# CS F211
# Data Structures and Algorithms
# Assignment - 0

Allow language: **C**

January 21, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, Use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# A: Big Sum

Depending on the platform, the largest integer data type in C will allow you to store numbers that are tens of digits long. In this question, you will write a program that will enable you to add non-negative integers that are at most a thousand digits long. To this end, create two strings that can be used to store upto 1000 digit positive integers in base 10; Each digit will be a character in the usual positional number system. Obtain these numbers as input from the user and assume that the first number is greater than or equal to the second and that the numbers are non-negative. Your program should compute their sum, store it in a string without padding zeros to the left and print it. *Note: You need to write the entire program in the* `main()` *function without having to write separate functions. Further, you are not allowed to use any header file other than* `stdio.h`

## Input

The first line of input contains the integer A ($0 \leq A \leq 10^{1000}$) represented as a string. The second line of input contains the integer B ($A \leq B \leq 10^{1000}$) represented as a string.

## Output

Print a single string, representing the sum of the two integers provided to you, without zeros padded to the left.

---

input
78977
98173

output
177150

---

input
123
1

output
124

---

input
687304576937243574529852345237 65
119742758248759287298755045879 07

output
807047335186002861828607391116 72

---

# CS F211
# Data Structures and Algorithms
# Assignment - 1

Allow language: **C**

January 21, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# A: Challenge Accepted!

After a mind numbing lockdown, you decide to step out of your house and have a jog in the park nearby. After a few laps around the park, you run into your old friend. He challenges you to find if a given string S contains the sub-string **baab** hidden in it, and wait for it, without using loops, arrays, strings, character pointers and only using recursion. Are you brave enough to accept the challenge?

## Input

The first line of input contains a single integer N ($1 \leq N \leq 500$) denoting the number of characters in the string S. The following line contains a sequence of N lowercase a's or b's.

## Output

If the given string S contains the sub-string **baab**, print "YES", followed by the index at which it is found. In case, the sub-string is present more than once, print all indices at which it is found, in ascending order, in space separated fashion (see sample case for better clarity). If the sub-string is not found in the given string, just print "NO". Note that the indices should be printed in 1-indexed order.

---

input
10
baababbaab

output
YES 1 7

---

input
17
baaaaaabbaaaaabaa

output
NO

---

input
12
aabaabaabaab

output
YES 3 6 9

---

# B: Palindrome?

In this problem, you are once again given a string S, and are asked to check if S is a palindrome. A palindrome is a string which reads the same left to right and right to left. The constraint is to solve this using only doubly-linked-lists. *Note that, no use of arrays or character pointers must be made to solve this problem and that this is case-sensitive palindrome.*

## Input

The first line of input contains a single integer N ($1 \leq N \leq 1000$) denoting the number of characters in the string S. The following line contains a sequence of N lowercase and uppercase alphabets.

## Output

Print "PALINDROME" if given string S is a palindrome or "NOT A PALINDROME" if it is not.

---

input
9
MaLaYaLaM

output
PALINDROME

---

input
6
XOFfoX

output
NOT A PALINDROME

---

input
4
KooK

output
PALINDROME

---

# C: Dinner Time

With the advent of the 2020 batch, Mess-1 faces a grave shortage of seats in the mess. At dinner time, every minute a certain number of students either enter the mess or leave it (assume that, in a minute, students do not enter and leave simultaneously). After dinner, the mess manager is curious to know the maximum number of seats that were occupied during dinner time. As he is busy preparing for the upcoming Moonlight dinner, he asks you to help him out. Assume that the students are very lazy and occupy the first unoccupied seat (unoccupied seat with lowest index) they find.

## Input

The first line contains a single non-zero integer T ($1 \leq T \leq 500$) denoting the number of minutes the mess is open for dinner. Each of the following T lines resembles either of the following query:

- E X: This statement denotes that X ($0 < X \leq 5000$) students enter the mess in that minute.
- L P $Y_1$, $Y_2$, $Y_3$ ... $Y_P$: This statement denotes that P ($0 < P \leq 5000$) students leave the mess that minute from the seats numbered $Y_1$ to $Y_P$. It is guaranteed that the P will be less than or equal to the number of students in the mess at that minute.

