American educational system has provided a very favorable learning environment and good resources for our children. However, to better use those resources, we have to spend time and effort to do research on all available resources, select a proper one or more project for our students based on their talents. Over the past years, ACES has been working so hard to reach our goals to help our children to exploit their own potential and to void missing any opportunity.

After we presented Mathematics Olympiad (USAMO), physical Olympiad (USAPO), here we like to introduce USACO (USA Computing Olympiad), a programming competition for high school students. We believe that this is another the most worthwhile competition that our students can engage in. Here are three reasons:

1. **First**, the students must learn a programming language. Language can be: C, C++, Java or Python. We all know that in the past few years, Computer Science has transformed from a specialized field of study to a subject that everyone should have some understanding of. It quite literally applicable to every other field. If our children start to learn a programming language since middle school even elementary school, it’s means we are giving them a lifetime skills and add a heavy weight in their future competitive environment. In fact, a requirement to participate USACO is not too high, but be fluent.

2. **OSACO is divided into three divisions**: Bronze, Silver, and Gold. Students are automatically placed into Bronze. It focuses on algorithmic and clever thinking, so knowledge of data structures and algorithms is a requirement.
   
   What we like to share with everyone is a very interesting discovery. Some OSACO questions are very similar with Google, Facebook and some of Wall Street's interview problems, both in form and contents. Actually, those interview questions are far simpler than difficult programming competition questions.

   We all know that many college student had spent so much time to work on those interview questions when they try to find summer inter job or the last year job hunting. Some of them lost opportunities because they are not familiar with those types of questions. By comparing USACO questions, you will find those interview questions will become quite easy as long as you finished bronze level, just like if you do well on MathCounts, then you will be automatic be at the SAT Math perfect score level. **Conclusion**: USACO helps future employment.

3. **There are a total of 4-6 contents during the year**, each of which takes approximately 3-4 hours and is composed of 3-4 problems. Each of the problems has a number of test cases as well as a time limit. Most people tend to code in Java/C++ because of their convenience/speed. You can take the test at home, though there is a strict honor policy put in place.

   Each contest has its own cutoff for promotion to the next division, so when you do well, you can be promoted to the next division. Participants that perform well (especially Gold) are
typically admitted to strong colleges who all hold these competitions with high regard (e.g., MIT, Harvard, etc.). Typically, achieving Gold division distinction requires a great deal of effort, but entering the Silver division is a very doable achievement, even within a year. Conclusion: USACO helps college admission.

We now know the USACO is a worthwhile competition. Next question is what is the rules and test format? Each contest has typically 3-4 problems to which you will submit solution in C, C++, Pascal, Java, or Python. Your score for each problem depends on the number of input cases your program can solve within the time limit (for most contests, 2 seconds per input case for C, C++, and Pascal, and 4 seconds per input case for Java and Python). So clever algorithm and data structure will necessary to solve all test cases correctly. Your program must be submitted within the specified period of time without compilation errors and run-time errors. Then the program will be run against a number of judging test cases. Contest winner (650 out of 1000 point) will be promoted to the next division. The difficulty level of each division increases progressively. We attached a February 2015 tournament week Bronze tournament title in the final article. You can feel the questions and level of difficulties.

We all know any Olympiads (total 5 of them: Math, Physics, Biology, Chemistry and Computer Science) require a great deal of effort and significant, regular practice to do well on. ACES will start to offer classes and club to help students in this area. Our plan is: First of all, to learn a programming language. We chose the Java or Python as an introductory language (our summer Java course is going on now). To participate the competition, students also need to know date input and output as well as basic data structures or algorithms. With this foundation, they can start competition training. Our competition training will use past competition questions and questions from other resource. Accordingly, we have designed a roadmap for the following courses:

Actually, our children are quite lucky because a majority of parents are working with the software and IT-related field. Let us work together to help our kids to learn a programming language at their early age, build up confidence for future development and more opportunities.
Farmer John has purchased a subscription to Good Hooveskeeping magazine for his cows, so they have plenty of material to read while waiting around in the barn during milking sessions. Unfortunately, the latest issue contains a rather inappropriate article on how to cook the perfect steak, which FJ would rather his cows not see (clearly, the magazine is in need of better editorial oversight).

FJ has taken all of the text from the magazine to create the string $S$ of length at most $10^6$ characters. From this, he would like to remove occurrences of a substring $T$ of length $\leq 100$ characters to censor the inappropriate content. To do this, Farmer John finds the \textit{first} occurrence of $T$ in $S$ and deletes it. He then repeats the process again, deleting the first occurrence of $T$ again, continuing until there are no more occurrences of $T$ in $S$. Note that the deletion of one occurrence might create a new occurrence of $T$ that didn't exist before.

Please help FJ determine the final contents of $S$ after censoring is complete.

\textit{INPUT FORMAT: (file censor.in)}

The first line will contain $S$. The second line will contain $T$. The length of $T$ will be at most that of $S$, and all characters of $S$ and $T$ will be lower-case alphabet characters (in the range a..z).

\textit{OUTPUT FORMAT: (file censor.out)}

The string $S$ after all deletions are complete. It is guaranteed that $S$ will not become empty during the deletion process.

\textit{SAMPLE INPUT:}
whatthemomooofun
moo

\textit{SAMPLE OUTPUT:}
whatthefun

The solution is available from http://usaco.org/current/data/sol_censor_bronze.html.