Category: Medium

Competition: Not yet used in any competition

Question:

We know that programmers have to work with people belonging to many fields. Two computer programmers have also won a Nobel Prize in Chemistry for simulating and creating the most accurate Quantum Mechanical model of water (Field now known as Computational Chemistry). Similar problems regarding simulation of a “real” world are also encountered in game development, physics simulations and even in engineering problems in other fields. One such problem was to design a model that simulates elastic collisions for games like carom, billiards and Granular Physics (new upcoming field in Physics). Generally programmers use softwares called physics engines but they are too slow and require good processing power and memory. Try simulating the following model in a computer program:

All motion is occurring in 2 dimensions. Given number of identical balls n, each of radius r and having initial position coordinates (xi, yi) and velocities (vxi, vyi). The 2D region is bounded by a square of 100 units by 100 units. Write a program that will give average speed of a ball b after t time units. Consider elastic inter-ball and ball-wall collisions.

Here is some Physics behind the problem:

V represents final velocity and u represents initial velocity.

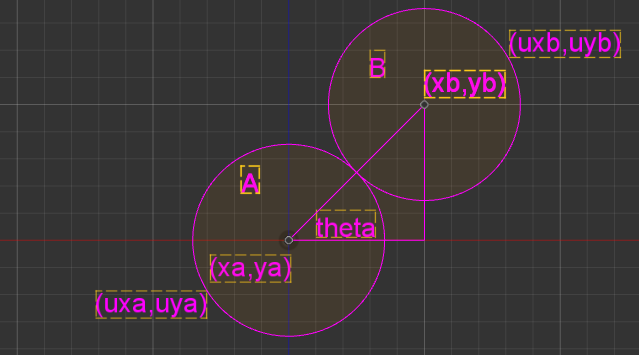
Collisions with left and right boundaries cause:

vx=-ux

Collisions with top and bottom boundaries cause:

vy=-uy

Collision between balls:



vxa = ( uyb sin(theta) + uxb cos(theta) ) cos(theta) – (uya cos(theta) – uxa sin(theta)) sin(theta)

vxa = ( uyb sin(theta) + uxb cos(theta) ) sin(theta) + (uya cos(theta) – uxa sin(theta)) cos(theta)

vxb = ( uya sin(theta) + uxa cos(theta) ) cos(theta) – (uyb cos(theta) – uxb sin(theta)) sin(theta)

vxb = ( uya sin(theta) + uxa cos(theta) ) sin(theta) + (uyb cos(theta) – uxb sin(theta)) cos(theta)

And for those who have forgotten all physics ;)

Average speed = Total Distance / Total Time

Inputs:

n: number of balls : positive integer

r: radius of ball : positive real number

t: duration in time units : positive real number

b: ball number whose average speed is needed : integer 1 to number of balls

Data regarding balls : both x and y coordinates should be entered which are real numbers between 1 and 100 followed by the x and y components of velocity which are again real numbers. Data for every ball should be entered together.

Output:

Average Speed: (Round off to nearest integer)

Example:

|  |  |
| --- | --- |
| 2  1  10  1  25 50 3 4  75 50 -3 4 | 5 |

Test Cases:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  1  10  1  25 50 10 0 | 10 |
| 2  1  10  1  25 50 10 0  50 50 0 0 | 2 |
| 2  1  10  2  25 50 10 0  50 50 0 0 | 8 |
| 2  1  50  1  25 50 10 0  50 50 0 0 | 4 |
| 2  1  50  2  25 50 10 0  50 50 0 0 | 6 |
| 2  1  10  1  25 50 3 4  75 50 -3 4 | 5 |
| 2  1  10  2  25 50 3 4  75 50 -3 4 | 5 |
| 2  1  1000  1  25 50 3 4  75 50 -3 4 | 5 |
| 2  1  1000  2  25 50 3 4  75 50 -3 4 | 5 |
| 1  1  1000 1  25 50 -2 2 | 3 |