Assume that the total number of students in campus are less than or equal to 5000 and seats are 1-indexed (see sample case for better clarity)

## Output

Print a single integer N, denoting the maximum number of seats that were occupied during dinner time.

---

```
input
6
E 5
E 2
L 3 1 4 5
L 1 2
E 3
E 2

output
8
```

---

# D: Amazon Shopping

With Amazon's yearly sale going on, you wish to buy as many goods as your wish with you limited pocket money you have collected till date. As the sale time is running out, you decide to write a simple program that will help you buy as many goods as possible without exceeding your budget. *Hint: Think in terms of subsets and how subsets can visualized with binary numbers.*

## Input

The first line contains a single integer M ($0 \leq M \leq 10^9$) that denotes your pocket money. The following line contains another integer N ($1 \leq N \leq 25$) denoting the number of goods that are for sale in Amazon. The last line of input contains N space-separated integers ($0 \leq P_i \leq 10^9$) denoting the prices of the goods for sale.

## Output

Print a single integer X, denoting the maximum number of goods you can buy without exceeding your budget.

---

```
input
44
6
14 13 21 8 56 3

output
4
```

---

```
input
70
7
71 101 766 125 908 682 75

output
0
```

---

# E: Digital Design

In this problem, you will be implementing a simple shift register and some of its functions using a doubly-linked-list. *Note: You are not allowed to use any other data structure other than linked-lists for this problem. Further, you are not supposed to convert the given bit string to decimal and perform operations on it.*

## Input

The first line contains a bit string B ($1 \leq |B| \leq 1000$) which denotes the initial content of the register. The second line contains a single integer T ($1 \leq T \leq 100$) denoting the number of queries. Each of the following T queries resemble either of the following:

- LR: This denotes that the content of the register has be left-rotated by one unit.
- RR: This denotes that the content of the register has be right-rotated by one unit.
- LS: This denotes that the content of the register has be left-shifted by one unit. (*empty space can be filled with 0*)
- RS: This denotes that the content of the register has be right-shifted by one unit. (*empty space can be filled with 0*)
- INC: This denotes that the content of the register has be incremented by 1.
- DEC: This denotes that the content of the register has be decremented by 1.

## Output

Print the content of the register after all operations have been performed. In case any out-carry is generated, you can ignore it.

---

```
input
11111001
4
LS
LS
LR
INC

output
11001010
```

---

# F: Koro-sensei and Primes

Koro-sensei has come up with a quite an eerie problem on primes for the students of class 3E. Given two numbers A and B, the students have to find the digit which occurs maximum number of times in the primes from A to B (both inclusive). As the class is getting ready for the annual school festival, they all turn to you, Karuma, for help.

## Input

The only line of input contains two space-separated integers A and B ($0 \leq A \leq B \leq 10^{12}$, $0 \leq$ B-A $\leq 10^6$)

## Output

Print two space-separated integers X and Y with X denoting the digit that occurs maximum number of times in the primes from A to B and Y denoting the frequency of X.

---

input
37 101

output
7 7

---

input
2692 15859

output
1 1367

---

# F: Tour de Goraq

Ash Ketchum, on his journey to be the Pokemon master, plans to tour the new archipelago Goraq. Goraq is a set of islands with a few of those islands being connected by long bridges. As part of his tour, Ash always starts from island 0, and tours all islands reachable from it before moving on to another cluster of islands. Help Ash figure out how many clusters of islands are present in the Goraq archipelago and along with number of islands per cluster. *Note: A cluster of islands is a set of islands that are reachable from one another, i.e. they form a connected component.*

## Input

The first line of input contains two space-separated integers N ($1 \leq N \leq 500$) and M ($0 \leq M \leq \frac{N(N-1)}{2}$) denoting the number of islands in the archipelago and the number of bridges between those islands respectively. The following M lines contain two space-separated integers $U_i$ and $V_i$ ($0 \leq U_i, V_i \leq$ N-1) denoting an undirected bridge between islands $U_i$ and $V_i$.

