Seven Lakes Kickoff 2015

Tubin' in Aruba

Table of Contents

Do not write to file. Always print to standard output.

Enjoy your stay in Aruba!

(1). Chauffeur Sign

Input File: None **Output Method:** Standard out

Problem Description:
The chauffeur, who is sent by the Aruban Limo Company to pick up Mr. McGee from the airport, completely forgot to print out a sign with Mr. McGee's name on it. He quickly drives to an internet cafe, but the only free computers there run MS-DOS. The chauffeur quickly creates a text file with the edit command but cannot figure out a way to create a large version of Mr. McGee's initials, as he cannot change the text size.

Help the chauffeur create the sign for Mr. McGee with ascii art.

Input Description:
There is no input.

Output Description:
M.G. stands for "Mr. McGee".

Sample Input:
(None)

Sample Output:

```
\\\\\\\\\\\\\\\\\\\\\
\.....................\
\..@@@@.....@@@@....\
\..@@.@@...@@.@@....\
\..@@..@@.@@..@@....\
\..@@...@@@...@@....\
\..@@.........@@....\
\.....................\
\.....................\
\......  @@@@@@@@....\
\......@@@....@@@...\
\......@@...........\
\......@@...@@@@@...\
\......@@...@@.@@...\
\......@@@.....@@...\
\.......@@@@@@@@....\
\.....................\
\\\\\\\\\\\\\\\\\\\\\
```

(2). RECT_U!

Input File: rect_u.in **Output Method:** Standard out

Problem Description:
Mr. McGee wants to create the largest rectangle in existence and smash (rekt) all the other records for the largest rectangle. To do so, he has set up a coordinate grid in Aruba with (0, 0) representing the center of the island. He then places two points, forming the two diagonal corners of the rectangle.

Given these two points, find the total area of the rectangle.

Input Description:
The first line contains an integer which specifies the number of data sets to follow.
Each data set includes four integers $\mathbf{X_0}$, $\mathbf{Y_0}$, $\mathbf{X_1}$, and $\mathbf{Y_1}$ which represent a corner point of the rectangle and its opposite point, respectively. Because Mr. McGee wants to break the world record for the largest rectangle ever created, these numbers may be quite large, some exceeding one billion!

Output Description:
For each data set, output the area of the rectangle formed if the two points given in the data set form the two diagonal corners of the rectangle.

Sample Input:

```
5
100 100 -100 -100
4 3 0 0
56 78 88 90
1 1 2 2
75 -20 -45 23
```

Sample Output:

```
40000
12
384
1
5160
```

(3). Tubing in Aruba

Input File: tubing.in **Output Method:** Standard out

Problem Description:
Mr. McGee is super excited to tube in Aruba; however, he spent most of his money making rectangles, and this has left him with a meager sum to buy his tube. Each tube sold in Aruba has a strength index rating associated with it, and each "tube trail" in Aruba has a tube strength requirement. Mr. McGee can only go on a trail if his tube's strength index is greater than or equal to the requirement. Given the strength index rating of Mr. McGee's tube, determine which tube trails he can go on.

Input Description:
The first number in the input, **N**, represents the number of data sets to follow.

Each data set begins with integers **S** and **C**, representing the strength index of Mr. McGee's tube and the number of tube trails in Aruba, respectively. The following **C** lines each contains a description of a trail, in the format of "[name] (which may contain spaces!), [strength requirement] (an integer)".

Output Description:
For each data set, print out the NAMES of the trails on which Mr. McGee can go, putting one trail name per line and in the order in which they were given.

Sample Input:

```
1
10 5
Lazy River, 7
Fast Rapids, 100
Banana Split, 18
Serene Pond, 1
Jovial Happy Trail, 9
```

Sample Output:

```
Lazy River
Serene Pond
Jovial Happy Trail
```

(4).Tower Rating

Input File: sample.in **Output Method:** Standard out

Problem Description:
The premier telecommunications company in Aruba, McGee Telecoms, is running routine maintenance checks on their cell towers. Each tower is given a rating based on the following formula:

$$\frac{(yh)^{1/t}}{2u}$$

Where *y* is how old the tower is in years, *h* is the height of the tower, *t* is the transmission speed of the tower, and *u* is the maximum amount of users the tower supports.

