CSC 2100B
Data Structures

Midterm Programming Examination
10:00AM - 1:00PM March 28, 2009, Rm. 924, HSB

Notes

1. **Limits:** For all questions, unless explicitly specified in the question, the time limit is 15 seconds and the memory limit is 32M bytes.
2. **Data Type:** For all questions, unless explicitly specified in the question, the numbers that appear in the input and output can be stored in an integer without causing overflow.
3. **Input/Output:** For all questions, the input/output is **stdin** and **stdout**.
4. **About newline character:** In the sample inputs and outputs, unless explicitly specified in the question, **\n** represents a newline character, and they are invisible in the actual inputs or outputs.
5. **Question ID:** The question IDs for questions $1, 2, ..., 9$ are **q1, q2, ..., q9**.

---

1 Prefix and Suffix

Let $S = s_1 s_2 ... s_m$ and $T = t_1 t_2 ... t_n$ be two strings and $m \leq n$. S is the prefix of T if $t_i = s_i$ for $i = 1$ to m. S is the suffix of T if $t_i = s_i$ for $i = n - m + 1$ to n.

In this problem, you are given N ($N \leq 100$) pairs of strings (S_j, T_j) and you need to determine whether S_j is the prefix and/or suffix of T_j. The input has N lines (N is not part of the input). In each line, there is a pair of strings for which you need to determine their relationship.

Input

Each line of input contains two strings. The two strings are separated by a blank character. The length of a string is at most 40. The strings may consist of alphanumeric characters, underscore('_') and hyphen ('-').

Output

For each pair of input strings (S, T), output "Prefix" if S is a prefix of T, "Suffix" if S is a suffix of T, "Both" if S is both prefix and suffix of T, or "Neither" if S is not prefix nor suffix of T. The comparison should be case sensitive.

Sample Input

```
relation relationship\n
string substring\n
gram programming\n
sample sample\n
superstring string\n
peer peer-to-peer\n
sample SAMPLE\n
```

Sample Output

```
Prefix\n
Suffix\n
Neither\n
Both\n
Neither\n
Both\n
Neither\n
```

2 Day of the Week

Given the day of the week of a date, e.g., 1st of March is Sunday, you need to find out what week day another given date is in the same year. You can assume that this is not a leap year, i.e., there is no 29th of February.

Input

The first line is an integer N that is the number of test case. The second line is a blank line. After this, there are N input blocks. The first line of an input block shows the weekday of a specific date in the format "mm dd w". e.g., "03 28 6" means "28th of March is Saturday", "03 29 0" means "29th of March is Sunday". Then it is followed by Q queries, each query consists of a date in the same year. The date is in the format "mm dd". e.g., "03 28" means "28th of March". Two input blocks are separated by a blank line.

Output

For each query, output the weekday of the date in the same year in a new line, i.e., output "0" if the date is Sunday, "1" if it is Monday, "2" if it is Tuesday... The output for two test cases should be separated by one blank line.

Sample Input

```
2\n
\n
03 01 0\n
02 28\n
06 17\n
10 22\n
\n
04 13 2\n
05 21\n
11 04\n
08 30\n
```

Sample Output

```
6\n
3\n
4\n
\n
5\n
4\n
1\n
```

3 Stirling Numbers of the Second Kind and Bell Numbers

The Sirling numbers of the second kind $S(n, k)$ count the number of ways to partition a set of n elements into k nonempty subsets. They obey the recurrence $S(n, k) = S(n-1, k-1)+kS(n-1, k)$ with $S(n, 1) = S(n, n) = 1$ for all non-negative integer n. The sum $B_n = \sum_{k=0}^{n} S(n, k)$ is the $(n+1)$-th Bell number (B_0 is the first Bell number). It is the number of partitions of a set with n members, or equivalently, the number of equivalence relations on it. Starting with $B_0 = B_1 = 1$, the first few Bell numbers are:

$$1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, ...$$

Your tasks are to compute the Bell numbers.

Input

The first line of input is an integer N which is the number of test cases. The second line is a blank line. Each of the following N lines has one integer n_i. You are required to compute the $(n_i + 1)$-th Bell number, B_{n_i}. You may assume that the largest n_i is no larger than 15.

Output

The output should consist of N lines. For each test case, you should print B_{n_i}, the (n_i+1)-th Bell number, on a single line.

Sample Input

```
5\n
\n
2\n
3\n
4\n
8\n
5\n
```

Sample Output

```
2\n
5\n
15\n
4140\n
52\n
```

4 Finding Roots of Polynomials

Given a polynomial $\sum_{i=0}^{d} a_i x^i$ and two integers a and b, you need to find if there is a root that is an integer that is no smaller than a and no greater than b. A root of a polynomial is a number which if substituted into the variable will make the polynomial evaluates to zero, e.g., 1 and -1 are roots of $x^2 - 1$.

Input

The first line of input is a number N that indicates the number of test cases. The next line is a blank line. The following are N test cases. A test case has two lines. In the first line of a test case, there is a sequence of integers. The i-th integer is the coefficient of the term x^i in the polynomial. e.g., "1 2 3 4" represents the polynomial $1 + 2x + 3x^2 + 4x^3$. The second line has two integers a and b. a is the lower limit of the range and b is the upper limit of the range. Two test cases are separated by a blank line. You may assume that in the given range, there is at most one root.