## Output

In the first line, print a single integer X denoting the number of clusters in the archipelago and in the following line, print X space-separated integers denoting the number of islands per cluster (this can be printed in any order).

---

```
input
12 12
5 7
7 9
1 7
4 10
3 8
10 8
10 3
0 2
0 6
6 11
2 11
2 6

output
3
4 4 4

explanation
As you can observe, there are three clusters wiz.  1-5-7-9, 0-2-6-11, 4-10-8-3
```

---

# H: Cost Difference

Let us define the cost of an array as the difference of sum of squares of elements at even indices and sum of cubes of elements at odd indices. In this problem, you need to find the array rotations with maximum and minimum cost and report the difference between those costs. *Sample array rotation: [7 1 -9 6 0 1] → [1 7 1 -9 6 0]*

## Input

The first line contains a single integer N ($1 \leq N \leq 500$) denoting the number of elements in the array. The following line contains a sequence of N space-separated integers ($-10^3 \leq A_i \leq 10^3$) denoting the array A given you.

## Output

Print a single integer S denoting the difference between of costs of the maximum cost rotation and minimum cost rotation.

---

input
```
7
6 -1 8 -1 6 8 -4
```

output
```
1452
```

---

input
```
10
-5 67 12 -78 5 0 -6 100 6 -37
```

output
```
795506
```

---

input
```
5
8 8 8 8 8
```

output
```
0
```

---

# I: Reverse it!

In this problem, you are given a stream of positive numbers. You are expected to generate a singly-linked-list out of them and reverse it. *Note: You must not to store the numbers in any array and are supposed to completely reverse the list by reversing all the pointers, i.e. you must not create a new list to store the numbers in reversed order.*

## Input

The only line of input contains a sequence of numbers A ($0 \leq A_i \leq 10^9$). It is assured that the final number in the sequence will be -1 which can be used to as a termination condition and the length of the sequence will be less than $5 \cdot 10^5$.

## Output

Print the linked list after reversing it, in space-separated format. Note that, you need not print the -1 at the end.

---

input
```
9 4 67 2 7 1 7 655 8 0 91 -1
```

output
```
91 0 8 655 7 1 7 2 67 4 9
```

---

input
```
78 -1
```

output
```
78
```

---

input
```
1 78 4 28 9 68 18 19 36 9 -1
```

output
```
9 36 19 18 68 9 28 4 78 1
```

---

# J. Rangoli

Roshni has drawn a new rangoli design (which can be visualized as an undirected graph) and wishes to colour it. But unfortunately, she has only two colours Red and Blue. She is very particular about her colouring scheme and wants the colour the vertices in the rangoli in such a way that the two vertices connected by an edge have different colour. As Roshni is still perfecting her rangoli design, help her generate a coloring scheme using only the Red and Blue colours available.

## Input

The first line of input contains two space-separated integers N ($1 \leq N \leq 500$) and M ($0 \leq M \leq \frac{N(N-1)}{2}$) denoting the number of vertices and edges in the rangoli. The following M lines contain two space-separated integers $U_i$ and $V_i$ ($0 \leq U_i, V_i \leq$ N-1) denoting an undirected edge between vertices $U_i$ and $V_i$ in the rangoli.

## Output

If coloring the rangoli with just two colours is not possible, print "NEED MORE COLOURS" else, print a sequence of N characters denoting the colouring scheme for the rangoli. Assume the $0^{th}$ vertex is always coloured Red, in case a colouring scheme exists. (see sample case for clarity).

---

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

output
```
RBRBRB
```

---

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

output
```
NEED MORE COLOURS
```

# CS F211
# Data Structures and Algorithms
# Assignment - 2

Allowed languages: **C**

January 29, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# A: Ghot's Riddle

The students taking the C programming course recently learnt how to find the $n^{th}$ number in the fibonacci sequence, but found it to be too easy. So they asked the class ghot to come up with a more difficult function to find. The relation he came up with is defined as follows:

$$f(x) = \begin{cases} f(div_1) + f(div_2) & \text{if } x \text{ has two or more distinct divisors other than 1 and itself} \\ x & \text{if } x \text{ has 1 divisor or less apart from 1 and itself} \end{cases}$$

Where $div_1$ and $div_2$ are the two largest distinct divisors of $n$, other than 1 and the number itself. In this question, you will be provided with a number $n$ and your task is to find $f(n)$. *Note: this is **not** the Fibonacci sequence defined above. Please read the question.*

## Input

The first line of input contains a single integer $N$ ($1 \leq N \leq 10^5$), for which you must find $f(N)$.

## Output

The output should contain a single integer that is the answer as described by the function above.

---

input
10

output
7

explanation
$f(10) = f(5)+f(2)$, Since 5 and 2 do not have any proper divisors, $f(5) = 5$ and $f(2) = 2$ and hence $f(10) = 5 + 2$ which is equal to 7.

---

input
8

output
6

---

# B: Tourism

One day you decide to visit the Grand Line for some sightseeing. The grand line consists of $N$ different islands numbered from 1 to $N$. The waters between the Grand Line are incredibly dangerous and it's easy to get lost, so you buy a log pose to help you navigate. Depending on the island you are currently standing on, the log pose points to zero or more other islands and you can travel to any of these islands. You can start your journey from any island initially. You want to know the maximum number of **distinct** islands that can be reached from a starting island of your choice (the island you start from should also be counted) *Note that from your current island you can only travel to an island that the log pose points to. Keep in mind that you need to find the maximum number of islands that you can possibly reach from the starting island, not a path with the largest number of islands. See sample case for clarity.*

## Input

The first line of input contains two space-separated integers $N$ and $M$ ($1 \leq N \leq 1000$, $0 \leq M \leq 1000$) denoting the number of islands and island transitions you can make respectively. In the $M$ lines that follow, the $i^{th}$ line containing two space-separated integers, $U_i$ and $V_i$ representing that from island $U_i$, one of the islands that the log pose points to is $V_i$.

## Output

Print a single integer X denoting the maximum number of islands that you can reach if you start from some island.

---

input

```
5 5
1 2
1 5
5 4
5 3
3 2
```

output

```
5
```

explanation

if we draw the corresponding possible paths between islands, we can see that from the island 1 we can reach all other islands. From 2, we can't reach any other node. from 3 we can reach 3 and 2. from 4 we can reach only itself. from 5 we can only reach 5 and 4. Clearly we can see that if we start from node 1 our answer would be 5 which is maximum possible.

---

# C: Klutz

$N$ people are participating in the binary blitz competition this time, each numbered from 1 to $N$ and all of them have a particular rating on codeforces. It is known that if two people compete, the one with the higher rating wins for sure. It is guaranteed that the ratings of all the $N$ people are unique. The problem is that no one provided their actual codeforces handle so you don't know anyone's rating. All the $N$ people play exactly once against every other person. So obviously there will be a total of $\frac{N(N-1)}{2}$ games in total played by the $N$ people. The competition is over and you have kept a record of all the matches in the following format: You have $\frac{N(N-1)}{2}$ rows, and each row containing two space separated integers. The first number of these is the winner and the second one is the loser. It turns out you have lost exactly one row from this list and you have to try and fill in the missing record so that it doesn't contradict any of the other rows. If it is not possible to determine the winner and loser, print them in any order.

## Input

The first line contains a single integer $N$ which is the number of people participating ($3 \leq N \leq 200$). The following $\frac{N(N-1)}{2} - 1$ lines contain two space-separated integers each $U_i$ and $V_i$, where $U_i$ is the winner of that round and $V_i$ is the loser.*As an extra challenge, try to solve this with the constraint* $3 \leq N \leq 2000$.