Input Description:
An integer ***n*** that represents the number of test cases to follow. Each data set contains four integers **y**, **h**, **t**, **u** in that order.

Output Description:
The rating of the tower to the nearest hundredth.

Sample Input:

```
4
43 1 29 190
4 10 999 10
1 2 3 4
12 75 90 134
```

Sample Output:

```
108.16
5.02
2.52
72.26
```

(5). Aruban Visas

Input File: visas.in **Output Method:** Standard out

Problem Description:
Aruba is such a popular destination that every summer the Aruban government has to limit the number of tourist visas given out to visitors by gender. Given the number of male and female tourists that the Aruban government will allow into the country, and a list of tourists who want to get into the country with their genders, print out the list of male and female tourists who will be allowed into Aruba.

Input Description:
The input starts with two space separated numbers $\mathbf{N}_f$ and $\mathbf{N}_m$, which specify the number of female and male tourists that the Aruban government will allow into Aruba, respectively. Starting the next line will be a list (of unspecified length) of tourists who want to go tubing in Aruba. Every line will be in the format of "[Name] (Only the first name, no spaces in the name), [Gender] (male/female)". The tourists are allowed in on a first-come, first-serve basis.

Output Description:
First print out the names of the female tourists who are allowed in, one per line, in the order in which they were in line. Next, print out the names of the male tourists who are allowed in, one per line, in the order in which they were in line.

Sample Input:

```
2 3
Diana, female
Mark, male
Tom, male
Helen, female
Nathan, male
Sanjay, male
Erik, male
Mr.McGee, male
```

Sample Output:

```
Diana
Helen
Mark
Tom
Nathan
```

(6). Aruban Phonebooks

Input File: phonebook.in **Output Method:** Standard out

Problem Description:
The Aruban Phonebook Company needs someone to help them organize their phonebooks. Being the helpful person that he is, Mr. McGee has volunteered to help the company organize the data so that it can be put into a phonebook. Mr. McGee is given a list of names, and he is asked to put them into ascending alphabetical order. Mr. McGee is told to first sort ascending (from A to Z) based on the last names, and then to sort ascending based on the first names only if the last names of two people are the same. There will be no people with the same last names and first names.

Input Description:
The input starts with a number, **N**, which specifies the number of lines to follow.

Each of the following lines contain a name entry in the format of "[firstName] [lastName]". Both the first and last names will not contain any spaces.

Output Description:
Print out the list of names in ascending order as described by the problem description, one name per line in the format "[lastName], [firstName]".

Sample Input:

```
6
Mr. McGee
Paul Stroud
Tres Brenan
Colby Brown
Miguel Obregon
Jennifer Stroud
```

Sample Output:

```
Brenan, Tres
Brown, Colby
McGee, Mr.
Obregon, Miguel
Stroud, Jennifer
Stroud, Paul
```

(7). Sweatin' Bullets

Input File: bullets.in **Output Method:** Standard out

Problem Description:
Little Jimmy thought Mr. McGee was bluffing about AP exam prep, so he spent all weekend tubing in Aruba. Unfortunately, Mr. McGee is never one to bluff. Jimmy sees the dreaded folders between the computers upon entering Mr. McGee's room, and he begins frantically flipping through his AP Computer Science review book. However, Jimmy accidentally tears the binding, causing the whole book to fall apart. Luckily, Jimmy remembered the merge sort algorithm, and began to get to work to try and repair the book before the bell rang. The merge sort algorithm is a neat little trick that sorts any set of elements and always take the same amount of operations, which is calculated with the equation:

$$s \times log_2 s$$

where **s** is the number of pages in the book. Jimmy can only perform one of these operations every second. Given the amount of time in seconds that Jimmy has left and the number of pages in his AP Computer Science book, determine whether Jimmy will have his book together fast enough to find the number of bits in a int, or if he'll be sweatin' bullets, flipping through the jumbled mess.

Input Description:
The first line contains an integer, **N**, which specifies the number of data sets to follow.

Each data set contains 2 integers, **s** and **t**, where **s** is the number of pages in Jimmy's book and **t** is the amount of time in seconds Jimmy has before the bell rings and his fate is sealed.

Output Description:
For each data set, output "Sweatin' bullets" if Jimmy is unable to sort his packet before the bell rings or "It's 32!" if Jimmy is able to successfully sort his packet and thus, find the number of bits in an int.

