ciclo.

43 **d e f a u lt** : //Si la operación no es una suma, resta, multiplicación o division, se ejecuta este bloque “por defecto”

44 p r i n t f ( ”Op . d e s c o n o c i d a %c n” , c ) ; //Se imprime “Operación desconocida” junto con la operación ingresada

\

45 **break** ; //Se termina el ciclo.

46 } //Se Cierra el bloque por defecto

47

48 //Se Cierra la condición “mientras”

}

49 **return** 0 ; //Termina el programa

50 } //Se cierra la función principal

