MID2

Exam time: 9:20am-12:10pm

During the exam:
No chat and discussion.
No mobile phone.
Do not use the browser to search for any information, including downloading OJ programs.
Do not use any communication software.
Do not change seats at will.
If the above situation is discovered by TA, the test will be counted as 0 points.

You can check books and related paper materials during the exam.
You can put the lesson slides in the computer before the exam started.
But downloading any information after the start of the test is considered cheating and is counted as 0 points.

You can use the USB flash drive to backup projects to avoid the computer crash and program being washed out by the recovery card.
But please plug in the USB flash drive into the computer before the exam starts.
Plugging in the computer after the start of the exam is regarded as cheating and will be counted as 0 points.

If an OJ system failure occurs during the exam, the exam time will be increased after the OJ system restarts. If the OJ system fails and cannot be used during the exam, students will be asked to submit the project folder at the end of the exam, and TA will manually upload it to the OJ system for score judgement afterwards.

✓

1. Print Triangle Waves

In this problem you are to generate several triangular waveforms according to some specified amplitude values.

Input:

The input is a string composed by *N* (1 ≤ *N* ≤ 20) wave amplitudes. Each wave *Amplitude* is represented by a single digit number (2 ≤ *A* ≤ 9).

Output:

For the output of your program, you will be printing waveforms between each wave amplitude. The horizontal "height" of each wave equals the amplitude. The waveform itself should be filled with integers on each line which indicate the "height" of that line.

Sample Input:

```
232
```

Sample Output:

```
1
22
1
22
333
22
1
22
1
```

2. Sorting Points

There are N points in 2D Cartesian coordinate denoted as $P_i = (x_i, y_i)$ for ith point. Write a program to reorder those points from bottom-left to top-right (sort x and y in ascending order prioritizing on the x value).

Input:

The input begins with an integer N ($0 < N \leq 10$) in the first line by itself representing the number of points to follow. Each following line consists of two floating numbers separated by a space. The numbers are the values of x and y in point respectively.

Output:

Show the reordered result. Each number should be rounded to 2 decimal places.

Sample Input:

```
5
1.2 3.7
5.98 7.75
10.13 7.75
-7.51 -3.7
1.2 -6.23
```

Sample Output:

```
-7.51 -3.70
1.20 -6.23
1.20 3.70
5.98 7.75
10.13 7.75
```

✓

3. Caesar Cipher 2

One of the oldest known encryption techniques is the Caesar cipher, attributed to Julius Caesar. It involves replacing each letter in a message with another letter that is a fixed number of positions later in the alphabet. (If the replacement would go past the letter Z, the cipher "wraps around" to the beginning of the alphabet. For example, if each letter is replaced by the letter two positions after it, then Y would be replaced by A, and Z would be replaced by B.) Write a program that encrypts a message using a Caesar cipher. The user will enter the message to be encrypted and the shift amount (the number of positions by which letters should be shifted).

Input:

Input will consist of two lines; first line is the message to be encrypted. You may assume that the message does not exceed 100 characters. Second line is the shift amount (an integer) used in Caesar cipher.

Output:

Print encrypted message. Characters other than letters should be left unchanged. Lower-case letters remain lower-case when encrypted, and upper-case letters remain upper-case.

Sample Input:

```
Hello world!
4
```

Sample Output:

```
Lipps asvph!
```

Output:

1. Please implement the Caesar Cipher function in `func.h`. Any modification in `main.c` will be discarded during compilation.
2. Use `strlen` to get the length of input message.

D 3/15

4. String Replacement

Write a program to replace string in the given input with another string.

The string to be replaced in the input string **may appear multiple times.**

Input:

The input will consist of three lines of string; First line is the original string. Second line is the string segment needs to be replaced in the original string. Third line is the new string used to replace the matched string segments in the original string.

The length of all input strings will not exceed 50.

The length of the new string after replacement will not exceed 100.

Output:

Print the new string after replacing the specific string in the original string with given string.

Sample Input:

```
Hello World!
World
C Programming
```

有多個重複

Sample Output:

```
Hello C Programming!
```

5. Infix to Postfix 2

Infix and postfix are traversal methods commonly used in compilers and calculators.

For example, a single expression can be written in each form

- Infix: ((1+2)*3)/4+5^6
- Postfix: 12+3*4/56^+

The advantage of postfix expression is that parentheses are unnecessary to prevent ambiguity.
In our traversal the following symbols are operators with precedence rules going from highest to lowest:

- ^ power
- */ multiply and divide
- +- add and subtract

Another symbol used in the infix expression is parentheses ( ).
Anything warped by parentheses should be processed first.
If parentheses warped by other parentheses(nested), process the innermost content first.

Please read "Infix2Postfix.pdf" attached in project for more details.

Input:

There are no spaces in infix expression and its length will not exceed 50.
The range of numbers in the expression is [0, 999].

Output:

Print out postfix notations.

Sample Input:

```
((1+2)*3)/4+5^6
```

Sample Output:

```
12+3*4/56^+
```

Hints:

Stack and Array.

6. Message Decoding

Chico and Maria are relatives that live in different towns. As they inhabit a rural area, it is very difficult for them to keep in touch. One way they found to overcome their communication problem was to send a line through their parents that used to visit each other.

The point is that Chico and Maria did not want that their parents read their messages, and they decided to create a secret code for the messages. The code is not very sophisticated, but you should keep in mind Chico and Maria are just children.

In general, the meaning of a message is based on a letter of each word, in a way that they will form a message with the first letter of the first word, the second letter of the second word and so on. If a word does not have enough letters, the following word should be used. For example, if you are analyzing the third word, you should consider its third letter, but if it just has two letters, then you should try to form a decoded word with the third letter of the fourth word.

When the end of a line is reached, you should finish the current decoded word and should start to form another one from the first letter of the first word in the next line.

Your task is to translate a message according to Chico and Maria's secret code.

Input:

The first line of input gives the number of cases, *T* ($1 \le T \le 30$), then there is a blank line before the first test case. Each test case represents a message, which is composed by $1 \le N \le 100$ lines and each line is composed by $1 \le M \le 30$ words. Two words in the same line are separated by one or more white spaces. A word is formed by the letters A-Z and a-z and has at most 30 letters. The only symbols that appear in the input are the alphabetic letters and white spaces. There will be a blank line after each message.

Output:

For each test case you must print the number of the test case and each word of the decoded message, one per line (look the sample output for the exact format). You must print a blank line between each test case.

Example Input:

```
2


Hey good lawyer
as I previously previewed
yam does a soup


First I give money to Teresa
after I inform dad of
your horrible soup
```

最多 30 筆 case.

每筆 case 最多 100 行

每行最多 30 個字

Example Output:

```
Case #1:
How
are
you

Case #2:
Fine
and
you
```