South African
Computer Olympiad Training
DAY 1

Overview

Author	Francois	Harry	Graham	Heinrich
Problem	**Math Parser**	**Row**	**Wandering Rabbits**	**Nine Letter Words**
Program name	`parse.exe`	`row.exe`	`bunny.exe`	`word9.exe`
Source name	`parse.pas` `parse.jav` `parse.cpp`	`row.pas` `row.jav` `row.cpp`	`bunny.pas` `bunny.jav` `bunny.cpp`	`word9.pas` `word9.jav` `word9.cpp`
Input file	`parse.in`	`row.in`	`bunny.in`	`word9.in`
Output files(10)	`parse.out`	`row.out`	`bunny.out`	`word9.out`
Time limit	2 seconds	2 seconds	2 seconds	2 seconds
Num. of tests	10	10	10	10
Points per test	10	10	10	10
Total points	**100**	**100**	**100**	**100**

The maximum total score for Day1 is 300 points.

Math Parser

Author:

Description:

You are required to write a program that will read an equation from a text file and produce the answer.

The equation will consist of the following: Digits and Operators. The Operators will consist of the following: (In order of importance)

1. ( ) : Normal brackets e.g. 2(3+2)2 = 2*(3+2)*2 = 2*5*2 = 20 (If there is no operator between the bracket and the number then you can assume that it's multiplication.)
2. ^ : Exponents e.g. 2^3^2 = 8^2 = 64
    i. (Process the equation from left to right if the operator is the same. So in this example you will not get 2^3^2 = 2^9 = 512)
3. % : Mod e.g. 5%2 = 1
    i. (Will never get 0 as a parameter e.g. 5%0)
4. * : Multiply e.g. 2*3 = 6
5. / ; Division e.g. 6/3 = 2
    i. (Will never get 0 as a parameter e.g. 6/0
    ii. The first parameter will always be a multiple of the second.)
6. + : Addition e.g. 2+3 = 5
7. - : Subtraction e.g. 3-2 = 1

(4 and 5); (6 and 7) are equal and should be processed which ever occurs first.

If your output is correct you will score 10 marks per test case.

Input :

The first line of *parse.in* will contain the equation. It will not exceed 255 characters in length.

Output :

Output a single integer in *parse.out* which is the answer for the equation.

Example:

    parse.in
    (3*5)^2

    parse.out
    225

Constraints:
Time Limit: 2 seconds
Equation max length: 255
Integer range: 32 Bit Integer
Test cases: 10
All intermediate results will be non-negative integers.

Row

Author: Harry Wiggins

Description

Alice wrote the number 1 to n in a certain order. She added two adjacent numbers and wrote sum beneath the initial row. She continue doing it, until one number is left. Giving this number to Bob, she wants him to find the row.

4 2 1 3
6 3 4
9 7
16

Input data

The first line would contain n (1•n •12).
The second line would contain the number Alice gives Bob.

Output data

The output should contain the numbers 1 to n in the order to achieve this.

Example

row.in
4
16

row.out
4 2 1 3

Constraints

Time limit: 2 seconds/test

Wandering Rabbits

Author: Graham Poulter

Introduction

A colony of rabbits had their warren washed out in a flood, and they are looking for a new home. You are given a muddled-up rabbit map of the property market, which consists of all the pairs of rooms which have a tunnel between them.

Tasks

- Your first task is to figure out how many warrens there are in the map.
- The rabbits have a large extended family, so your second task is to find the size of the largest warren in the map in which an Eulerian tour is possible (sometimes they get lost and like to know they can get back to where they started just by walking randomly and not backtracking). Ties don't matter – the largest size is still the same.
- Your third task is to show the rabbits around their new home by taking them on an Eulerian tour of a warren in which an Eulerian tour is possible and with the same size as the largest warren in which an Eulerian tour is possible (this is why ties don't matter).

Definitions & Constraints

- A warren is a set of rooms, some of which are connected by tunnels, such that from each room in the warren it is possible to find at least one path to each of the other rooms in the warren. A single unconnected room counts as a warren.
- Starting in one warren you cannot reach a room in another warren.
- A room cannot be part of more than one warren.
- The size of a warren is the number of rooms which it contains.
- Only in some of the warrens is an Eulerian tour possible.
- To do an Eulerian tour means that, starting in any room in the warren, you travel through every tunnel in the warren exactly once and end up back to where you started.
- Each tunnel may be traversed in either direction, but may only be traversed once during the Eulerian tour.
- The number of rooms in the map is between 1 and 5000.
- The number of tunnels in the map is between 1 and 60000.
- The number of warrens in the map is between 1 and the number of rooms.

Input Format (bunny.in)

Line 1: Two integers, R and T. R is the number of rooms in the map. T is the number of tunnels in the map

Lines 2..T+1: Two integers per line, A and B, each pair representing the end-points of a tunnel in the map . $1 \bullet A,B \bullet R$ and $A \bullet B$. The tunnels are listed in no particular order (I shuffled them well), can only be traversed once in either direction during the Eulerian walk, and may appear multiple times in the list for multiple tunnels between two rooms.