## Output

Output a single line containing two space-separated integers that is the missing entry. If you cannot determine the winner and loser with the give information then print them in any order

---

input

```
4
4 2
4 1
2 1
3 1
2 3
```

output

```
4 3
```

explanation

notice that if we put 3 4 then it would contradict line 1 and line 5

---

# D: Geass Code

You are the leader of the resistance against Britannia and you have $N$ Knightmares that you can use during the revolution. The $i^{th}$ Knightmare initially has a power $P_i$. You decide that the current power of your Knightmares isn't enough so you ask your lead scientist Rakshata to help you out. She comes up with $M$ power multipliers placed one after the other in a row. each multiplier has a different power where $i^{th}$ multiplier has a power $Mult_i$. She uses this in the following way: Start by placing the first index of the multiplier over the first element in $P$. Then for all $1 \leq i \leq M$, $P_i = P_i \cdot Mult_i$. Then move the starting position of the multiplier to the second position of $P$. Then for all $1 \leq i \leq M$, $P_{i+1} = P_{i+1} \cdot Mult_i$. Then move the starting position of the multiplier over to the third element of $P$ and for all $i$ $1 \leq i \leq M$, $P_{i+2} = P_{i+2} \cdot Mult_i$. Keep moving the starting position of the multiplier and multiplying its elements in a similar manner. Do this process a total of $n-m+1$ times. You must output the final powers of all the knightmares after all the operations are completed. Since the numbers can be very large, output them modulo $(10^9+7)$. *Note: Please ensure to use long int/long long int to avoid integer overflow since upon multiplication, the numbers can exceed the size of int.*

## Input

The first line consists of two space-separated integers $N$ and $M$, the number of Knightmares and the number of multipliers. $(1 \leq M \leq N \leq 10^5)$. The second line contains $N$ space-separated integers which are the initial powers of the knightmares $P_1, P_2, ..., P_n$. The third line contains $M$ space-separated integers, representing the power of each of the $M$ multipliers, $A_1, A_2, ..., A_m$ $(1 \leq P_i, A_i \leq 10^9 \ \forall i)$.

## Output

Print a single line of $N$ space-separated integers, the final powers of the knightmares after all operations modulo $(10^9+7)$.

---

input

```
5 3
5 3 1 7 2
1 2 4
```

output

```
5 6 8 56 8
```

explanation

Initially the array $A$ gets multiplied on to the first three elements of $P$, then the second three, and so on. It is easy to see that this is the answer.

---

# E: Who Will Win Today?

Shirogane wants to impress Kaguya and so he tells her that given a set of letters from the english alphabet, he can enumerate all possible words of length K using only those letters in 1 second or less. Obviously, this is an impossible task so he asks you, Ishigami, to bail him out so that he doesn't look stupid in front of Kaguya.

## Input

The first line consists of two space-separated integers $N$ and $K$, the number of letters you can use, and the length of each word respectively (it is guaranteed that $N^K \leq 10^5$ and sum of lengths of all possible words does not exceed $10^5$). The second line contains a string of $N$ **distinct** lowercase English alphabets (no capital letters or special characters) representing the possible letters that you can use. Note that you cannot use any character or letter not present in the string.

## Output

print $N^K$ lines, each line containing a distinct string of length $K$. Two strings are considered distinct if they differ in at least one position.

---

```
input
2 3
ab

output
aab
bbb
abb
bab
baa
aba
bba
aaa
```

---

# F: Koro-sensei and the Powers of Two

While the students of class 3E thought they bought some respite after last week's problem on primes, Koro-sensei is back at it again, with another weird question. Given a number $N$, the students have to find the *number of ways* it can be represented as a sum of power of 2. Can you help them?

## Input

The only line of input contains a single integer N. ($0 \leq N \leq 10^2$)

## Output

Print one integer, $X$ denoting the number of ways $N$ can be expressed as a sum of powers. Please note that reorderings of the same sum do not count as multiple ways - eg. (2+2+1), (2+1+2), (1+2+2) are all treated as the same thing.

