South African Computer Olympiad
Camp 1, 2008
Day 1

Overview

Author	**Max Rabkin**	**Julian Kenwood**	**Marco Gallotta**
Problem	**repmod**	**delay**	**tour**
Source	repmod.java repmod.py repmod.c repmod.cpp repmod.pas repmod.hs	delay.java delay.py delay.c delay.cpp delay.pas delay.hs	tour.java tour.py tour.c tour.cpp tour.pas tour.hs
Input file	repmod.in	delay.in	tour.in
Output file	repmod.out	delay.out	tour.out
Time limit	1 second	2 seconds	2 seconds
Number of tests	20	10	10
Points per test	5	10	10
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Repunit Division

Author

Max Rabkin

Introduction

Since you are hoping to go to Egypt later this year, you are trying to earn some money (for the purchase of replica pyramids or, if you make a lot of money, the purchase of genuine pyramids). You have therefore challenged Bruce to a high-stakes competition of mental arithmetic.

You are confident that some high-quality pizza and a Monty Python DVD will distract him enough for you to beat him at the usual addition, subtraction, square roots, etc. Unfortunately, your appointed referee has added a rather unusual set of questions to the competition. Therefore, you have decided to cheat by writing a computer program to assist you.

Task

The problem concerns a special kind of numbers called *repunits*. The nth repunit, R_n, is the positive integer consisting of n ones. The questions ask you to find the remainder when a large repunit, R_n is divided by a small integer, k.

We use $a\%b$ to represent the remainder of a divided by b.

Example

$R_3 = 111$ is divisible by 3. Therefore $R_3\%3$ is 0. On the other hand, $R_4 = 1111 = 5 \cdot 222 + 1$, so $R_4\%5 = 1$.

Input (repmod.in)

The input consists of two lines: the first contains n and the second contains k.

Sample input

```
4
5
```

Output (repmod.out)

The output consists of a single line containing a single integer $R_n\%k$

Sample output

```
1
```

Constraints

- $1 \leq n \leq 2\,000\,000\,000$
- $1 \leq k \leq 1\,000\,000$

50% constraints

- $1 \leq n \leq 1\,000\,000$

Time limit

1 second.

Scoring

If the output is correctly formatted and contains the correct answer, you will score 100%. Otherwise, you will score zero.

Help Bruce Delay Carl

Author

Julian Kenwood

Introduction

Bruce has a predicament. He has invited Carl round to test him on his Monty Python knowledge. However, Bruce hasn't studied yet. Bruce, being a devious plotter, has scattered one-way teleporters in the tiled corridor leading to his room.

Bruce needs to know how long it will take Carl to reach his room, and has asked you to write a program to figure out how long he has until Carl reaches room. Bruce is busy brushing up on his Monty Python, so you won't be able to ask him for help.

Task

In order to get to Bruce, Carl must traverse Bruce's corridor. Your goal is to write a program that outputs the fastest time it would take, in seconds, for Carl to reach Bruce. Carl always starts on the first tile, numbered 1, and must get to the last tile, numbered N. Carl can move forward 1 tile in 1 second, or he can jump up to 6 tiles forward at a time, ignoring any teleporters in between, at a rate of 1 second per tile.

It is guaranteed that there will be no teleporters entrances or exit on tile 1 or tile N. Also, there will be at most 1 teleporter entrance or exit on any given tile. If Carl lands on a square with a teleporter entrance then he is instantly teleported to its exit.

Example

Bruce's corridor is 10 tiles long, and there are 2 teleporters. The first teleporter goes from tile 8 to tile 2, and the second teleporter goes from tile 5 to tile 6.

The quickest path is the following:

- Carl moves to the teleporter entrance on tile 5 (4 seconds)
- Carl gets teleported from 5 to tile 6 (0 seconds)
- Carl moves to Bruce on tile 10 (4 seconds)

This gives a total time of 8 seconds.

Input (delay.in)

The input consists of a number of lines. The first line contains two numbers, N and K. The next K lines will each contain two numbers, S_i and E_i. These numbers refer to the number of the tile where the teleporter entrance and teleporter exit is respectively.

Sample input

```
10 2
8 2
5 6
```

Output (delay.out)

The output will be a single line containing the number of squares Carl must move in order to get to Bruce, if there is no path from Carl to Bruce then -1 is the output.

Sample output

```
9
```

Constraints

- $1 \leq N \leq 100\,000$
- $1 \leq K \leq 10\,000$

30% constraints

- $1 \leq N \leq 1\,500$
- $1 \leq K \leq 200$

Time limit

2 seconds.

Scoring

If the output is correctly formatted and contains the correct answer, you will score 100%. Otherwise, you will score zero.

Bitonic Tour

Author

Marco Gallotta

Introduction

Dr Bruce Merry and Dr Carl Hultquist, having recently obtained their "Dr" titles, have been banished to the workforce. It didn't take them long before they discovered boredom and started challenging one another to what has turned into a now weekly game. Carl has refused to accept that Bruce is invincible and Bruce has come to enjoy proving him wrong.

For this weeks challenge Carl has selected a classic problem – the Travelling Salesman. The challenge is to come up with a novel solution to this problem that has already been inspected by many over the years.

Task

Carl has asked that the IOI squad check his idea by solving the first part as one of their training problems, so that's why you're reading this blurb.

The part of his solution that you have to solve is to find what is called a bitonic tour with the shortest distance. A bitonic tour starts at the left-most co-ordinate, continues right only (never moving left) and then turns around and continues left only (never moving right) until the left-most point is visited. The tour must visit all co-ordinates just like the full Travelling Salesman problem, just with that added condition.

Once you have solved this part of the problem, Carl will take the best solution and perform his magic on it to get a super-efficient solution to the full Travelling Salesman problem (or so, that's what he thinks ;)). Since it is in Carl's best interests that you solve the problem he has provided you with the following formula for calculating the distance between two points a and b in 2D:

$\sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$

Example

Figure 1 illustrates the graph with points loicated at $(2, 8)$, $(3, 4)$, $(1, 0)$, $(7, 7)$ and $(4, 2)$. The directed edges indicate the bitonic tour which goes from point $3 \to 1 \to 4 \to 5 \to 2 \to 3$ in that order. The total distance travelled is $\sqrt{65} + \sqrt{26} + \sqrt{34} + \sqrt{5} + \sqrt{20} = 25.70$.

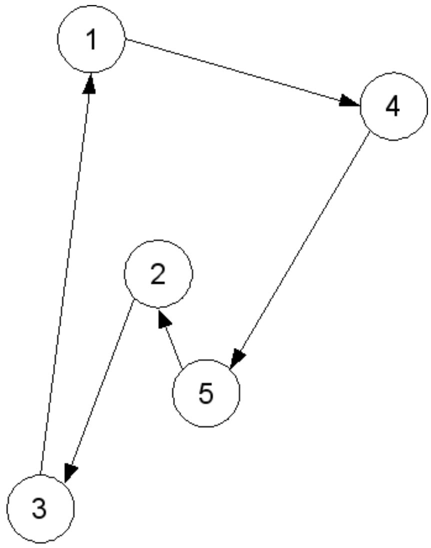

Figure 1: Example of a bitonic tour.

Input (tour.in)

The first line of the input contains a single integer, N. The next N lines each contain two space-separated integers describing a co-ordinate in the graph – x_i followed by y_i. An increasing x co-ordinate indicates travelling to the right.

Sample input

```
5
2 8
3 4
1 0
7 7
4 2
```

Output (tour.out)

The output must contain a single number representing the total distance travelled in the shortest bitonic tour as described above, rounded off to two decimal places.

Sample output

```
25.70
```

Constraints

- $1 \leq N \leq 750$
- $0 \leq x_i, y_i \leq 100000$

- $x_i \neq x_j \forall i \neq j$

30% constraints

- $1 \leq N \leq 20$

Time limit

2 seconds.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.