DATA STRUCTURES AND ALGORITHMS

ASSIGNMENT-2
Language allowed: C

January 13, 2020

A. Rotations

Roshan has a non-circular doubly linked list of integers. He wants to find the K^{th} *left* or *right* rotation of that linked list. Help Roshan find the the rotation he desires.
Note that this problem has to be solved using doubly linked lists only. No use of arrays should be made to store or manipulate the integers.

Input

The first line contains a sequence of integers B ($-10^9 \leq B_i \leq 10^9$). It is guaranteed the length of the sequence will be less than 10^6. The next line contains a integer K ($0 \leq$ K $\leq 10^9$) followed by a character X (X $\in \{$R, L$\}$) denoting the K^{th} left rotation has to be found if X is **L** or the K^{th} right rotation has to be found if X is **R**.

Output

Output a sequence of space-separated integers denoting the K^{th} left/right rotation of the given linked list according to the problem.

---

input
```
10 12 -45 -54 162 3906 4 -1 59 345 7
3 R
```

output
```
59 345 7 10 12 -45 -54 162 3906 4 -1
```

---

input
```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
407941 L
```

output
```
6 7 8 9 10 11 12 13 14 15 16 1 2 3 4 5
```

B. Jumps

Goku and Gohan are stranded on the planet 0 for quite a while. After some analysis, they realize that they can jump to planet $ax+b$ or $cx+d$ or $ex+f$ from planet x (the planet they are currently on) in *one single jump*. As they need to reach planet Y as soon as possible, can you help them find the minimum number of jumps they need to make to reach it for given values of a, b, c, d, e and f?

Input

The first line contains six space separated integers a, b, c, d, e and f ($0 \leq$ a, b, c, d, e, f $\leq 10^6$) denoting the coefficients of the jumps Goku and Gohan and make. The next line contains a single integer Q ($1 \leq$ Q $\leq 10^5$) denoting the number of queries for the problem. The last line of input contains Q space separated integers ($0 \leq Y_i \leq 10^6$) denoting the planet they have to reach. The planets are numbered from 1 to N continuously.

Output

Print Q lines, each containing a single integer, where the i^{th} integer denotes the minimum number of jumps needed to reach the planet Y_i. If a certain planet is unreachable, print -1.

---

input
```
2 3 8 0 1 6
4
5 456 12 78
```

output
```
-1
5
2
4
```

C. Construct it!

After learning OOP, you decide to build a data structure that resembles the C++ vector that has methods to *insert* elements at random locations, *remove* elements at random locations, *fetch* any element in the data structure and *reverse* the data structure. You plan to implement the following using *singly linked lists*. The data structure is 1-indexed.

Input

The first line of input contains a single integer N ($1 \leq N \leq 1000$) the number instructions that need to be performed on the data structure. The following N lines each contain instructions of the format:
a. *add Y* ($-10^9 \leq Y \leq 10^9$): add element Y to the end of the list.
b. *insert Y X* ($-10^9 \leq Y \leq 10^9$): insert element Y at the X^{th} position. It is guaranteed that X will be less than or equal to length of the data structure.
c. *remove Y* ($-10^9 \leq Y \leq 10^9$): remove the first and last occurrence of element Y from the data structure.
d. *reverse*: reverse the data structure.
e. *fetch X*: return the element at the X^{th} position. It is guaranteed that X will be less than or equal to length of the data structure.

Output

After each operation, print the data structure element-wise. During the remove operation if specified element is not present in the data structure, print "Element not found". The data structure is empty, print a blank line.

```
input
12
add 3
add 4
reverse
insert -34 2
reverse
fetch 1
insert 7655 1
add 3
add 3
add 334
remove 23
remove 3

output
3
3 4
4 3
4 -34 3
3 -34 4
3
```

```
7655 3 -34 4
7655 3 -34 4 3
7655 3 -34 4 3 3
7655 3 -34 4 3 3 334
Element not found
7655 -34 4 3 334
```

D. Babysitting

Bored of babysitting Halley and Micheal, you (Raj) and Howard decided to play yet another silly game of yours. In this game, Howard constructs an undirected graph while you are blindfolded. Next, you have to feel your way around the graph while making a mental note of its nodes and edges. Finally, you have to report the vertices that are disconnected from the *larger section* of the graph. The *larger section* of a graph is the connected part of a graph that has maximum number of vertices in it. Howard thinks that you can't do it. Prove him wrong.

Input

The first line of input contains two space-separated integers N ($1 \le$ N ≤ 500) denoting the number of vertices in the graph Howard constructs and M ($1 \le$ M $\le \frac{N(N-1)}{2}$) denoting the number of edges in it. Each of the following M lines contain two space-separated integers U_i, V_i ($1 \le U_i$, $V_i \le N$) denoting an edge between nodes U_i and V_i. It is guaranteed that a unique *larger section* will be present. The vertices are numbered serially from 1 to N. Nodes and vertices are used interchangeably.

Output

In the first line, print the number of nodes that are not connected to the *larger section* of the graph and in the next line print the nodes itself. The nodes should be printed in ascending order. If no disconnected nodes are present, just print -1.

