



Escola Superior de Tecnologia e Gestão de Águeda
Universidade de Aveiro

TIUP 2009

TORNEIO INTER-UNIVERSITÁRIO DE PROGRAMAÇÃO

FIRST STAGE

Águeda, 25 of March 2009
17:00 – 20:00

<http://mooshak.estga.ua.pt/~tiup/>

Problems

- A – N Candy Boxes
- B – Solar Panels
- C – Air Traffic Control
- D – Object Similarity
- E – SquarO

Scientific Committee

Gladys Castillo <gladys@ua.pt>

Luís Seabra Lopes <lsl@ua.pt>

Joaquim Madeira <jmadeira@ua.pt>

Fábio Marques <fabio@ua.pt>

António Pereira <antoniop@ua.pt>

Mário Rodrigues <mjfr@ua.pt>

Telmo Silva <tsilva@ua.pt>

Delfim Torres <delfim@ua.pt>

General Information

1. The contest follows ICPC rules (to know more, go to <http://icpc.baylor.edu/>).
2. It has a duration of 3 hours for the 5 problems.
3. The programs must read from the standard-input and write to the standard-output.
4. The generated test inputs use the following norms:
 - There are no spaces at the end of the lines and each line ends with a carriage return.
 - No multiple spacing is being used, unless it is explicitly mentioned on the problem description.
5. The generated test outputs follow the same rules.
6. All problems have 1 second timeout.
7. Each team (3 persons maximum) should use one PC and it is strictly forbidden to use any online resource other than Mooshak or what is given in the local PC disk.
8. Further information can be obtained by clicking on “Help” on your Mooshak account screen.
9. Please ask your questions to the jury using the “Questions” button.

Compilers

Language	Compiler	Version	Command Line	Extension
C	gcc	4.2.4	gcc -ansi -lm \$file	.c
C++	gcc	4.2.4	g++ -Wall \$file	.cpp
Java	jdk	1.6.0_07	javac -nowarn \$file	.java
Pascal	Free Pascal	2.2.0	fpc -v0w -oprogram \$file	.pas
Haskell	Hugs 98	September 2006	runhugs \$file	.hs

Problem A

N Candy Boxes

Introduction

Ana decided to celebrate the New Year of 2009 with 2009 Candy Boxes...

Problem

We have 2009 boxes, each containing respectively 1, 2, 3, ..., 2009 sweets. An operation is to choose a set of boxes and from each box to remove exactly the same number of sweets. What is the minimum number of operations necessary to empty all the boxes?

Input

A positive integer N describing the number of boxes available (in the description of the problem above, $N = 2009$).

Constraints

The N given in the input belongs to the interval $[1, 10^9]$.

Output

The minimum number of operations necessary to empty all the N boxes.

Sample Input

16

Sample Output

5

Problem B

Solar Panels

Introduction

Nowadays, getting energy from renewable sources is an important scientific research area. Probably, the survival of human race will depend on the results reached in this line of investigation.

Problem

Compute the total amount of energy (in KW) produced by solar panels given the value of the following parameters:

- P1. Total area of solar panels (m^2)
- P2. Total number of days to consider
- P3. Number of sunny hours in a day
- P4. Initial efficiency rate (the efficiency of solar panels before first operation)
- P5. Absolute loss of efficiency per day

Moreover, the efficiency rate of the solar panels given the current initial efficiency for a particular day (DIE) is represented as a function of the sunny hours (h), where h goes from the first sunny hour until the sunset hour (Hm):

$$Efficiency(h) = DIE + \frac{(1 - DIE)}{2} * (1 - (\frac{h}{Hm} - 0.5)^2)$$

Note:

The solar constant (S) is the mean energy density of solar radiation in a plane perpendicular to the sun's rays just above the Earth's atmosphere. It has a value of about $1367 W/m^2$ that fluctuates daily by a few parts per thousand. Further, we consider that the panels are installed horizontally and that for 12 sunny hours, horizontal solar panels receive about $7,46 * S$ per day.

Input

Value of the five parameters P1, P2, P3, P4, P5 separated by a space character.

