

ПРОЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Плодив, 14 юни 2008 г.

Група В (9-10 клас)

Задача В1. ОКРЪЖНОСТИ

В равнината са дадени са n окръжности: C_1, C_2, \dots, C_n .

Разглеждаме неориентиран граф с върхове дадените окръжности, като между две окръжности има ребро, тогава и само тогава, когато окръжностите имат точно две общи точки.

Напишете програма **circle**, която намира броя на ребрата в най-късия път от C_1 до C_n .

На първия ред на стандартния вход е записано числото n ($2 \leq n \leq 1000$), а на следващите n реда са записани по три цели числа x, y, r , задаващи координатите на центъра и дължината на радиуса на поредната окръжност; $-10000 < x, y < 10000$; $0 < r < 10000$. Номерацията на окръжностите е според появяването им във входния файл.

На стандартния изход да се изведе търсеният брой или -1 , ако не съществува път от C_1 до C_n .

ПРИМЕР

Вход

```
3
0 0 1
4 0 4
1 0 2
```

Изход

```
2
```


Задача В2. ТРИЪГЪЛНИЦИ

Дадени са n ($2 < n < 1000$) тройки точки с координатите си в равнината. Разглеждаме само тези тройки точки, които образуват триъгълник, които е такъв, че има страна, успоредна на някоя от координатните оси и има целочислено лице. Задачата ни е да разпределим тези триъгълници на две подмножества така, че сумата от лицата на триъгълниците от едното подмножество да е равна или възможно най-близка до сумата от лицата на триъгълниците от другото подмножество. Двете подмножества трябва да нямат общи елементи и обединението им да съвпада с множеството на всички разглеждани триъгълници. Нека s_1 и s_2 означават сумите от лицата на триъгълниците от едното и другото подмножество. Напишете програма **triangle**, която намира стойности на s_1 и s_2 така, че да е минимална абсолютната стойност на разликата $s_1 - s_2$.

От стандартния вход се въвежда n . От всеки от следващите n реда се въвеждат по шест цели числа в диапазона от -100 до 100 , задаващи трите двойки координати на поредната тройка точки. Спазено е изискването данните във входа да са такива, че множеството на разглежданите триъгълници да има поне два елемента.

На стандартния изход да се изведат, разделени с един интервал две цели числа, равни на намерените от програмата стойности за s_1 и s_2 , като първата от тези стойности трябва да е по-малка или равна на втората.

ПРИМЕР

Вход

```
6
8 4 9 4 8 1
1 0 0 0 4 4
9 8 3 5 1 5
8 3 4 5 1 3
8 3 4 5 1 3
9 4 3 2 8 5
```

Изход

```
9 10
```

Пояснение: В примера първият триъгълник има лице 1.5, което не е целочислено и затова не го разглеждаме. Вторият триъгълник има лице 2, третия – 3, четвъртия и петият геометрически съвпадат и имат лице 7. Тези триъгълници имат страна, която е успоредна на координатна ос и имат целочислени лица. Шестият триъгълник не разглеждаме, защото той няма страна, успоредна на поне една от координатните оси. В едното подмножество ще попаднат вторият и четвъртия триъгълник, чиято сума от лицата е 9, а в другото подмножество – третият и петият, чиято сума от лицата е 10.

Задача В3. ПАРЕТО

Нека p_1 и p_2 са точки от равнината, съответно с координати (x_1, y_1) и (x_2, y_2) . Казваме, че точката p_2 е по-добра от p_1 , ако $x_1 \leq x_2$ и $y_1 \leq y_2$. Разглеждаме дадено множество от краен брой различни точки в равнината. Една точка p от това множество се нарича оптимална по Парето, ако в множеството не съществуват по-добри точки от нея. Една точка q от множеството се нарича субоптимална по Парето, ако не е оптимална и ако точките от множеството, които са по-добри от нея, са оптимални по Парето за множеството. Напишете програма **pareto**, която извежда броя на оптималните и субоптималните точки в дадено множество.

От първия ред на стандартния вход се въвежда броя n на точките от даденото множество ($0 < n < 500\,000$). От всеки от следващите n реда се въвеждат по две цели положителни числа в диапазона от 1 до 1 000 000, задаващи координатите на точка от даденото множество. Спазено е изискването във входа да няма повтарящи се двойки координати.

На стандартния изход да се изведат, разделени с един интервал две цели числа, равни на търсения брой оптимални и субоптимални точки.

ПРИМЕР

Вход

```
6
1 1
2 1
3 1
1 2
2 2
1 3
```

Изход

```
3 2
```