Hector Javier Salazar Alvarez

**Teclear y ejecutar los programas que vienen en las diapositivas 12, 15 , 16, 18 y 19 de Clase03.pdf. (les recomiendo MUCHO que no corten y peguen, que los tecleen desde principio a fin).**

**/\*-----------------DIAPOSITIVA 12-----------------\*/**

**#include <stdio.h>**

**int main(void) {**

**printf("Hola clase! \n");**

**getchar();**

**return 0;**

**}**

**/\*-----------------DIAPOSITIVA 15-----------------\*/**

**#include <stdio.h>**

**int main(void) {**

**int edad;**

**printf("\n introduce tu edad: ");**

**scanf("%d", &edad);**

**printf("tu edad es %d \n", edad);**

**getchar();**

**return 0;**

**}**

**/\*-----------------DIAPOSITIVA 16-----------------\*/**

**#include <stdio.h>**

**int main(void) {**

**int a;**

**a = 0;**

**a = a + 5;**

**if (a == 5) {**

**printf("\n a valia 0 antes \n");**

**}**

**getchar();**

**return 0;**

**}**

**/\*-----------------DIAPOSITIVA 18-----------------\*/**

**#include <stdio.h>**

**int main(void) {**

**float cels, farh;**

**farh = 95.0;**

**cels = 5.0 \* (farh - 32.0) / 9.0;**

**printf(">>>%f F son %f C\n", farh, cels);**

**getchar();**

**return 0;**

**}**

**/\*-----------------DIAPOSITIVA 19-----------------\*/**

**#include <stdio.h>**

**int main(void) {**

**int age;**

**printf("please your age: ");**

**scanf("%d", &age);**

**if (age < 100) {**

**printf("you are pretty young!\n");**

**} else if (age == 100) {**

**printf("you are old\n");**

**} else {**

**printf("you are really old\n");**

**}**

**getchar();**

**return 0;**

**}**

b) Probar los modificadores de formato para la función printf() que se indican en la

diapositiva 13, sobre el programa de la dipositiva 12.

#include <stdio.h>

int main(void)

{

    // Uso de \a (alerta/bell) en este genera un sonido de alerta del sistema operativo al ejecutarse

    printf("Hola clase! \a\n");

    // Uso de \b (backspace) para sobrescribir

    printf("Hola clase! \bX\n");

    // Uso de \f (form feed) - Hace el mismo efecto de la máquina de escribir de retroceder

    printf("Hola clase! \fNueva linea\n");

    // Uso de \n (newline) - salto de linea

    printf("Hola clase! \nNueva linea\n");

    // Uso de \r (carriage return) - regresa al inicio de la linea

    printf("Hola clase! \rRegresado al inicio\n");

    // Uso de \t (tabulacion) - inserta una tabulacion horizontal

    printf("Hola clase! \tCon tabulacion\n");

    // Uso de \v (vertical tab) - puede ser menos visible

    printf("Hola clase! \vNueva linea vertical\n");

    // Uso de \\ (backslash)

    printf("Hola clase! \\ barra invertida\n");

    // Uso de \' (comilla simple)

    printf("Hola clase! \'Comilla simple\'\n");

    // Uso de \" (comilla doble)

    printf("Hola clase! \"Comilla doble\"\n");

    // Uso de \000 (octal escape) - representando el caracter con codigo octal

    printf("Hola clase! \001 (octal 001)\n");

    // Uso de \xHH (hexadecimal escape) - representando el caracter con codigo hexadecimal

    printf("Hola clase! \x41 (hexadecimal 41)\n"); // 'A' en hexadecimal

    getchar();

    return 0;

}

C) Investigar el uso de la función sizeof( ) en internet o en algún libro, y modificar el

programa de la diapositiva 16 para escribir a pantalla cuántos bytes usa la variable int

a;. Usando la misma función, reportar en pantalla cuántos bytes ocupa una variable

de tipo char, float, double, long int y unsigned int.

codigo:  
int main()

{

int a;

char b;

float c;

double d;

long e;

unsigned int i;

a = 4 \* 6;

a = a + 5;

printf("el tamano que ocupa el entero es de %d\n", sizeof(a));

printf("el tamano que ocupa el char es de %d\n", sizeof(b));

printf("el tamano que ocupa el float es de %d\n", sizeof(c));

printf("el tamano que ocupa el double es de %d\n", sizeof(d));

printf("el tamano que ocupa el long es de %d\n", sizeof(e));

printf("el tamano que ocupa el unsigned int es de %d\n", sizeof(i));

getchar();

return 0;

}

Definiciones:

* Tipo de dato: int

Resultado: 4 bytes

Explicación: En la mayoría de las plataformas modernas, un int ocupa 4 bytes.