Output

The amount of total energy in KW produced, rounded to the second decimal place.

Sample Input

```
36 1000 8 0.98 0.000000005
```

Sample Output

242089.54

Problem C

Air Traffic Control

Introduction

Air traffic control (ATC) is a service provided by ground-based controllers who direct aircraft on the ground and in the air. The primary purpose of ATC systems worldwide is to separate aircraft to prevent collisions, to organize and expedite the flow of traffic, and to provide information and other support for pilots when able.

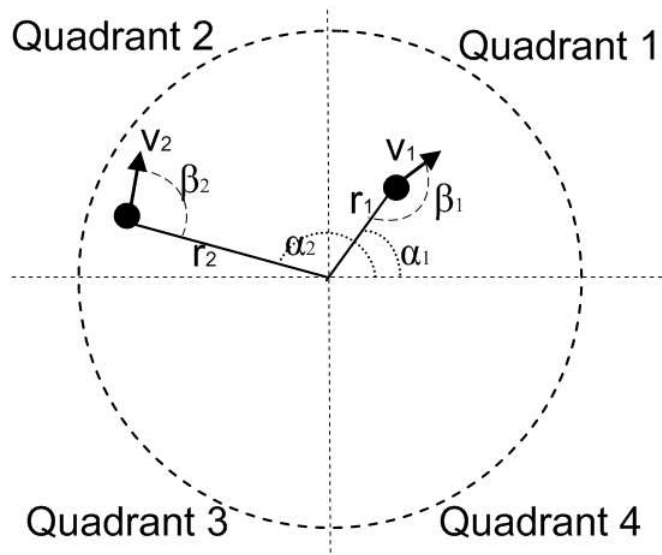


Figure 1: Airplanes coordinate set

Problem

Given two airplanes position, speed and direction, detect if a collision is likely to occur inside the radar range area - a 50Km radius circle. A collision is likely to occur every time the distance between the airplanes is less than 250 meters. Consider that both airplanes are at the same height.

Input

Two lines, one line for each airplane coordinate set. The coordinates are coded by the radar angle α , the distance r , the airplane speed v , and the airplane direction angle β , separated by blank spaces (see Figure 1).

The angles are represented by the number multiplied by π . For example, the angle $\frac{3\pi}{2}$ is represented by the number 1.5.

Constraints

- $0 \leq \alpha < 2$ ($\times \pi$ radians);
- $0 \leq r < 50000$ (m);
- $0 \leq v < 250$ (m/s);
- $0 \leq \beta < 2$ ($\times \pi$ radians).

Output

One line with the quadrant number where the collision will occur, or 0 if no collision is predicted or if it is outside the radar range.

If the airplanes are in different quadrants in the moment of the predicted collision, return the collision quadrant of the first airplane given in the input.

- $[0; \pi/2[$, 1st quadrant;
- $[\pi/2; \pi[$, 2nd quadrant;
- $[\pi; 3\pi/2[$, 3rd quadrant;
- $[3\pi/2; 2\pi[$, 4th quadrant;

Sample Input

```
0.65 27950 234 1.85
0.25 14142 200 0.0
```

Sample Output

```
3
```

Problem D

Object Similarity

Introduction

Suppose an agent has seen some objects of a given category, thought these objects were useful to its goals, and decided to search for more objects of that category. For this purpose, the agent needs to assess the similarity between the new objects and the previously seen objects. Consider that objects are represented simply as aggregations of circles.

Problem

Your problem is to implement similarity assessment for any two given objects. As mentioned, an object is a set of circles. Therefore, it is described by the radius of the circles as well as the coordinates of those circles in a convenient coordinate frame. For example, one centred in the geometric centre of the object. Assume that objects are positioned and oriented with respect to the coordinate frame in such a way that similarity is maximized, so you don't have to test different orientations. Examples of two object representations are graphically illustrated in figure 2.