Sample Input:

```
5
1 20
12 6
9 9
200 2000
963 10
```

Sample Output:

```
It's 32!
Sweatin' bullets
Sweatin' bullets
It's 32!
Sweatin' bullets
```

(8).Solid Mcgee

Input File: solid.in **Output Method:** Standard out

Problem Description:
Mr. McGee's visit to Aruba was not just to have a great time tubing; he was also there on a secret mission to infiltrate the base of Liquid McGee, his evil clone, to prevent him from taking control of "The System". However, before Liquid can be vanquished, Mr. McGee (a.k.a. Solid McGee) has to sneak through the rooms of Liquid's base, which, although do not have any walls, are filled with nasty guards who will raise an alarm upon seeing Solid. The good thing is, because all the guards are controlled by nanomachines, Solid McGee knows that they will **all be looking west when he enters the room**, that they can only look directly west, north, east, and south, and that the guards will rotate clockwise once (west to north, north to east, east to south, south to west) instantly every time Solid takes one step. Furthermore, the guards have an extremely narrow field of vision (one row/one column on the map) and always stay in the same spot. Finally, as Solid McGee has been nearly incapacitated from the effects of the virus MCGDIE, he can only take one step to the west, north, east, or south at a time. If Solid Mcgee is standing in a spot in the line of vision of a guard, he is seen, unless he has reached the exit. Provided with a map detailing the positions of the guards, Solid McGee wants to know if he can sneak past these rooms without being noticed by the guards, or if he has to resort to neutralizing some guards.

Input Description:
The first number of the input, **N**, represents the number of data sets to follow.

Each data set begins with **R** (1 <= R <=15) and **C** (1 <= C <= 15) which represent the numbers of rows and columns of the map which follows. In each map, empty spaces are represented by the "**.**" character while Solid McGee's entry position is represented by the "M" character, guards are represented by the "G" character, and the exit is represented by the "E" character.

Remember, every guard can only look west, north, east, or south in a one-cell wide straight line and rotates once clockwise for every step that Solid takes **and all are looking west at the time Solid McGee enters the room.**

Output Description:
For each data set, output "SNEAKY BEAKY" if Solid is able to sneak past the guards and get to the exit without being detected and output "RESORT TO VIOLENCE" if Solid is not able to sneak past the guards.

Sample Input:

```
2
5 5
M....
G....
.G...
G....
.G..E
2 2
M.
EG
```

Sample Output:

```
RESORT TO VIOLENCE
SNEAKY BEAKY
```

(9). Tourist Survey

Input File: survey.in **Output Method:** Standard out

Problem Description:
The Aruban government offered a free tube for anyone who was willing to help them in analyzing the data from their most recent tourist survey. Mr. McGee, being the helpful, magnanimous person that he is, agreed to write a program to analyze the data from the tourist survey. Every person who answers the survey is required to provide his/her name, gender, age, and whether or not he/she plans to go tubing while visiting Aruba. Given this information, Mr. McGee's goal is to generate some useful statistics on the tourist population visiting the beautiful country of Aruba.

Input Description:
There will be an unknown number of lines in the input. Each line will include the name of the respondent in the format of "Last, First", followed by a space and the gender of the respondent as either "Male" or "Female", followed by a space and the age of the vacationer as an integer value, followed by a final space and the answer "No" or "Yes" detailing whether the respondent plans to go tubing in Aruba.

Output Description:
Output the following statistics on separate lines in the following order in the format of "[StatisticName]: [Value]" for example "`Number of Respondents : 42`". The Aruban government wants to find out: Number of Respondents, Percentage of Male Tourists (rounded to 3 decimal places), Percentage of Female Tourists (rounded to 3 decimal places), Average Age of Tourists (rounded to 3 decimal places), and Percentage of Tourists who plan to go tubing (rounded to 3 decimal places). After this, print out "`Have a great trip, everyone!`" on a new line.

