

шената е разделена на двадесет номерирани и еднакви по размер сектора, като сектор 20 е симетричен спрямо ординатната ос и секторите са номерирани по посока на часовниковата стрелка както следва: 20, 1, 18, 4, 13, 6, 10, 15, 2, 17, 3, 19, 7, 16, 8, 11, 14, 9, 12, 5.

В центъра на дъската са изобразени два концентрични кръга - с радиуси 0,5 cm и 1 cm, попадението в които се оценява съответно с 50 точки и 25 точки. Съществуват и два други концентрични обръча – вътрешен и външен. Вътрешният се определя от две окръжности с радиус 9,5 и 10,0 cm, а външният – от окръжности с радиуси 19,5 и 20,0 cm. Ако дадено попадение е между външния и вътрешния обръч или между вътрешния обръч и кръговете в центъра, се дават брой точки, равни на номера на сектора. Когато има попадение във външния обръч, играчът получава броя на точките за съответния сектор, умножени по 2. Ако попадението е във вътрешния обръч, точките от сектора се умножават по 3. Попадането на стреличка извън външния обръч не носи точки.

Когато дадено попадение е на границата на някой от гореописаните кръгове или обръчи, се счита, че попадението е в тази зона, която носи повече точки. Когато попадението е на границата между два сектора, се счита, че стреличката е попаднала в по-далечния сектор по посока, обратна на часовниковата стрелка (например, попадение между сектори 1 и 20 се счита в сектор 20, между 20 и 5 – в сектор 5, а между 17 и 2 – в сектор 2).

Във всеки рунд играчът хвърля 3 стрелички, като точките за рунда са сума от отделните попадения. Максимално възможният резултат е 180 точки (ако играчът попадне и с трите стрелички във вътрешния обръч на сектор 20).

Нека долният ляв ъгъл на дъската има координати (0,0), а горният десен – координати (48,48). Направете програма **darts**, която отчита броя точки за даден рунд при зададени координати на трите попадения.

Вход: От стандартния вход се въвеждат три двойки числа (x,y) – координатите на стреличките, с точност не повече от три знака след десетичната точка.

Изход: На стандартния изход се извежда единствено цяло число – броят точки за съответния рунд.

Примерен вход:

2.212 1.423 24 24 24 34

Примерен изход:

110

Решение:

Преди всичко, изчисляваме центъра на мишената с координати (24,24). За да се реши задачата, всъщност трябва да се открие отговорът на три подзадачи:

1. На какво разстояние от центъра на мишената е попадението? Това разстояние може да се сметне чрез Питагоровата теорема или разстоянието между точка А (x_1, y_1) и точка В (x_2, y_2) е $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Според разстоянието от центъра можем да направим следните изводи:

- При разстояние по-малко или равно на 0.5, се дават 50 точки.
- При разстояние по-малко или равно на 1.0, се дават 25 точки.
- При разстояние по-голямо от 20.0, се дават 0 точки.

В тези случаи попадението е отчетено и алгоритъмът приключва до тук.

В останалите случаи се определя коефициент за умножение М:

- При разстояние между 9.5 и 10.0, М=3.
- При разстояние между 19.5 и 20.0, М=2.
- Във всички останали случаи М=1.

2. Какъв е ъгълът между попадението и абсцисата на координатна система с център центъра на мишената? Ъгълът в градуси се пресмята по формулата:

$$\angle \alpha = \text{atan}(y/x) * (180/\pi);$$

3. В кой сектор е попадението? Тъй като сектор 20 е симетричен спрямо ординатата, можем да се досетим, че обхваща интервала $[81^\circ, 99^\circ)$. Нормализираме получения в подзадача 2 ъгъл: $\angle \alpha = \angle \alpha + 279^\circ$. Ако $\angle \alpha > 360^\circ$, то го нормализираме чрез $\angle \alpha = \angle \alpha - 360^\circ$. По този начин сектор 20 се намира в ъгловия диапазон от 0° до 18° , сектор 5 – от 18° до 36° и т. н.

Всъщност, тъй като секторите са през 18 градуса, можем да използваме цялата част от делението $\angle \alpha / 18^\circ$. Секторът можем да получим чрез следната таблица:

Резултат от $\alpha/18$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Сектор	20	5	1	9	14	1	8	16	7	19	3	17	2	15	10	6	13	4	18	1

Накрая, получаваме брой точки, равен на $M * \text{Сектор}$.

Забележка: Откриването на ъгъла може да стане и без *atan*. Например, може да се използват отсечки, минаващи през центъра и чрез насочено лице да се сметне в кой от образуваните сектори попада дадена точка.

```
// Solution of "Darts" problem
#include <cstdlib>
#include <iostream>
#include <math.h>
#define pi 3.14159

using namespace std;

// Sector numbers

int Sectors[20]={
    20, 5, 12, 9, 14, 11, 8, 16, 7, 19,
    3, 17, 2, 15, 10, 6, 13, 4, 18, 1};

// Center of the target

double cx=24;
double cy=24;

double DistanceFromCenter(double x, double y)
// Returns the distance from the center
{
    return sqrt((x-cx)*(x-cx)+(y-cy)*(y-cy));
}

double Angle (double x, double y)
// Angle in degrees from the center of the target
{
    double ang=atan2(y,x)*(180/pi);
    if (ang<0) ang+=360;
    return ang;
}
```

```

    }

int GetSector (double x, double y)
// Returns the number of the sector
{
    double z=Angle(x-cx,y-cy)+279;
    while (z>360) z-=360;
    return (int) floor(z/18);
}

int Score(double x, double y)
// Calculates scores according to distance and sector
{
    double DfC=DistanceFromCenter(x,y);
    if (DfC<=0.5) return 50; else
        if (DfC<=1) return 25; else
            {
                int MultiFactor=1;
                if ((DfC>=19.5)&&(DfC<=20)) MultiFactor=2; else
                if ((DfC>=9.5)&&(DfC<=10)) MultiFactor=3; else
                if (DfC>20) MultiFactor=0;
                int Sec=GetSector(x,y);
                return Sectors[Sec]*MultiFactor;
            }
    cout<<'\\n';
}

int main()
{
    double x1,y1,x2,y2,x3,y3;
    cin>>x1>>y1>>x2>>y2>>x3>>y3;
    cout<<Score(x1,y1)+Score(x2,y2)+Score(x3,y3);
    cin>>x1;
}

```

Петър Събев