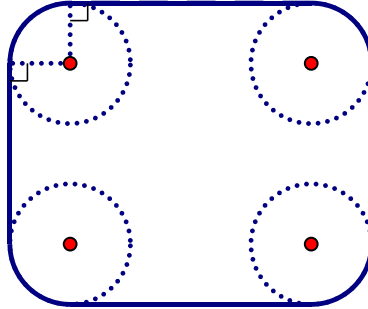


信用卡凸包

【问题描述】

信用卡是一个矩形，唯四个角作了圆滑处理，使它们都是与矩形的两边相切的 $1/4$ 圆，如下图所示。现在平面上有一些规格相同的信用卡，试求其凸包的周长。注意凸包未必是多边形，因为它可能包含若干段圆弧。



【输入格式】

输入的第一行是一个正整数 n ，表示信用卡的张数。

第二行包含三个实数 a, b, r ，分别表示信用卡（圆滑处理前）竖直方向的长度、水平方向的长度，以及 $1/4$ 圆的半径。

之后 n 行，每行包含三个实数 x, y, θ ，分别表示一张信用卡中心（即对角线交点）的横、纵坐标，以及绕中心逆时针旋转的弧度。

【输出格式】

输出只有一行，包含一个实数，表示凸包的周长，四舍五入精确到小数点后 2 位。

【输入样例 1】

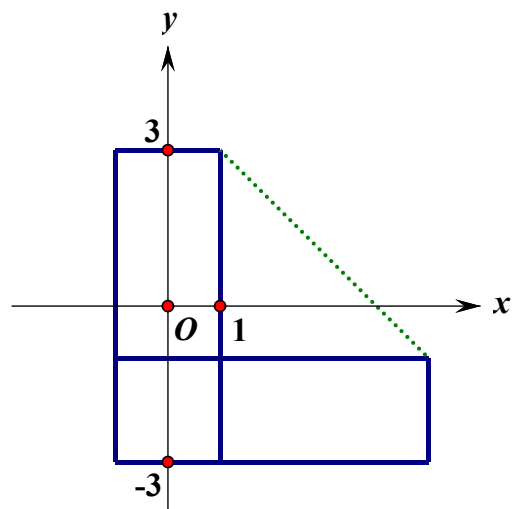
```
2
6.0 2.0 0.0
0.0 0.0 0.0
2.0 -2.0 1.5707963268
```

【输出样例 1】

```
21.66
```

【样例 1 说明】

本样例中的 2 张信用卡的轮廓在上图中用实线标出，如果视 1.5707963268



为 $\sqrt{2}$ ，则其凸包的周长为 $16 + 4\sqrt{2}$ 。

【输入样例 2】

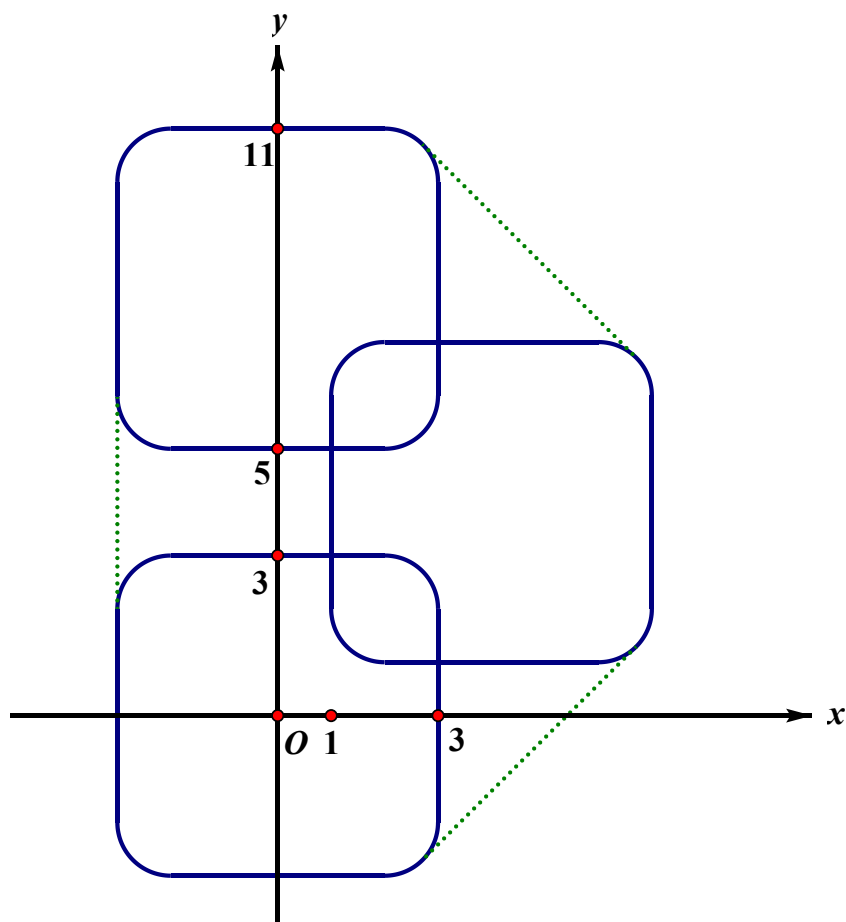
```
3
6.0 6.0 1.0
4.0 4.0 0.0
0.0 8.0 0.0
0.0 0.0 0.0
```

