ACM International Collegiate Programming Contest 2017

Fourth UPDS Local Contest
Jul 15th, 2017

This problem set contains 6 problems;

Problemtesters:

Grover A. Irusta Miranda

UVa Online Judge

12918 Lucky Thief

A very lucky thief found n keys on a street with m houses. He knows that each key opens exactly one door so he wants to know which door is, but he also wants to do the less number of tries in order to avoid the security systems.

Input

The first line contains a single integer, T ($T \leq 100000$) indicating the number of test cases. The following T lines contain two integers n and m, $1 \leq n \leq m \leq 100000$.

Output

Print the minimum number of tries that the thief must perform in order to know what key opens each door with an end of line.

Sample Input

```
3
4 6
1 2
500 900
```

Sample Output

```
14
1
324750
```

UVa Online Judge

10852 Less Prime

Let n be an integer, $100 \leq n \leq 10000$, find the prime number x, $x \leq n$, so that $n - p * x$ is maximum, where p is an integer such that $p * x \leq n < (p + 1) * x$.

Input

The first line of the input contains an integer, M, indicating the number of test cases. For each test case, there is a line with a number N, $100 \leq N \leq 10000$.

Output

For each test case, the output should consist of one line showing the prime number that verifies the condition above.

Sample Input

```
5
4399
614
8201
101
7048
```

Sample Output

```
2203
311
4111
53
3527
```

UVa Online Judge

294 Divisors

Mathematicians love all sorts of odd properties of numbers. For instance, they consider 945 to be an interesting number, since it is the first odd number for which the sum of its divisors is larger than the number itself.

To help them search for interesting numbers, you are to write a program that scans a range of numbers and determines the number that has the largest number of divisors in the range. Unfortunately, the size of the numbers, and the size of the range is such that a too simple-minded approach may take too much time to run. So make sure that your algorithm is clever enough to cope with the largest possible range in just a few seconds.

Input

The first line of input specifies the number N of ranges, and each of the N following lines contains a range, consisting of a lower bound L and an upper bound U, where L and U are included in the range. L and U are chosen such that $1 \leq L \leq U \leq 1000000000$ and $0 \leq U - L \leq 10000$.

Output

For each range, find the number P which has the largest number of divisors (if several numbers tie for first place, select the lowest), and the number of positive divisors D of P (where P is included as a divisor). Print the text '`Between` L `and` H`,` P `has a maximum of` D `divisors.`', where L, H, P, and D are the numbers as defined above.

Sample Input

```
3
1 10
1000 1000
999999900 1000000000
```

Sample Output

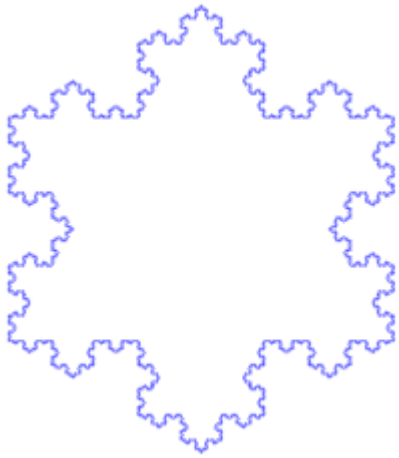
```
Between 1 and 10, 6 has a maximum of 4 divisors.
Between 1000 and 1000, 1000 has a maximum of 16 divisors.
Between 999999900 and 1000000000, 999999924 has a maximum of 192 divisors.
```

URI Online Judge | 1760

Koch Snowflake

By Michele Selivon, URI Brazil

Timelimit: 1



Lapland is a quiet and very cold place. There is not much to do there after Christmas (where the elves work diligently in Santa's toy factory). The marasmus has made the elf Tod review research about the only thing that could be seen in Lapland: Snow.

In their studies, Tod found very interesting things about the snowflakes. How did incessant search the sites for information about snowflakes, eventually finding links that talk about a theory called Koch Snowflake.

Tod found the very interesting theory because the Koch snowflake is a fractal that is obtained from an equilateral triangle. Then divide each of its sides into three equal parts and added, from each middle section, a new equilateral triangle of side equal to 1/3 of the measure on the side of the initial triangle.

At each iteration fractal perimeter and after n iterations increases, it tends to infinity but remains smaller area than the area of the circle surrounding the original triangle. Thus, an infinitely long line is surrounded by a finite area.

Based on this information and knowing that the area of a triangle is equal to $l^2 \sqrt{3} /4$ (where *l* is equal to measure the length of a side of the equilateral triangle) your task is to help Tod find the area of a Koch snowflake based on the measurement of the length of side of the equilateral triangle given.

Input

The input has several test cases and consists of an integer **l** (1mm ≤ *l* ≤ 1000mm) that represents a measure of the length of one side of the equilateral triangle. The end of input is determined by EOF.

Output

The output should show the value Koch Snowflake area with two decimal places.

Input Sample	Output Sample
2	2.77
3	6.24
4	11.09

Contest de Natal 2014

URI Online Judge | 1401

Generating Fast, Sorted Permutation

By Shahriar Manzoor, SEU Bangladesh

Timelimit: 2

Generating permutation has always been an important problem in computer science. In this problem you will have to generate the permutation of a given string in ascending order. Remember that your algorithm must be efficient.

Input

The first line of the input contains an integer **n**, which indicates how many strings to follow. The next **n** lines contain **n** strings. Strings will only contain alphanumeric and never contain any space. The maximum length of the string is 10.

Output

For each input string print all the permutations possible in ascending order. Note that the strings should be treated, as case sensitive strings and no permutation should be repeated. A blank line should follow each output set.

Sample Input	Sample Output
3 ab abc bca	ab ba abc acb bac bca cab cba abc acb bac bca cab cba

URI Online Judge | 1068

Parenthesis Balance I

By Neilor Tonin, URI Brazil

Timelimit: 1

Considering an expression with parenthesis, print a message informing if the among of parenthesis is correct or incorrect, without considering the rest of the expression. Example:

`a+(b*c)-2-a` is correct
`(a+b*(2-c)-2+a)*2` is correct
when

`(a*b-(2+c)` is incorrect
`2*(3-a))` is incorrect
`)3+b*(2-c)(` is incorrect

Resuming, all closing parenthesis must have an open parenthesis and it's not possible a closing parenthesis without a previous open parenthesis, and the quantity of closing and open parenthesis must be the same.

Input

The input file contains **N** expressions (1 <= **N** <= 10000), each one with up to 1000 characters.

Output

The output must be **correct** or **incorrect** for each test case according with above rules.

Input Sample	Output Sample
a+(b*c)-2-a	correct
(a+b*(2-c)-2+a)*2	correct
(a*b-(2+c)	incorrect
2*(3-a))	incorrect
)3+b*(2-c)(	incorrect