---

input
7

output
6

explanation
7 can be represented as: $(4+2+1)$, $(4+1+1+1)$, $(2+2+2+1)$, $(2+2+1+1+1)$, $(2+1+1+1+1+1)$, and $(1+1+1+1+1+1+1)$, since the powers of 2 are $2^0, 2^1, 2^2 \ldots$ which are $1, 2, 4 \ldots$

---

input
4

output
4

explanation
$(4), (2+2), (2+1+1), (1+1+1+1)$

---

# G: The COVID Vaccine

It's finally here and all this can end! However, before normalcy can completely return, enough people need to be vaccinated for everything to be safe. The owner of a vaccine manufacturing facility wants to start selling and shipping vaccines as soon as possible. However, business being business, she wants to ensure that the company spends the *least* amount of money for shipping the products. Vaccines are first stored in small boxes, which are then stacked inside big boxes before being loaded onto cargo ships and planes. All big boxes have a fixed weight $B$, but the weights of the small boxes can vary considerably. Given a list of weights of $N$ small boxes, what is the *minimum* number of big boxes you'd need to ship all the vaccines available? **At most two boxes** $(s_i, s_j)$ **can be placed in one big box**, due to government regulations. $s_i + s_j \leq B$. *Additional thinking: How would the answer change if this government regulation was removed? To be clear this is* ***not*** *a part of the problem statement - just something to think about.*

## Input

The first line of input contains $N$ ($1 \leq N \leq 2 \times 10^3$). The second line of input contains a sequence of $N$ space-separated integers ($s_i \leq B \leq 10^4$), representing the weights of the small boxes that we have. The third line of input contains the number $B$, the weight of the large box.

## Output

Print a single integer X denoting the minimum number of large boxes you would need.

---

```
input
2
2 1
3

output
1

explanation
We can fit both these small boxes in
one big box of size 3
```

```
input
10
9 12 7 5 4 3 3 1 8 4
13

output
5

explanation
One valid configuration:  (12, 1),
(9, 4), (8, 5), (7, 4), (3,3)
```

---

# H: Aggregating the Binary Tree

You are given a complete binary tree in which all the nodes are at the same depth relative to the root. The binary tree is represented as an array $A$, where $A[0]$ is the root. For every $i^{th}$ index, it's left child is stored in $A[2i+1]$ and its right child in $A[2i+2]$. You need to aggregate all the elements in the binary tree following a special rule: $S = \sum_i A_i \times L_i$, where $A_i$ is the value at that node and $L_i$ is the level. *Note: the level of the root node is 1, and the level of it's children would be 2, it's grandchildren 3, and so on.*

## Input

The first line contains a single integer N ($1 \leq N \leq 2^{18} - 1$) denoting the number of elements in the binary tree where $N$ is of the form $2^k - 1$ where $k$ is a positive integer. The next line contains N positive space separated integers.

## Output

Print a single integer S, denoting the aggregate sum of the binary tree.

---

input
3
21 9 7

output
53

explanation
The root of the binary tree is 21 (level 1), while it's left child is 9 (level 2) and its right child is 7 (level 2). Aggregated sum = $(21 \times 1) + (9 \times 2) + (7 \times 2) = 53$

---

input
7
1 2 4 6 7 5 3

output
76

explanation
$1 * 1 + (2 + 4) * 2 + (6 + 7 + 5 + 3) * 3 = 76$

---

# I: The Dinosaur Conundrum

The inauguration of the Jurassic Park is about to happen tomorrow, and John Hammond has a problem on his hands. While they could successfully clone $N$ different species of dinosaurs, they soon realized that dinosaurs would start attacking each other as soon as they woke up. After a lot of observation, the handlers notice that not all dinosaurs are aggressive towards each other - only dinosaurs that have been cloned from the same raw genetic material fight each other. Each dinosaur has a tag, which indicates the source of their genetic material. If the tags of two dinosaurs start with the same first character, they fight, otherwise they don't. You have $C$ different cages to store the dinosaurs in, each of which can store any number of dinosaurs. Given the tags of all $N$ dinosaurs, separate the dinosaurs into cages such that a *minimum* number of pairs of dinosaurs fight. For eg, if a cage has three dinosaurs - ("XAHSG-2737", "XAGFS-89" and "XSGFS-999") - each dinosaur will fight with the other two to form 3 pairs of fights - ("XAHSG-2737", "XAGFS-89"), ("XAHSG-2737", "XSGFS-999"), ("XAGFS-89" and "XSGFS-999").