Output Format (bunny.out)

Line 1: Four integers, W, R, T and S. W is the number of warrens in the map. R is the number of rooms in the largest warren in which an Eulerian tour is possible. T is the number of tunnels in that warren, which is also the length of

the Eulerian tour of that warren because the Eulerian tour traverses each tunnel precisely once. S is the room in which you choose to start the tour in. The only condition on S is that it must be in an Eulerian warren which has R rooms.

Lines 2..S+1: One integer per line representing each successive room being visited in the tour (beginning with the first room visited after S). Each room must be reachable from the room on the previous line. The last room must the be room S from which the tour started. By the end of the tour all of the tunnels in the warren must have been traversed precisely once, although that one traversal may be in either direction.

Scoring

There are 10 cases, a time limit of 1 second on each case, and a 16Mb memory limit. Your score out of 10 on each case is the sum of your scores in these three sections:

A) You get 2 points if you have the right number of warrens.
B) You get 3 points if you have the right number of rooms in the largest warren.
C) You get 5 points, but only if your answer for (B) is correct, *and* you do a correct Eulerian walk of a warren which has the same number of rooms as your answer in (B). (This flexibility is why ties for the largest Eulerian warren don't matter).

Example Input
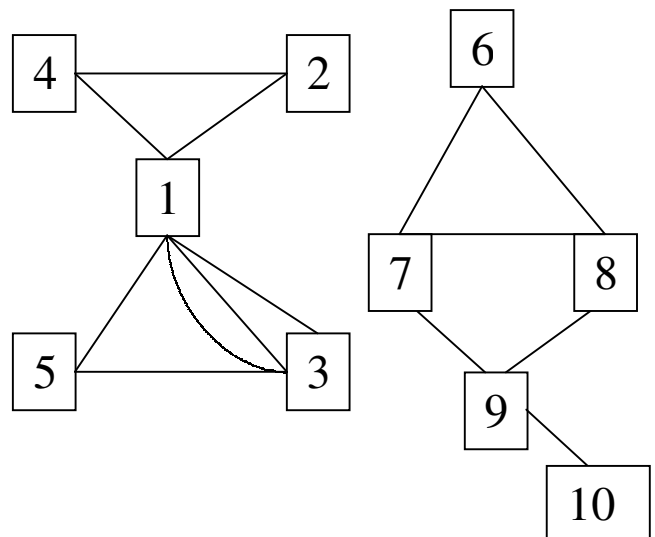
```
10
14
6 7
1 2
4 2
1 5
3 1
5 3
3 1
8 9
7 9
1 3
7 8
10 9
6 8
4 1
```

Line 1 = 10 rooms
Line 2 = 14 tunnesl
Lines 3 to 17 = (A,B)
end-points of each tunnel

Example Output

```
2
5
8
4
2
1
5
3
1
3
1
4
```

Line 1 = 2 warrens
Line 2 = largest Eulerian warren has 5 rooms
Line 3 = 8 tunnels in largest warren.
Line 4 = starting the tour in room 4.
Lines 5 to 13 = rooms in the order in which they
are traversed. Path here goes (4) $\rightarrow$ 2 $\rightarrow$ 1 $\rightarrow$ 5
$\rightarrow$ 3 $\rightarrow$ 1 $\rightarrow$ 3 $\rightarrow$ 1 $\rightarrow$ 4   (The (4) is the
starting room)

Nine Letter Words

AUTHOR: Heinrich du Toit

Description:
You have to write a program to sort 9 columns of letters of
length N so that they make N 9 letter words.

INPUT: word9.in
First line contains 1 integer N: The number of rows (words)
there are.
The next N lines will each contain 9 letters. (A..Z)

OUTPUT: word9.out
The output consist of N rows each containing a valid word.
Order of the words is not important.
If multiple solutions exist you only have to give 1 valid
solution.
If no solution exists output:
'none' (small letter)

Then the Dictionary follows: (second part of input file)
The dictionary will only contain 9 letter words in capital
letters.
The first line will contain 1 integer W: The number of 9
letter words (In captial letters).
The next W lines each contains one 9 letter word.
The dictionary will not contain any duplicates.

Example
Input:
```
4
PUAVILFGN
BNIERTEIY
UERUTIALL
CRCIOUSUE
5
PRIVILEGE
UNCERTAIN
BEAUTIFUL
CURIOUSLY
ATTICLIKE
```

Output:
```
PRIVILAGE
UNCERTAIN
BEAUTIFUL
CURIOUSLY
```

Contraints
(2 <= N <= 8)
(1 <= W <= 10000)
A column may not contain the same letter twice.
Max time: 2 seconds

Scoring
A correct solution will score 100%
An invalid solution will score 0%
If your answer is partially correct:
You will get (50/N)% for every unique word that is correct
on its own.
If all your words fit you will get on top of your 50% an
additional
(30/N)% for every word that fit in with the previous words.
(Not for
the first one)

Invalid Solution
If you have more or less than N lines.
If your words have non-capital letters or other characters.
If your words aren' t of length 9.
If you give a word not in the dictionary.
If you did not produce an output file.
(More conditions may be added)