Sample Input:

```
Gosling, James Male 60 Yes
Stroustrup, Bjarne Male 64 No
McGee, Harold Male 20 Yes
Hopper, Grace Female 108 Yes
```

Sample Output:

```
Number of Respondents : 4
Percentage of Male Tourists : 75.000
Percentage of Female Tourists : 25.000
Average Age of Tourists : 63.000
Percentage of who plan to go tubing : 75.000
Have a great trip, everyone!
```

(10). The Real McGee

Input File: real.in **Output Method:** Standard out

Problem Description:
"The real McGee" is an idiom and metaphor used in much of the Aruban-speaking world to mean "the real thing" or "the genuine article", *e.g.*, "he's the real McGee"(this phrase is completely unrelated to The Real McCoy). Why is this important? Because McGee's hotel has an issue with fraudulent check-ins, as Aruba is such a popular destination that its hotels become overbooked and people go in pretending to be other people. In order to stop people from falsely impersonating real guests, the hotel has implemented a 3-question system where the guest submits 3 questions and their answers to the questions upon booking their room, then tells the correct answers upon arriving. However, because the hotel didn't use Java, its software is vulnerable and can be hacked into at any time. As such, the hotel placed semicolons after every answer in order to make it harder for hackers to steal the answers. Your task is to determine whether the person checking in is 'The Real McGee'. Be careful, as the person checking in was in a rush and may have made capitalization errors, and these do not count against them.

Input Description:
The first line will contain an integer containing the number of data sets to follow.
Each data set will contain 3 lines of questions, and then 3 lines of the correct, 'encrypted' answers that respectively correspond to each question. Finally, the last 3 lines of the data set will contain the answers that the person checking in submitted.

Output Description:
For each data set, you must output 4 lines:
The first line will contain the first question and the first answer of the person checking in, separated by a space, a colon, and another space.
The second and third lines shall be in the same format as the first line, except with the second and third questions and answers.
The last line will contain "`The Real McGee`" if the person answered their questions correctly and "`shoulda used Java`" if he/she did not.

Sample Input:

```
2
What is the name of your first pet?
Who wrote Java?
Does the compiler care?
Duke;
James Gosling;
No;
duke
JaMes Gosling
no
Is C++ better than Java?
Aren't lambdas great?
What should you do when you get a null pointer exception?
```

```
In Stroustrup's Dreams;
YAAAASSSSSS;
Right click and delete project;
Of course it’s better
Yes
RIGht click and delete project
```

Sample Output:

```
What is the name of your first pet? : duke
Who wrote Java? : JaMes Gosling
Does the compiler care? : no
The Real McGee
Is C++ better than Java? : Of course it’s better
Aren’t lambdas great? : yes
What should you do when you get a null pointer exception? : RIGht click
and delete project
shoulda used Java
```

(11). Who Wants to Tube?

Input File: whoWants.in **Output Method:** Standard out

Problem Description:
Every day, the Aruban schoolteachers like to play a game with their students called "Who Wants to Tube?" In this game, the teacher asks the children, "Who wants to tube?", and the children are expected to scream "WE DO". This repeats until the children lose their voices, rendering them incapable of disrupting the teacher during instruction. Given the popularity of this game, the Aruban government has again interrupted Mr. McGee's vacation by asking him to create a simulation of this game. Help Mr. McGee write a program which will mimic this game.

Input Description:
The input will consist of an unspecified amount of lines which all say "`Who wants to tube?`".

Output Description:
For each "`Who wants to tube?`" question, print out "`WE DO!`" on a new line. Once the teacher stops asking questions, it can be inferred that the children have gone quiet. When this happens, print "`Silence is a friend that never betrays.`"

Sample Input:

```
Who wants to tube?
Who wants to tube?
Who wants to tube?
Who wants to tube?
```

Sample Output:

```
WE DO!
WE DO!
WE DO!
WE DO!
Silence is a friend that never betrays.
```

(12). The Compiler Don't Care

Input File: compiler.in **Output Method:** Standard out

Problem Description:
Mr. McGee forgot to take the compiler with him on his visit to Aruba, and so now he can't run his code. How's a man to enjoy his vacation without some healthy Java? However, you can help him! As you may already know, the compiler doesn't care; it only does what you tell it to do. If you tell it to blow up the computer, then it will faithfully follow your mandate. If you tell it to print out one million lines, then it will do that too! For this problem you're going to take the role of the compiler and do what you're told to do. The syntax is as follows:

Syntax rule #1:
Syntax: print **A B**
Rule: **A** is an integer representing the number of times to print out **B**, which will never contain any spaces. You then start a new line.
Syntax rule #2:
Syntax: println **A B**
Rule: **A** is an integer representing the number of lines to print string **B**, which will never contain any spaces.
Syntax rule #3:
Syntax: selfdestruct
Rule: print out "DONE" and then start a new line. After this statement, ignore any other commands given by the data set.
Syntax Rule #4:
Syntax: add **A B**
Rule: **A** and **B** are integers; print out the sum of **A** and **B** then start a new line.
Syntax Rule #5:
Syntax: Care
Rule: Print out "No!" then start a new line.