## Input

The first line of input contains two space-separated integers, $N$ and $C$. The following $N$ lines contain strings (of length $\leq 20$) denoting the tags of all the dinosaurs. It is guaranteed that the characters in the tag will be from the set [A-Z,0-9].

## Output

Print one integer, $P$, the number of pairs of fights that will happen in the minimal case.

---

input
```
6 4
XGB189
ABFG99
XFG983
BST535
TIN846
XC67TP
```

output
```
0
```

input
```
6 2
FFT998
DFFT99
BWFT36
BERT99
BLIP87
BHIJ78
```

output
```
2
```

---

# J. Air Tickets

You have been invited to the Annual Computing Conference which is going to be held in Tokyo this year. Sadly, the organizers have declined to reimburse your flight tickets to the conference. So, you decide to take the flight path (a flight path may have multiple hops) to Tokyo which will cost you the least. Given the all the hops possible between the major cities of the world, find the cost of the minimum flight path. Assume that each hop costs you the same, i.e. \$100. *Note: You must use adjacency lists to solve this problem.*

## Input

The first line of input contains four space-separated integers N ($1 \leq N \leq 5000$), M ($0 \leq M \leq \frac{N(N-1)}{2}$), S ($0 \leq S \leq$ N-1) and T ($0 \leq T \leq$ N-1) denoting the number of cities, the number of hops, the city you live in and city you wish to reach (Tokyo) respectively. The following M lines contain two space-separated integers $U_i$ and $V_i$ ($0 \leq U_i, V_i \leq$ N-1) denoting an undirected hop between the cities $U_i$ and $V_i$. It is guaranteed that atleast one flight path exists to Tokyo.

## Output

Print a single integer C, denoting the cost of the minimum cost flight path to Tokyo from your city.

---

input

```
10 13 0 8
0 1
0 2
2 1
2 3
3 1
3 4
4 5
5 6
5 7
4 7
7 9
9 5
8 7
```

output

```
500
```

explanation

As you can observe, the minimum cost flight involves the hops, 0-2-3-4-7-8 (this is one of the flights paths, but it has minimum cost).

---

# CS F211
# Data Structures and Algorithms
# Assignment - 3

Allowed languages: **C**

February 3, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# A: Ice Cream

There are $N$ kids and $M$ ice cream cones. The $i^{th}$ ice cream cone has a size $S_i$. Each kid has a preferred cone size with the $i^{th}$ child preferring a size $A_i$. Each child will accept an ice cream cone if the size of the ice cream cone is between $A_i \pm k$ inclusive. You now have to find the largest number of children that will get an ice cream cone if you distribute the ice cream cones optimally. *Please note that you cannot give an ice cream cone to more than one kid and each child can have no more than one ice cream cone.*

## Input

The first line of input contains three space-separated integers $N$ and $M$ ($1 \leq N, M \leq 10^5$) and ($1 \leq k \leq 10^9$). The second line contains $N$ space-separated integers representing the array $A$ that is the preferred sizes of ice cream cones for each of the $N$ kids. The third line contains $M$ space-separated integers representing the size of the array $S$, the sizes of the available ice cream cones.

## Output

The output should have exactly one integer, the maximum number of children that can get an ice cream cone if they are distributed optimally.

---

input

```
4 3 5
60 45 60 80
30 75 60
```

output

```
2
```

explanation

here a kid with preference 60 can get the ice cream cone with size 60 and the kid with preference 80 can get an ice cream cone of size 75. Hence the answer is 2. It is easy to see that we can satisfy no more than two kids.

---

# B: Assassins

You have a total of $N$ assassins and each has a skill $a_i$. There also exist $M$ nobles. Each of these nobles has a bodyguard with a skill $b_i$ and a certain amount of gold $g_i$. An assassin can kill a bodyguard of a noble if the assassin's skill is greater than or equal to the bodyguard's skill ($a_i \geq b_j$). If an assassin kills a bodyguard he can steal all the gold of the noble. How much gold can each assassin steal? Please calculate the answer for each assassin *independent* of the other assassins. Do not assume that if a noble's gold is stolen by one assassin then other assassins can't steal from him. The assassins are not actually killing the body guards as such, you just need to find how much gold each of them *can* steal hypothetically.