【输出样例 2】

```
41.60
```

【样例 2 说明】

本样例中的 3 张信用卡的轮廓在下图中用实线标出，其凸包的周长为。



【输入样例 3】

```
3
```

```

6.0 6.0 1.0
4.0 4.0 0.1745329252
0.0 8.0 0.3490658504
0.0 0.0 0.5235987756

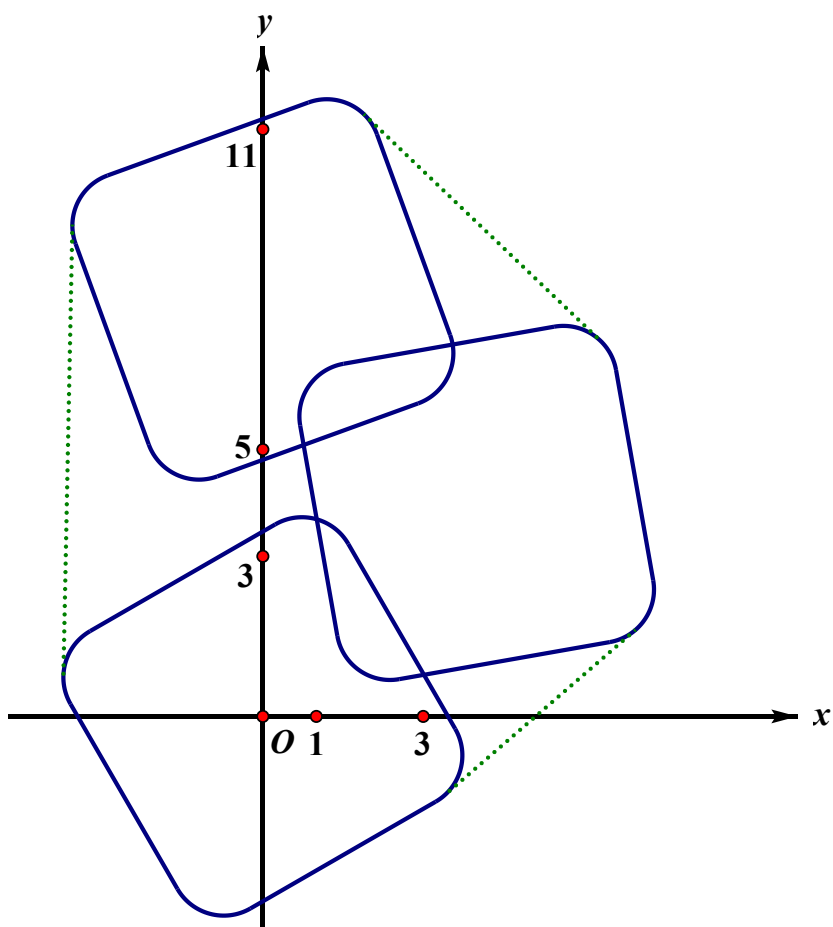
```

【输出样例 3】

41.63

【样例 3 说明】

本样例中的 3 张信用卡的轮廓在下图中用实线标出，其凸包的周长约为 41.628267652。



【数据规模】

测试数据编号	n	r	θ
1	$n = 1$	/	/
2	$n = 2$	$r = 0.0$	所有的 θ 均为 0.0
3	$n = 2$	/	所有的 θ 均为 0.0
4	$n = 2$	$r = 0.0$	/
5	$n = 2$	/	/

6	$1 \leq n \leq 100$	/	所有的 θ 均为 0.0
7	$1 \leq n \leq 100$	/	/
8	$1 \leq n \leq 10,000$	/	所有的 θ 均为 0.0
9	$1 \leq n \leq 10,000$	$r = 0.0$	/
10	$1 \leq n \leq 10,000$	/	/

对于 100% 的数据，有 $0.1 \leq a, b \leq 1000000.0$ ，以及 $0.0 \leq r < \min\{a/4, b/4\}$ ，对所有的信用卡，有 $|x|, |y| \leq 1000000.0$ ，以及 $0 \leq \theta < 2\pi$ 。

【提示】

本题可能需要使用数学库中的三角函数。不熟悉使用方法的选手，可以参考下面的程序及其输出结果：

Pascal 程序	输出结果
<pre>uses math; const Pi = 3.141592653589793; begin writeln(sin(30.0 / 180.0 * Pi) : 0 : 10); writeln(cos(60.0 / 180.0 * Pi) : 0 : 10); writeln(tan(45.0 / 180.0 * Pi) : 0 : 10); writeln(arcsin(1.0) : 0 : 10); writeln(arccos(0.0) : 0 : 10); writeln(arctan(1.0) : 0 : 10); end.</pre>	<pre>0.5000000000 0.5000000000 1.0000000000 1.5707963268 1.5707963268 0.7853981634</pre>
C++程序	输出结果
<pre>#include <iostream> #include <math.h> using namespace std; const double Pi = 3.141592653589793; int main() { cout.setf(ios::fixed); cout.precision(10); cout<<sin(30.0 / 180.0 * Pi)<<endl; cout<<cos(60.0 / 180.0 * Pi)<<endl; cout<<tan(45.0 / 180.0 * Pi)<<endl; cout<<asin(1.0)<<endl; cout<<acos(0.0)<<endl; cout<<atan(1.0)<<endl; return 0; }</pre>	<pre>0.5000000000 0.5000000000 1.0000000000 1.5707963268 1.5707963268 0.7853981634</pre>