**Objetivo**: Trabajar con estructuras

**La salida del programa debe ser como en el ejemplo.**

**Ejercicio 1.-** Escribir un **programa** que:

* Pida al usuario las coordenadas de un círculo
* Lea de un fichero de texto las coordenadas de puntos del plano, si un punto **es interior al círculo** lo añade a una lista de puntos. Este proceso finalizará bien cuando finalice el fichero, o bien cuando la lista esté llena.
* Finalmente, escribe en consola la lista creada de puntos interiores al círculo.

En el fichero de texto, aparece, para cada punto, dos valores reales que se corresponden con su abscisa y su ordenada.

NOTA: Un punto es interior a un circulo si la distancia entre el punto y el centro del circulo es menor (o igual) al radio.

En este programa deberá utilizarsela siguiente **declaración de estructuras**, que representan información de un punto del plano (abscisa y ordenada), de una lista de puntos y de un círculo (radio y centro):

#define N 20 // máxima cantidad de puntos

struct Tpunto{ // punto en un plano, x abscisas, y ordenadas

 float x;

 float y;

};

struct TlistaPuntos{ //lista de puntos del plano

 int tam; //cantidad de valores en el vector puntos

 struct Tpunto puntos[N];

};

struct Tcirculo //circulo: punto central y longitud del radio

{

 struct Tpunto centro;

 float radio;

};

Y las siguientes funciones:

1. Función que a partir de dos puntos del plano, devuelva la distancia entre dichos puntos.

Sabemos que la distancia entre dos puntos (x1, y1) y (x2, y2) se calcula mediante la fórmula:

$$distancia= \sqrt{(x\_{2}-x\_{1})^{2}+(y\_{2}-y\_{1})^{2}}$$

1. Función de prototipo:

 void leerPunto (FILE \*f, struct Tpunto \*p);

Que lee las coordenadas de un punto del plano (abscisa y ordenada) y las guarda en la variable apuntada por p.

1. Función de prototipo:

 void leerCirculo (struct Tcirculo \*c);

Que pide por consola al usuario las coordenadas del centro (abscisa y ordenada) y el radio de un círculo y guarda estos datos en la variable apuntada por c.

1. Una función que a partir de una lista de puntos y de un punto, añade el punto a la lista: prototipo:

 void aniadirPunto (struct TlistaPuntos \*l, struct Tpunto p);

 En la lista hay capacidad para insertar el punto.

1. Una función de prototipo:

 void escribirLista (struct TlistaPuntos l);

Que a partir de una lista de puntos escribe por consola la coordenadas de todos los puntos de la lista l.

Primero escribirá el siguiente encabezado:

PUNTOS DE LA LISTA

ORDEN ABSCISA ORDENADA

y para cada punto:

*nn ( xx.xx yy.yy)*

Donde:

* *nn* es el número de orden de ese punto en la lista (debe escribirse en un campo de 3 posiciones),
* *xx.xx yy.yy* sonsus coordenadas (deben escribirse en un campo de 6 posiciones con 2 decimales cada una).

Ejemplos de funcionamiento:





Copia a continuación el programa desarrollado: