Caso N°1: Ejemplo Original
Descripción: Ejemplo original presentado en el enunciado.

in:
3
1.49 5.00
1.50 3.00
1.51 -3.00
2.10 6.00
2.00 -6.00
1.00 2.00
2.30 -35.00
1.30 -92.00
1.00 1.00

out:
1 2
2 1 3

Caso N°2: No hay tiros validos
Descripción: Verificamos el caso en el que de no haber tiros validos el programa funcione correctamente.

in:
3
1.50 180.00
1.50 175.00
1.50 -90.01
2.00 125.00
2.00 -125.00
2.10 -145.00
3.50 -95.75
3.50 -115.45
3.00 90.50

out:
0
0

Caso N°3: Jugador Preciso

Descripción: Verificamos que la consistencia sea propiamente calculada presentando un jugador con tres tiros idénticos.

in:

```
3
1.50        7.00
1.50        7.00
1.50        7.00
2.00        125.00
2.50        -125.00
1.00        4.00
3.50        -95.75
2.50        -115.45
1.50        90.50
```

out:

```
1
1 2
```

Caso N°4: Desempate por ángulos

Descripción: Dado que dos jugadores tienen la misma similitud en sus distancias, deben desempatar en base a la consistencia de sus ángulos.

in:

```
3
1.50        7.00
1.50        -7.00
1.50        7.00
2.50        4.00
2.50        4.00
2.50        4.01
3.50        -95.75
2.50        -115.45
1.50        90.50
```

out:

```
1 2
2 1
```

Caso N°5: Descalificación de consistencia
Descripción: Verificar que no se tenga en cuenta al lanzador para el premio por consistencia por tener un tiro descalificado por el ángulo aunque tenga la mejor varianza.

in:
3
1.50 7.00
1.50 8.00
1.50 9.00
2.50 3.00
2.50 4.00
2.50 5.10
3.50 89.90
4.50 89.90
3.50 90.01

out:
1 2
2 3 1

Caso N°6: Gana por tener mejor ángulo
Descripción: Verificar que calcula correctamente el modificador de distancia.

in:
3
2.50 60.00
2.50 60.00
2.50 60.00
2.30 1.00
2.30 2.00
2.30 3.00
1.00 102.00
1.00 104.00
1.00 0.00

out:
1 2
2 1 3

Caso N°7: Descalificado por 3 tiros inválidos
Descripción: Verificar que cuando un concursante realiza 3 tiros descalificados este no aparece en el podio de distancia.

in:
3
2.50 60.00
2.50 60.00
2.50 60.00
2.30 1.00
2.30 2.00
2.30 3.00
1.00 100.00
1.00 100.00
1.00 100.00

out:
1 2
2 1

Caso N°8: Caso de fatiga
Descripción: Verificar que se ejecute en un tiempo aceptable en el peor de los casos.

in:
1000000
0.00 0.00
0.00 0.00
[dist]* 0.00
…

* detalle de la entrada: el valor de la tercera distancia, [dist], es mayor al del concursante anterior, de manera que, al calcular la varianza de distancia cada concursante tendrá un valor mayor al anterior. El ángulo es el mismo ya que la distancia tiene el mismo peso en la comparación que el ángulo, pero este último no puede variar lo suficiente para que todos tengan diferente varianza (no puede haber empate).

out:
1 2 3 … 999998 999999 1000000
1000000 999999 999998 … 3 2 1

detalle de la salida: por lo mencionado en el detalle de la entrada, la consistencia irá decreciendo para cada concursante y la distancia aumentando, por esto la primera línea tiene a los concursantes en orden ascendente y la segunda en orden decreciente.

Observación del caso de fatiga: Por la manera en que se maneja la escritura del archivo el programa utiliza mucha memoria, ya que pasa el contenido de la salida como un string para a la clase que escribe el archivo de salida. Se podría solucionar cambiando la forma en la que la clase que resuelve le pasa los datos a la clase que escribe el archivo.