Esto es común en muchas arquitecturas y compiladores.

* Tipo de dato: char

Resultado: 1 byte

Explicación: Un char siempre ocupa 1 byte por definición del estándar C.

es el tipo de dato más básico y se utiliza para representar caracteres.

* Tipo de dato: float

Resultado: 4 bytes

Explicación: En la mayoría de las arquitecturas, un float ocupa 4 bytes.

Esto es típico en muchas plataformas y compiladores.

Este formato proporciona una representación de números decimales

con una precisión adecuada para muchas aplicaciones sin usar demasiada memoria

* Tipo de dato: double

Resultado: 8 bytes

Explicación: Un double generalmente ocupa 8 bytes en muchas plataformas modernas.

Esto permite una mayor precisión y un rango más amplio de valores en comparación con float.

* Tipo de dato: long

Resultado: 4 bytes

Explicación: En algunas plataformas, como muchas plataformas de 32 bits, un long es de 4 bytes.

Sin embargo, en plataformas de 64 bits, un long puede ser de 8 bytes.

* Tipo de dato: unsigned int

Resultado: 4 bytes

Explicación: Un unsigned int típicamente ocupa 4 bytes, similar a un int.

es un entero sin signo que, en la mayoría de las plataformas, ocupa el mismo tamaño que un int,

es decir, 4 bytes.

La única diferencia es que unsigned int no puede representar valores negativos.

**d) Hacer un programa en C que convierta una cantidad dada en pesos mexicanos a**

**dólares Estado-Unidences y Canadiences (las dos conversiones en el mismo**

**programa). Debe de leer la cantidad de pesos desde el teclado (con scanf() ).**

#include <stdio.h>

#include <math.h>

int main(void)

{

    int opcion;

    float mxn, usa, canada;

    printf("Hola Bienvenido, ingresa 1 si quieres pasar tus pesos a dolares de usa o 2 a dolares canadiences\n");

    scanf("%d", &opcion);

        if (opcion==1){

            printf("ingresa tus pesos mexicanos\n");

            scanf("%f", &mxn);

            usa=mxn\*19.13;

            printf("tienes %.2f en dolares estadounidenses\n\a", usa);

        }

        else if (opcion==2){

            printf("ingresa tus pesos mexicanos\n");

            scanf("%f", &mxn);

            canada=mxn/14.12;

            printf("tienes %.2f en dolares canadiences\n\a", canada);

        }

        else {

        printf("Opcion no valida.\n");

