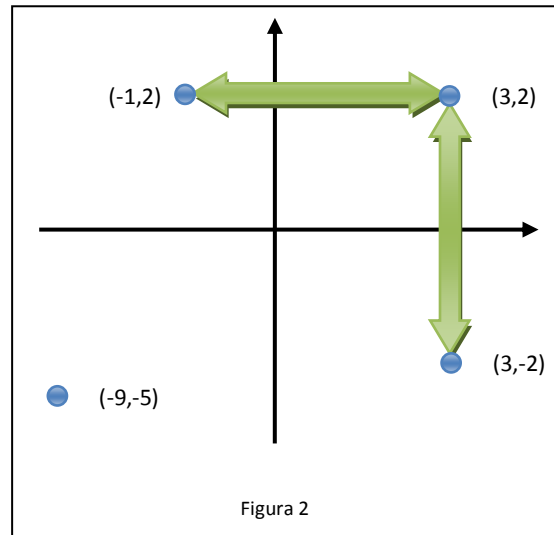
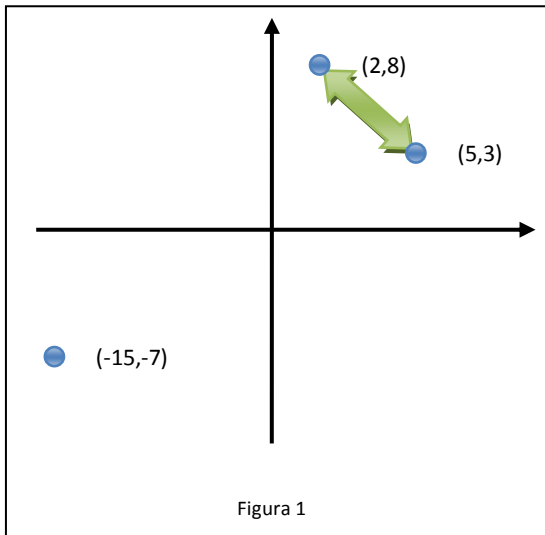


Puntos cercanos

Dado un *array* con una cantidad par de elementos que representan una secuencia de coordenadas x, y de puntos en el plano donde la x aparece siempre delante de la y. Por ejemplo, los puntos (5,3), (2,8) y (-15,-7) se representan en este *array* de la siguiente manera: {5, 3, 2, 8, -15, -7}

Implemente un método que devuelva el par de puntos entre los cuales existe la menor distancia. La respuesta debe ser dada en el mismo formato del *array* inicial. Asuma que en el *array* original existen al menos 2 puntos. Si existe más de un par de puntos con distancia mínima, puede devolver cualquiera de ellos, pero solo un par. La longitud del *array* resultante deberá ser siempre 4. Si por ejemplo el *array* original fuera {5, 3, 2, 8, -15, -7} solo existe un único par de puntos con distancia mínima, por lo que la respuesta debería ser el *array* {5, 3, 2, 8} o {2, 8, 5, 3}. En la Figura 1 se muestra la representación aproximada de los puntos del ejemplo en el plano. Otro ejemplo sería el *array* original de puntos {-1, 2, 3, 2, 3, -2, -9, -5}, en este caso hay dos pares de puntos con distancia mínima, como se muestra en la Figura 2, por ello tanto la respuesta {-1, 2, 3, 2} como la respuesta {3, 2, 3, -2} son correctas.



Usted deberá implementar una biblioteca de clases (DLL) `ExamenPuntosCercanos` con un método dentro de la clase `Plano` con nombre `PuntosMasCercanos`.

```
namespace ExamenPuntosCercanos
{
    public class Plano
    {
        public static int[] PuntosMasCercanos(int[] puntos)
        {
            // TODO: Implemente aquí su algoritmo
        }
    }
}
```