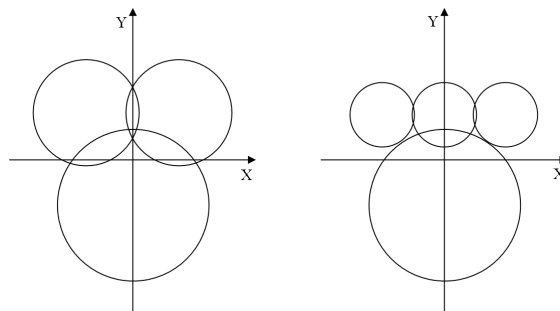


Figure 2: Graphic representation of two objects

The similarity assessment procedure is as follows:

1. Find the pair of circles, one from each object, with largest overlap. Overlap between two circles is measured as the ratio between the area of the intersection between the circles and the area of the largest of the two circles.
2. Remove the two circles from the object descriptions
3. If there are still circles in both objects, go to step 1
4. $I \leftarrow$ sum of all intersection areas identified in step 1
5. $M \leftarrow$ total area of the larger object (the one with largest total area)
6. $\text{Similarity} \leftarrow \text{round}(10 * I / M)$

Input

The input consists of two object descriptions, in the following format:

```
 $N_A$   
 $X_{A1} Y_{A1} R_{A1}$   
 $X_{A2} Y_{A2} R_{A2}$   
...  
 $N_B$   
 $X_{B1} Y_{B1} R_{B1}$   
 $X_{B2} Y_{B2} R_{B2}$   
...
```

where N_i is the number of circles of the object i and (X_{ij}, Y_{ij}, R_{ij}) are the coordinates and radius of the j -th circle of the i -th object. ($i = A, B, j = 1..N_i$)

Output

The above similarity measure.

Sample Input

This sample input corresponds to the objects given above in the figure:

```
3  
-1.25 1.25 1.4  
1.25 1.25 1.4  
0 -1.25 2  
4  
-1.7 1.25 0.85  
0 1.25 0.85  
1.7 1.25 0.85  
0 -1.25 2
```

Sample Output

```
6
```

Problem E

SquarO

Introduction

SquarO is a logic puzzle game. It was invented by Marc Lebel. The game consists of a $n \times n$ square grid, and $(n + 1)^2$ circles located on the intersections of the lines of the grid. Each cell of the grid contains an integer value k , between 0 and 4. The goal of this puzzle is to fill as many circles delimiting a cell of the grid as the value k that cell contains (see figure 3).

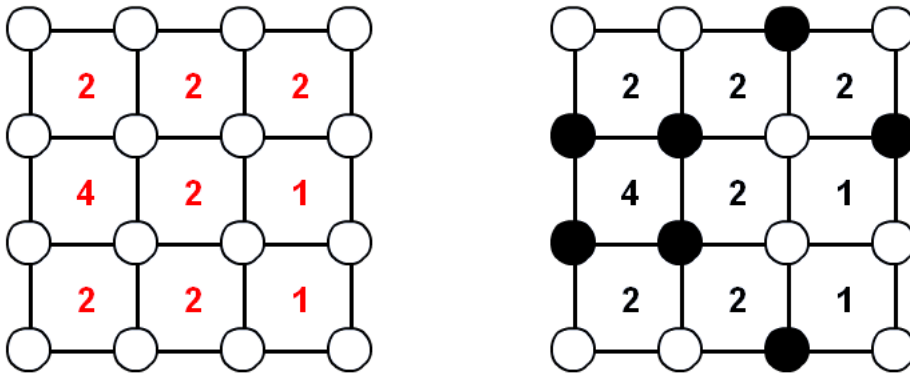


Figure 3: SquarO puzzle

Problem

Given a SquarO puzzle, determine its solution.

Input

The dimension n of the grid is given in the first line of the input. The following n lines represent the n grid lines and contain the value k of each cell. There are no blank spaces between those values.

Constraints

- $3 \leq n < 7, n \in \mathcal{N}$
- $0 \leq k \leq 4, k \in \mathcal{N}$

Output

The output is formed by the filled (1) or unfilled (0) state of the circles. Each line of the output represents a line in the grid. Each intersection is represented by the state of the corresponding circle $\{0, 1\}$, without any blank spaces.

Sample Input

3
222
421
221

Sample Output

0010
1101
1100
0010