        return 1;  // Salida del programa en caso de opción no válida

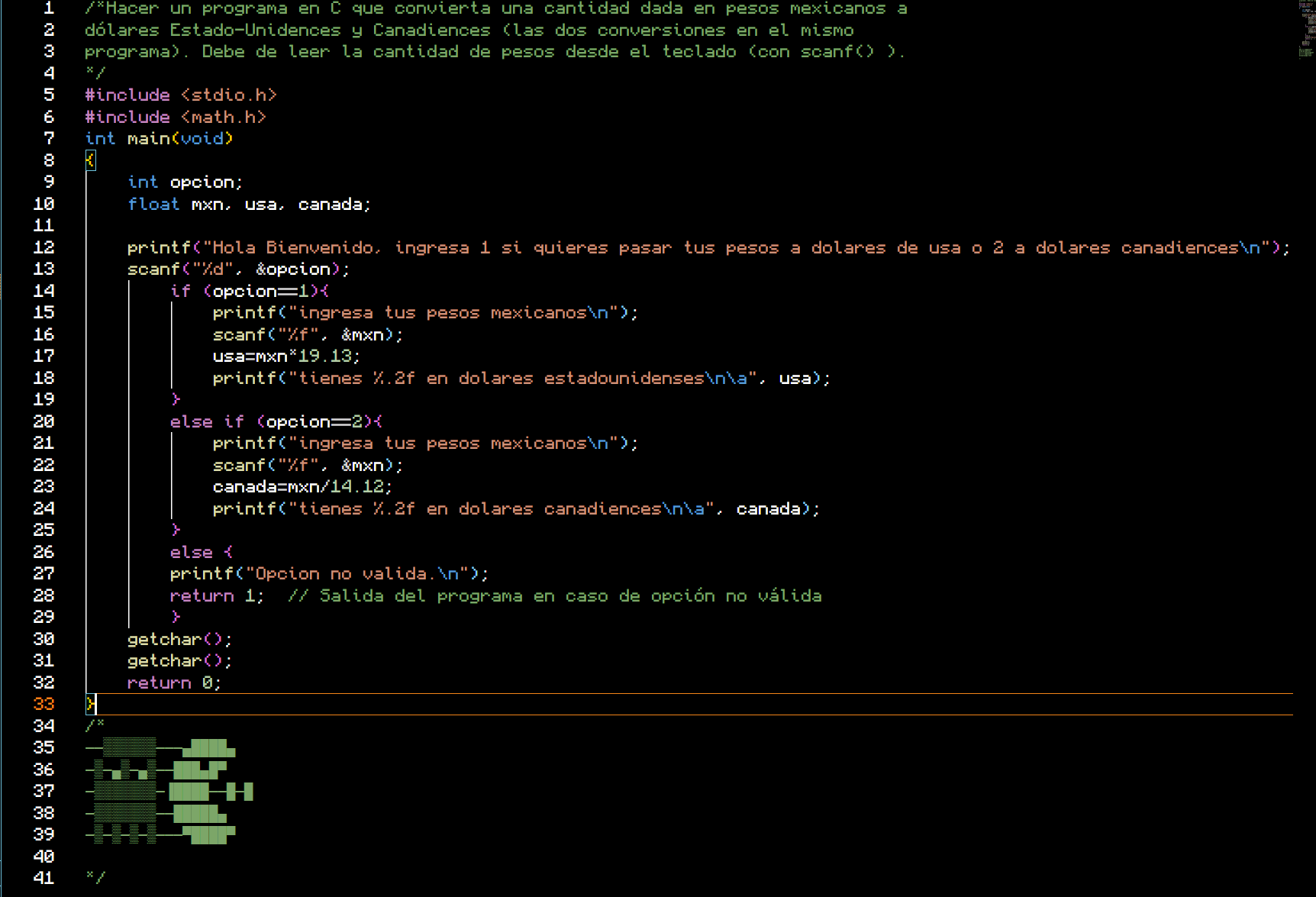
        }

    getchar();

    getchar();

    return 0;

}



Lo mostrado en la imagen anterior es el código adjunto a la tarea como D).c

Evidencia de que fue compilado y ejecutado correctamente:Texto

Descripción generada automáticamente

**e) Hacer un diagrama de flujo o pseudocódigo para resolver el problema: dados tres números enteros guardados en las variables int a,b,c; indique cuál es el mayor de los 3.**

Diagrama

Descripción generada automáticamente

**f) Implementar el problema anterior en un programa en C que lee los datos desde el teclado.**

Código:

#include <stdio.h>

int main(void)

{

    int a,b,c;

    printf("Hola Bienvenido, ingresa tu primer valor que quieres comparar\n");

    scanf("%d",&a);

    printf("ingresa tu segundo valor que quieres comparar\n");

    scanf("%d",&b);

    printf("ingresa tu tercer valor que quieres comparar\n");

    scanf("%d",&c);

    if (a>b && a>c)

        printf( "%d es el mayor", a);

    else if (b>c && b>a){

        printf( "%d es el mayor", b);}

    else

        printf("%d es el mayor",c);

        getchar();

    return 0;