Input Description:
The first line will contain an integer **z** specifying the number of data sets to follow.

Each data set will consist of the following:
A line that contains an integer **i** which specifies how many lines the data set has
i lines of code to be interpreted

Output Description:
For every data set, follow the instructions on each line according to the syntax rules until you encounter a self-destruct command or until the current data set ends

Sample Input:

```
4
3
print 10 SPAM
selfdestruct
add 2 1
4
println 3 HELLO
add 3 4
Care
Care
```

Sample Output:

```
SPAMSPAMSPAMSPAMSPAMSPAMSPAMSPAMSPAMSPAM
DONE
HELLO
HELLO
HELLO
7
No!
No!
```

(13). McGee Telecoms

Input File: mcgeeTele.in **Output Method:** Standard out

Problem Description:
In the beautiful island of Aruba, the telecommunications mogul, McGee Telecoms, has been struck by economic downturn, forcing the company to replan their coverage area. Prior to this catastrophic event, McGee Telecoms had the capital to run several telephone lines throughout the city. However, now they must use only one line starting from the McGee Telecoms headquarters and going throughout the neighborhoods of Aruba. You, having recently been promoted to COO, want to get an idea about how much money the company can earn using this one cable system. Your task is to write a program that finds the path that gives the most profit.

```
          7

        3   8

      8   1   0

    2   7   4   4

  4   5   2   6   5
```

The neighborhoods of Aruba are laid out in a triangular pattern; each number in this pattern denotes how many millions of dollars a neighborhood spends on McGee Telecom's services. The headquarters of McGee Telecoms (where the cable begins) are located in the neighborhood at the top of the triangle. The cable can only move forwards(towards the base of the triangle) relative to the headquarters and either left or right, and it will end at any of the neighborhoods at the base of the triangle.

Input Description:

The first line contains R (1 <= R <= 1000), the number of rows in the triangle. Each subsequent line contains the integers for that particular row of the triangle. All the supplied integers are non-negative and no larger than 100.

Output Description:

A single line containing the largest amount of money that can obtained using one cable.

Sample Input:

```
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

Sample Output:

```
30
```

(14). Palindromic Tubes

Input File: palinTubes.in **Output Method:** Standard out

Problem Description:
Mr. McGee is planning to go tubing on the Palindromic River in Aruba, and he needs your help to decide which tubes that he can buy! Although there are many tubes for sale in Aruba, the staff at the Palindromic River only allows people who have tubes which have names which can be expressed as the sum of two palindromes, with lengths of at least two, to come and tube on the river. A palindrome is a word that is the same when it is written backwards. Given the list of tube names which Mr. McGee finds in a tube catalogue, help Mr. McGee determine which tubes he can buy and take to the Palindromic River.

Input Description:
The first line will contain an integer, **N**, representing the number of data sets to follow.

Each data set consists of a string representing the name of a tube; the name will only contain lowercase characters.

Output Description:
Print out the names of the tubes which Mr. McGee can take on his trip to Aruba, one per line and in the order in which they were given.

Sample Input:

```
4
kayakracecar
besttubes
wowrotator
arubanvaluetubes
```

Sample Output:

```
kayakracecar
wowrotator
```

(15). The Aruban Game

Input File: game.in **Output Method:** Standard out

Problem Description:
The Aruban government is holding a tournament where tourists compete in the Aruban Game to win an all-expenses paid trip to the neighboring island of Nabura. Mr. McGee would love to take this trip, and he needs your help to write a program which will help win all the games in the tournament. The Aruban game is just a derivative of the classic rock-paper-scissors game. Instead of rock, there is a river; instead of paper, there are tubes, and instead of scissors, there are scythes. The game operates based on these rules:

- The river beats the scythe because water rusts metal
- The scythe beats tubes because it can cut into tubes
- The tube beats the river because it can float on water

Given what Mr. McGee's opponent chooses, write a program which tells Mr. McGee what to choose so that he can win.

Input Description:
The input starts with an integer, **N**, detailing the number of lines to follow. Each of the next **N** lines contains the choice of Mr. McGee's opponent - either "river", "scythe", or "tube".

Output Description:
Print out the choice that Mr. McGee needs to make so that he can win against his opponent.

Sample Input:

```
4
scythe
tube
river
scythe
```

Sample Output:

```
river
scythe
tube
river
```

(16). Hacking

Input File: hacking.in **Output Method:** Standard out

Problem Description:
A pr0 hax0r has broken into the Aruban government computers and changed the contents memory addresses of computers, and the Aruban government has called on Mr. McGee to help them get their computers working again. Mr. McGee is given a file containing what each memory address, starting from address zero, is supposed to contain, and he is expected to change the memory of each computer to match what is given in the file. Given the file and the contents of the current computer's memory address, help Mr. McGee by writing a program which tells him which memory addresses need to be changed so that the computer's memory will match what is given in the file.

Disclaimer: The Seven Lakes Computer Science Club does not promote hacking in any way, shape, or form.

Input Description:
The first number, **N**, will indicate how many memory addresses are in the file and in the computer.

The next **N** lines will detail the data from the file which Mr. McGee is give about the memory address, in the format of "[address name] [value]" - the address name will be an integer while the value will be a hexadecimal number. This data will be in ascending order based on the the address name and will always start at address zero (from 0 to **N**.

After this, the next **N** lines will detail the current memory on the computer which Mr.McGee is trying to fix in the same format as the contents of the file.

Output Description:
Print out the address names which contain incongruous values separated by commas and in ascending order so that Mr.McGee will know which addresses he needs to change.

Sample Input:

```
3
0 21a
1 197
2 135
0 21a
1 aaa
2 32a
```

Sample Output:

```
1,2
```

(17). Safe Crack

Input File: safe.in **Output Method:** Standard out

Problem Description:
After successfully sneaking past or neutralizing the guards of Liquid Mcgee, Mr. Mcgee was met with a nasty surprise: "The System" is protected by a complicated combination lock! This device requires any list of integers that have a sum of **n**, but each of these numbers cannot contain a digit greater than 1. Since Mr. Mcgee doesn't want to spend an eternity finding a solution, write him a program to find the minimum number of such integers required to have a sum of **n**.

Input Description:

The first number of the input, **N**, represents the number of data sets to follow.

Each data set consists of a string representing an integer **n** no greater than 10^{20}.

Output Description:

Output the minimum number of integers that can have a sum of **n**, if each integer cannot contain a digit greater than 1.

Sample Input:

```
2
9
100000000000000000
```

Sample Output:

```
9
1
```

(18). RECT_U_2!

Input File: rect_u_2.in **Output Method:** Standard out

Problem Description:
After spending so much time creating the largest rectangle in the world, Mr. McGee has become somewhat obsessed with rectangles. He feels the urge to place them everywhere in Aruba, and this has drawn the attention of the Aruban government, which believes that the random rectangles all over the place is detrimental to the tourist industry. However, before the bureaucrats can allocate the funds needed to remove all the rectangles, they must assess the damage so that their payment to the damage-control company (which the president holds a 40% stake in) does not seem too large. They will use the same coordinate system which Mr. McGee used to measure the area of his rectangle to try and assess the damage caused by Mr. McGee's rectangle rampage. Given the coordinates of the corners of at most 500 rectangles, calculate the area occupied by at least one of them (eg. the total area covered by all the rectangles, without any overlap).

Input Description:
The first line contains an integer, **n** (1 <= **n** <=500), which specifies the number of lines to follow. Each line contains four integers $\mathbf{X_0}$, $\mathbf{Y_0}$, $\mathbf{X_1}$, and $\mathbf{Y_1}$, which specify a corner point of a rectangle and its opposite corner point, respectively.

Output Description:
Given the data which specifies the locations of the rectangles, print out the total area occupied by at least one rectangle (eg. the total area covered by all the rectangles, without any overlap).

Sample Input:

```
2
2
0 0 10 10
2 2 4 -2
1
-10 -10 10 10
```

Sample Output:

```
104
400
```