Output

For each test case, if the given polynomial has an integer root in the specified range [a, b], then output that root. If there is no such root, then output "No root is found in [a, b].". The output for individual test case should be in one single line.

Sample Input

```
3\n
\n
-1 0 1\n
0 2\n
\n
10 7 1\n
-4 0\n
\n
-6 -3 6 3\n
10 20\n
```

Sample Output

```
1\n
-2\n
No root is found in [10, 20].\n
```

5 Tree Reconstruction

Given a binary search tree T_1 of $2^k - 1$ nodes for some positive integer k, you need to find the binary search tree T_2 of the smallest depth that contains the same set of elements and output the new tree in preorder and postorder.

Input

The first line of input is an integer N that is the number of test cases. The second line is a blank line. After this, there are N input blocks, one for each test case. Each input block consists of a number of lines that describe the structure of the tree. Each line in the input block has three integers i, j, and k. This means that node j and node k are the left and right childs of node i respectively. If j (respectively k) equals 0, then node i has no left (respectively right) child. Two input blocks are separated by a blank line. You may assume that no two nodes in a tree have the same label and the description for a node always comes before that of its descendents.

Output

For each test case, you should output the elements in the corresponding tree in preorder and postorder, each in a single line. Output for two test cases should be separated by one blank line.

Sample Input

```
2\n
\n
1 0 2\n
2 0 3\n
3 0 0\n
\n
3 2 5\n
2 1 0\n
1 0 0\n
5 4 7\n
4 0 0\n
7 6 0\n
6 0 0\n
```

Sample Output

```
2 1 3\n
1 3 2\n
\n
4 2 1 3 6 5 7\n
1 3 2 5 7 6 4\n
```

6 Queue with Extra Functions

Implement a queue that supports the following operations:

1. **Enqueue:** Insert an element of integer value at the end of the queue
2. **Dequeue:** Output and remove the element at the front of the queue if it exists
3. **Return_Sum:** Return the sum of all the elements in the queue (without modifying the content of the queue, if the queue is empty, then the sum is 0)
4. **Add_Constant:** Add a constant to all the elements in the queue

Input

The input consists of a sequence of commands. A command starts with the name of the operation and follows by the operand if there is one. For the **Enqueue** command, the operand is the value of the element to be inserted. For the **Add_Constant** command, the operand is the constant to be added to each element in the queue.

Output

For each **Dequeue** command, you should output the element that is dequeued in a new line. In case if the queue is empty, you should report "Stack Empty" in a line. For each **Return_Sum** command, you should output the sum of all the elements in the queue in a new line.

Sample Input

```
Enqueue 10\n
Return_Sum\n
Add_Constant 16\n
Return_Sum\n
Enqueue 32\n
Return_Sum\n
Add_Constant -10\n
Return_Sum\n
Enqueue 5\n
Enqueue 7\n
Dequeue\n
Dequeue\n
Add_Constant 5\n
Return_Sum\n
Dequeue\n
Dequeue\n
Dequeue\n
```

Sample Output

```
10\n
26\n
58\n
38\n
16\n
22\n
22\n
10\n
12\n
Stack Empty\n
```

7 Queue with Extra Functions (With Time Limit)

The problem specification, input and output of this question are same as those of question 6, except that your program is now required to complete in a tighter time limit of 1 seconds. Note that the acceptance and time penality for questions 6 and 7 are counted separately. You have to submit to both to get full marks for both questions.

8 Find and Replace

In this problem, you are going to implement the **Find and Replace** function provided in typical document editing software. Given a list of pairs of words (x_i, y_i) and a passage P, you need to replace each occurrence of x_i in P by y_i.

Input

The first line of input is an integer L ($L \leq 15000$)that is the length of the word pair list. The second line is a blank line. It is followed by L pairs of words in the list, each pair in one single line. A word consists of numbers, lower and/or upper case alphabetic letters only (the comparison should be case sensitive). The length of a word is at most 40. You may assume that the first words in all word pairs are distinct and the second word in a pair never appears as a first word for another pair. After the word list, there is a blank line and then a passage (of length no longer than 200000 characters) follows. The passage is given in one single line.

Output

You need to output the passage with words replaced according to the given list. For each word pair (x_i, y_i) in the list, whenever the word x_i appears in the passage, you should replace it by y_i. All punctuation marks and space should be kept unchanged. They are not part of a word. You may assume that the only punctuation marks that can appear in the passage are the comma(',') , period('.') and hyphen('-'). The replacement should be done only in one pass.

Sample Input

```
9\n
\n
abc xyz\n
abcd wxyz\n
cde ghijk\n
efgh mnopq\n
ijklm rstuv\n
FGH OPQ\n
rstuvwx bcdefg\n
jklmn00 angho\n
1234 5678\n
\n
abc xxxx FGH 999, yurst  angho 1234. abc-efghij ghijk rstuvwx ABC -- fgh.\n
```

Sample Output

```
xyz xxxx OPQ 999, yurst  angho 5678. xyz-efghij ghijk bcdefg ABC -- fgh.\n
```

9 Find and Replace (With Time Limit)

The problem specification, input and output of this question are same as those of question 8, except that your program is now required to complete in a tighter time limit of 1 seconds. Note that the acceptance and time penality for questions 8 and 9 are counted separately. You have to submit to both to get full marks for both questions.