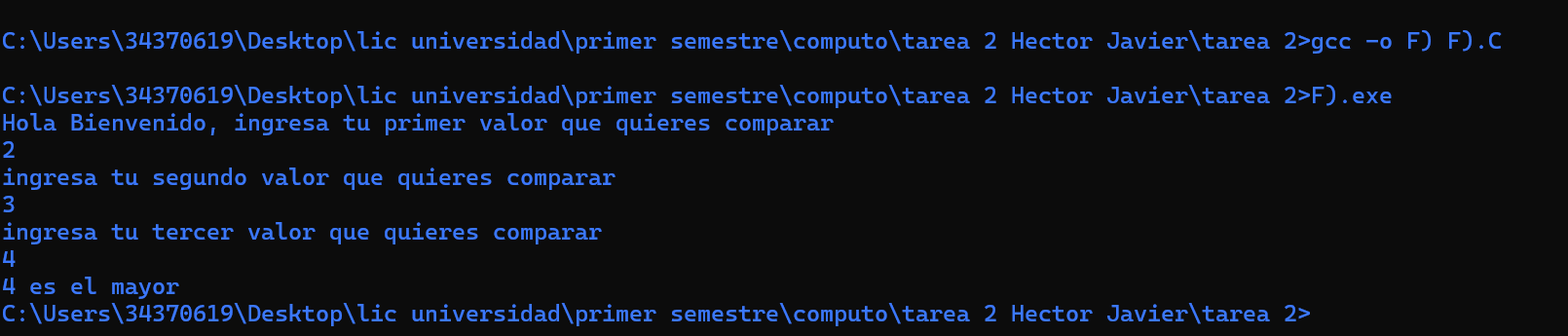
}

Texto

Descripción generada automáticamente

Lo mostrado en la imagen anterior es el código adjunto a la tarea como F).c

EVIDENCIA de que fue compilado y ejecutado correctamente:



**g) Hacer un programa en C que calcule el área de un trapecio isóceles. Leer los datos desde el teclado.**

Codigo:

#include <stdio.h>

int main(void){

    float B,b,h,A;

    printf("Hola, Bienvenido para saber el area de tu trapecio isoceles teclea la altura\n");

    scanf("%f",&h);

    printf("teclea la base mayor\n");

    scanf("%f",&B);

    printf("teclea la base menor\n");

    scanf("%f",&b);

    A=(((B+b)\*h)/2);

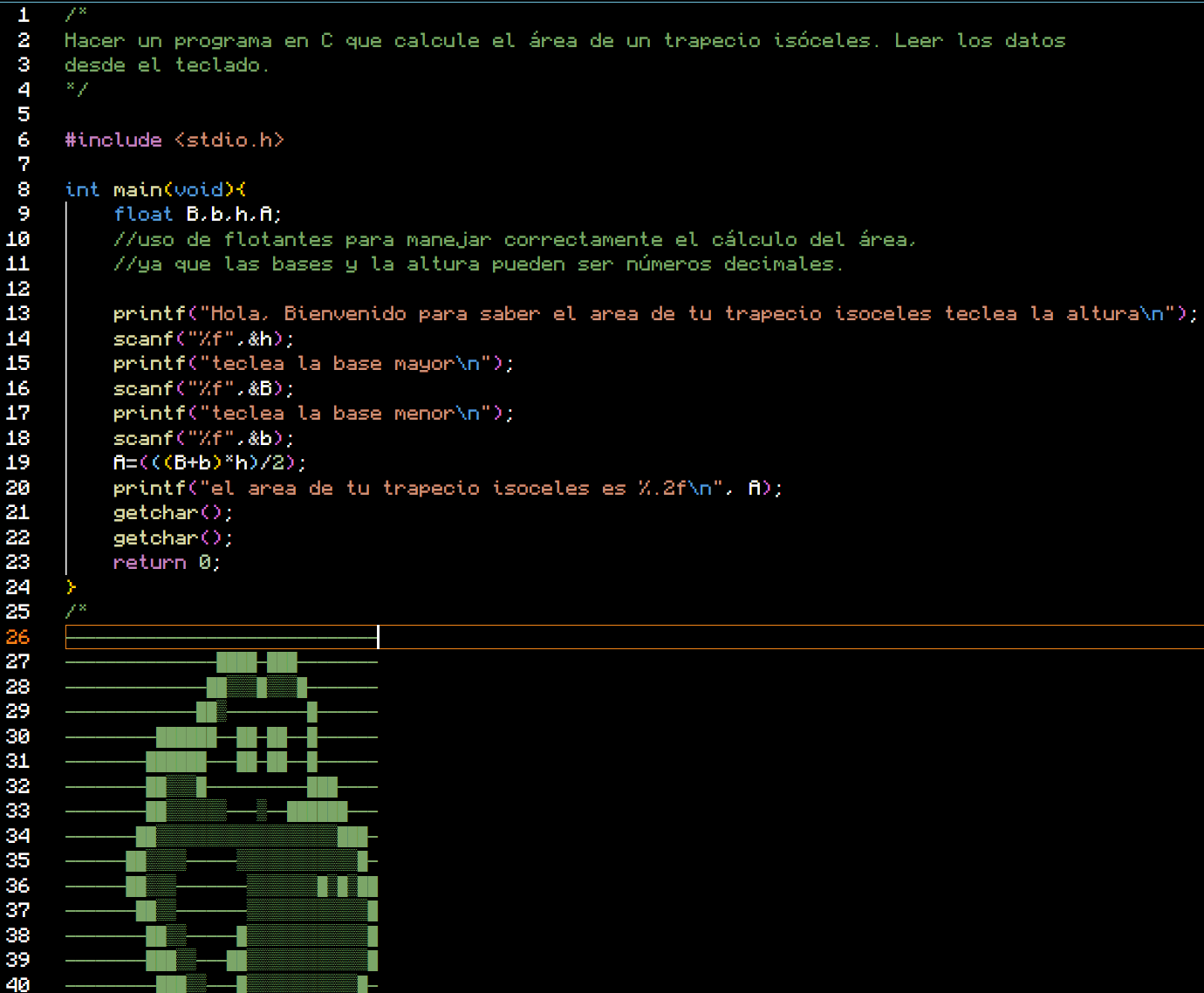
    printf("el area de tu trapecio isoceles es %.2f\n", A);

