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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | JUAN ALFREDO CARLÓN |
| *Asignatura:* | FUNDAMENTOS DE PROGRMACION |
| *Grupo:* | 1107 |
| *No de Práctica(s):* | 10 |
| *Integrante(s):* | Diaz Martinez Karla Lissette  Elvira Lopez Biaani Maria |
|  | Hernandez Garcia María Esther  Vazquez Leon Pamela Estefania |
|  |  |
| *Semestre:* | 2018-1 |
| *Fecha de entrega:* | 12/ Noviembre/2017 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

Fundamentos de Programaci´on 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** | Realiza la suma, no obstante no imprime ningún resultado | En la séptima línea se le tiene que agregar lo siguiente:  p r i n t f ( ” La r e s p u e s t a e s : %d\n”, respuesta ) ;  Para que así nos imprima el resultado |
| **#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** | Realiza la operación, sin embargo, el resultado que se imprime es un numero entero, u no el que el que debería | En la séptima línea se cambia el %d (El cual se utiliza únicamente para números enteros), por un %f el cual se utiliza para valores 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 | Dería imprimir el valor que el usuario le ingresara y en el caso de que el valor ingresado fuese “cero” se escribirá “No tiene dinero” pero cuando poníamos cero se escribía “Tienes 0 pesos” y no “No tiene dinero” | En la undécima line en el if se le debe de poner doble igual, para que el programa funcione de manera prevista  Ósea:  **i f** ( b a l a n c e == 0 ) { |

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.

1 **#include** *<* s t d i o . h*>*

2

3 **i n t** main ( )

El presente programa inicia con la variable A=0, después te pide que le indiques un operador (Multiplicación, resta, suma y división) posteriormente de que le indiques el operador te pide un número, al número ingresado se le efectuara la operación que el usuario indico, luego el valor de la variable A será el resultado de la operación efectuada, este proceso se repetirá, solo que ahora el valor de A comenzara con el resultado de la operación previamente realizada

{

4

5 **i n t** a ;

6 **i n t** b ;

7 **char** c ;

8 **char** d [ 1 0 0 ] ;

9 a = 0 ;

10

11 **while** ( 1 ) {

12 p r i n t f ( ”A: %d\n” , a ) ;

13

14 p r i n t f ( ” Operador ? : ” ) ;

15 f g e t s ( d , **s i z e o f** ( d ) , s t d i n ) ;

16 s s c a n f ( d , ” %c ” , &c ) ;

17

18 p r i n t f ( ”Numero ? : ” ) ;

19 f g e t s ( d , **s i z e o f** ( d ) , s t d i n ) ;

20 s s c a n f ( d , ” %d” , &b ) ;

21

22 **i f** ( c == ’ q ’ )

23 **break** ;

24 **i f** ( c == ’Q ’ )

25 **break** ;

26 **switch** ( c )

{

27 **case** ’+ ’ :

28 a = a + b ;

29 **break** ;

30 **case** ’ ’ :

−

31 a = a b ;

−

32 **break** ;

33 **case** ’ ’ :

∗

34 a = a b ;

∗

35 **break** ;

36 **case** ’ / ’ :

37 **i f** ( b == 0 )

{

38 p r i n t f ( ” E r r o r n” ) ;

\

39 **e l s e**

} {

40 a = a /b ;

}

41

42 **break** ;

43 **d e f a u lt** :

44 p r i n t f ( ”Op . d e s c o n o c i d a %c n” , c ) ;

\

45 **break** ;

46 }

47

48

}

49 **return** 0 ;

50 }

COEMNTARIOS

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

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

10

11 **while** ( 1 ) { //Se abre una condición: Mientras sea 1 y comienza el primer ciclo del programa

12 p r i n t f ( ”A: %d\n” , a ) ; //Imprime el valor entero correspondiente a “A” y pasa a la siguiente línea con \n

13

14 p r i n t f ( ” Operador ? : ” ) ; // Se le indica al programa que pida el tipo de operador que el usuario desea usar

15 f g e t s ( d , **s i z e o f** ( d ) , s t d i n ) ; // se utiliza el sizeof para devolver el tamaño en bytes que ocupa una variable y el stdin para limpiar el buffer y la siguiente línea ejecute correctamente

16 s s c a n f ( d , ” %c ” , &c ) ; // Se le indica al programa que el número que reciba del teclado se lo asigne al carácter c

17

18 p r i n t f ( ”Numero ? : ” ) ; //Imprime la frase: Numero?, para que el usuario proporcione algún número que se operara el valor de A

19 f g e t s ( d , **s i z e o f** ( d ) , s t d i n ) ; Se interpreta el número que ingreso el usuario, el cual puede ser de 99 dígitos, ya que en la línea 8 se puso d[100]

20 s s c a n f ( d , ” %d” , &b ) ; Se le indica al programa que el entero ingresado se le agregue a la variable b

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 división 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 división, 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

REPRESENTACION GRAFICA