---

input

```
14 16
3 7
3 13
13 11
8 14
11 10
10 9
9 6
7 6
2 1
12 14
1 14
2 12
5 7
6 5
9 5
10 5
```

output

```
6
1 2 4 8 12 14
```

E. SERN

While using the IBM 5100 to hack into SERN's systems, Okabe Rintarou has an epiphany that he need not access all the servers to gain necessary information. He realizes that if *he accesses a certain server S, he can access the information on all the servers reachable from S.* To complete the Phone Microwave machine he needs to access the information on *all the servers* in SERN. As he doesn't want SERN to know about the intrusion, he wants to complete this task in *one go*, i.e. by accessing *exactly one server.* As you have the blueprint of the servers and their connections, he turns to you for help. Help Okabe figure out which server he must access so that he can access the information on all the servers in *one go*. For security reasons, the direct connections between server pairs are unidirectional. Server S_2 is said to be reachable from server S_1 if there exists some connection from S_1 to S_2, which means that S_1 can access all information on S_2.

Input

The first line of input contains two space-separated integers N ($1 \leq$ N ≤ 500) denoting the number of servers in SERN and M ($1 \leq$ M $\leq \frac{N(N-1)}{2}$) denoting the number of unidirectional connections between those servers. Each of the following M lines contain two space-separated integers U_i, V_i ($1 \leq U_i, V_i \leq N$) denoting a unidirectional connection from server U_i to V_i. The servers are numbered serially from 1 to N.

Output

Print the node labels (server no.) of all such servers Okabe can access that will help him complete his task, in ascending order. If there are no such servers, print "BAD LUCK".

---

input

```
6 9
1 2
1 3
4 1
6 4
1 4
6 3
6 5
1 5
5 2
```

output

```
6
```

explanation

Accessing just server 6, Okabe can access information on all other servers as all other servers are reachable from server 6 by some or the other connection, and can therefore finish his task in *one go*

F. Merge it!

In this task, you are required to merge two sorted *doubly (non-circular) linked lists*. You have to complete this problem using doubly linked lists only and no use of arrays can be made to store the numbers or make any manipulations.

Input

The first line contains a sequence of space-separated integers A ($-10^9 \leq A_i \leq 10^9$). The next line also contains a sequence of space-separated integers B ($-10^9 \leq B_i \leq 10^9$). It is guaranteed that the length of both the sequences will be less than 10^6 and will be given in a sorted order.

Output

Output a single sequence of space-separated integers, which is the result of merging the given two linked lists. Note that, the final result should be stored in a *doubly linked list* as well.

---

input

```
-12 -10 -4 -3 -3 1 2 9 12 16 19 28 30 44
-1 -1 -1 6 7 9 10 18 278
```

output

```
-12 -10 -4 -3 -3 -1 -1 -1 1 2 6 7 9 9 10 12 16 18 19 28 30 44 278
```

G. Star Wars and Palindromes

The new Star Wars movie is out and you along with your friends Sheldon, Howard and Raj are in the line to buy the tickets. Despite Sheldon's endless pleas to leave early, you all had taken your own sweet time to get ready. Now with only one ticket left and a really long queue of fans fighting for it, the counter salesman gives everyone a string and asks them to find the longest palindromic substring in it. First to get the answer wins the ticket. As Sheldon is really cross with you all, it's up to you, Leonard, to figure out the answer as soon as possible.

Input

The only line of input contains a single string S ($1 \leq |S| \leq 10^6$) consisting of lowercase letters.

Output

In the first line, print the length of the longest palindromic substring in the given string and in the next line print the substring itself. If there are multiple answers, print any.

---

input
```
trweexxbababxxxuuvuuio
```

output
```
9
xxbababxx
```

---

input
```
thebigbangtheory
```

output
```
1
b
```

H. Levels

Given a tree T, your task is print the nodes belonging to the K^{th} level from the root, i.e. at a distance K from the root node. It is given that the root node is always numbered 0 and considered as level 0. Assume that every non-root node is reachable from the root.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10^3$) denoting the number of vertices in the given tree T, which are labeled from 0 to $N-1$, with 0 being the label of the root. The next line contains a sequence of N-1 space-separated integers P_i ($1 \leq i \leq N-1$) denoting that the parent of i^{th} vertex is P_i. The third and last line contains a single integer K ($1 \leq \mathrm{K} \leq 10^3$), denoting the level at which the nodes have to be found.

Output

In the first line print the number of nodes present in the K^{th} level and in the second line, print the nodes themselves in ascending order. If no nodes are present in the K^{th} level, i.e. all nodes are at a distance less than K from the root, just print -1.

```
input
17
0 14 0 0 0 1 4 5 5 9 5 3 6 12 4 0
2

output
7
6 7 8 9 11 12 15
```

```
input
15
0 1 1 0 4 4 2 2 3 3 5 5 6 6
4

output
-1
```

I. Find the difference

You find a very interesting question in your maths test. Given a positive integer X (in base 10), you have to find the *difference between the largest and smallest permutation of the digits of X.*
Note that, in this problem, a permutation beginning with 0s is *not considered* a proper permutation, i.e. *the smallest and largest permutations cannot have leading 0s.*

Input

The only line of input contains a single integer X ($0 \leq X \leq 10^{10^5}$).

Output

Print a single integer denoting the difference between the largest and smallest permutation of the given number X.

---

input

```
30411367001212
```

output

```
65323218877533
```

---

input

```
100000
```

output

```
0
```

J. Prime Factors

In this task, given an integer X (in base 10), you are supposed to find the **bitwise-xor** of the binary representation of all it's unique prime factors. If X is prime, xor with 0. Write a program to do the same.

Input

The only line of input contains a single integer X ($0 \leq X \leq 10^{15}$).

Output

Print a single integer S, denoting the **xor** of the prime factors of the given number X, in base 10.

---

input
```
6744518270181
```

output
```
39441627327
```

---

input
```
67231
```

output
```
67231
```