    getchar();

    getchar();

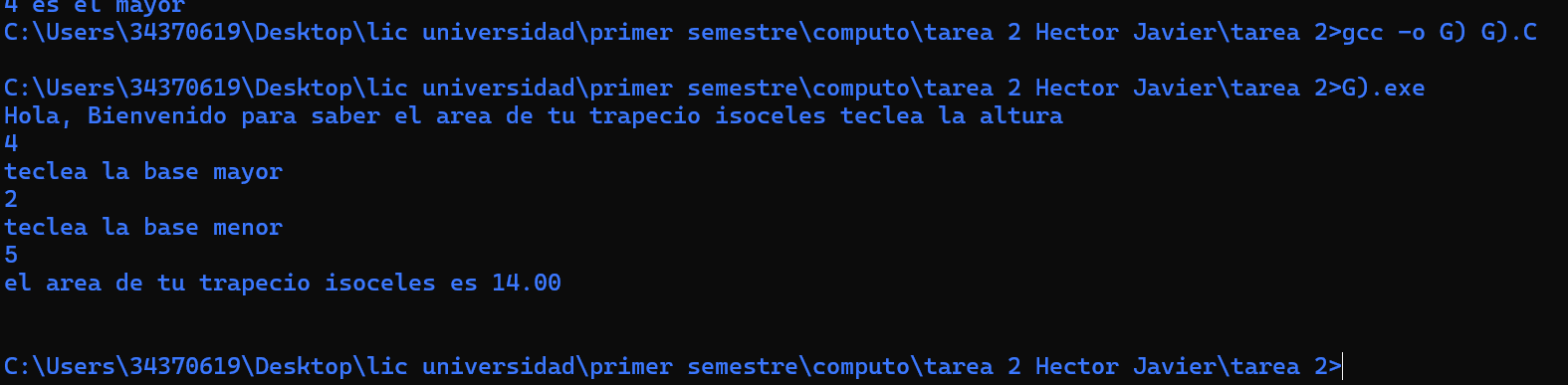
    return 0;

}



Lo mostrado en la imagen anterior es el código adjunto a la tarea como G).c

EVIDENCIA de que fue compilado y ejecutado correctamente:



**h) Hacer un programa que, dados dos círculos, cada uno de ellos definido por la**

**coordenada de su centro float x,y; y su radio float r;, indique en pantalla si**

**hay traslape entre ellos o no. Leer los datos desde el teclado.**

Código:

#include <stdio.h>

#include <math.h>

int main(void){

    float h1,k1,r1,h2,k2,r2,d;

    printf("Buenas, para saber si dos circulos se traslapan escribe el primer radio\n");

    scanf("%f",&r1);

    printf("el segundo radio?\n");

    scanf("%f",&r2);

    printf("el primer centro en x?\n");

    scanf("%f",&h1);

    printf("el primer centro en y?\n");

    scanf("%f",&k1);

    printf("el segundo centro en x?\n");

    scanf("%f",&h2);

    printf("el segundo centro en y??\n");

    scanf("%f",&k2);

    d=sqrt(((h2 - h1)\*(h2 - h1)) + ((k2 - k1)\*(k2-k1)));

    if (d < (r1 + r2) && d > fabs(r1 - r2)){

        printf("si hay traslape entre ellos\n");

    }

    else

        printf("no hay traslape entre ellos\n");

    getchar();

    getchar();

    return 0;

}



Lo mostrado en la imagen anterior es el código adjunto a la tarea como G).c

EVIDENCIA de que fue compilado y ejecutado correctamente:

Texto

Descripción generada automáticamente