## Input

The first line contains two integers $N$ and $M$ ($1 \leq N, M \leq 10^5$), the number of assassins and nobles. The second line contains $N$ integers representing the array $a$, where $a_i$ is the skill of the $i^{th}$ assassin ($-10^9 \leq a_i \leq 10^9$). Then $M$ lines follow, where the $i^{th}$ line contains the two integers $b_i$ and $g_i$, the skill of the bodyguard of the $i^{th}$ noble and the amount of his gold. ($-10^9 \leq b_i, g_i \leq 10^9$)

## Output

Print one line containing $N$ integers, where the $i^{th}$ integer represents the maximum gold that the $i^{th}$ assassin can steal.

---

```
input
5 4
1 4 3 2 5
4 2
0 1
2 8
9 4

output
1 11 9 9 11

explanation
The first assassin can only steal gold from the second noble.  The second can steal
gold from the first, second, and third nobles.  The third can steal from the second
and third.  The fourth can steal from second and third.  The fifth can steal from first,
second and third.
```

---

# C: Tree Planting

You want to plant $N$ trees in your garden. Your garden can be represented by a number line that contains fertile spots at certain points. In particular there are $M$ fertile spots ($M \geq N$), $x_1$, $x_2$, $x_3$ ... $x_M$ where you can plant a tree. You can only plant a tree in a fertile spot and a fertile spot can have a maximum of one tree. As we know from our high school biology, two trees cannot be kept too close to each other otherwise they will take up each others water and nutrients. To make sure that all the trees are healthy, you want to plant them in such a way that the minimum distance between any two of them is as large as possible. What is the largest possible minimum distance between any two trees?

## Input

The first line of the input contains two space-separated integers $N$ and $M$, ($1 \leq N \leq M \leq 10^5$). The second line of input contains $M$ space-separated integers, $x_1$, $x_2$, $x_3$ ... $x_M$, representing the co-ordinates of the fertile spots ($0 \leq x_i \leq 10^9$).

## Output

Output a single integer which is the largest possible minimum distance you can get by planting the trees in some way.

---

input

```
3 5
2 1 8 4 9
```

output

```
3
```

explanation

we can get a minimum distance of $3$ if we place trees at positions $1, 4$ and $8$. It can be easily seen that it is not possible to get a larger minimum distance no matter where you plant the trees

---

# D: Good Pairs

Given two arrays $x$ and $y$, both containing $N$ elements, find the number of pairs of integers $i, j$ such that $x_i + x_j + k_1 > y_i + y_j + k_2$ where $(i < j)$.

## Input

The first line contains three integers $N$, the size of the arrays, and $k_1$ and $k_2$ ($1 \leq N \leq 10^5$, $-10^9 \leq k_1, k_2 \leq 10^9$). The second line contains $N$ integers representing the array $a$. The third line contains $N$ integers representing the array $b$ ($-10^9 \leq a_i, b_i \leq 10^9$).

## Output

Print a single integer, the number of such pairs.

---

input

```
5 4 4
4 8 2 6 2
4 5 4 1 3
```

output

```
7
```

explanation

The pairs $i, j$ are $(1,4), (2,4), (3,4), (4,5), (1,2), (2,3), (2,5)$. We can easily verify that no other pair satisfies this inequality

---

# E: The Simplified Logo Compiler

Logo is a programming language that can be used to draw simple shapes on the screen. In this question, you will have to implement a simplified Logo compiler that supports to commands 'FD' and 'LOOP...END' statements for a one dimensional turtle. The program begins with the turtle (cursor) located at coordinate 0, and the command "FD $x$" ($x$ is an integer) can be used to move the turtle by $x$ units. Loop instructions consist of a line beginning with "LOOP M" (M is an integer) and end with the line "END". The commands between the LOOP and END need to be repeated M times. Given a valid logo program with these two commands, provide the final location of the turtle.

## Input

The first line consists of one integer $N$ ($N \leq 10^4$), the number of lines in the logo program. The next $N$ lines describe the program, and each line will have a maximum of 32 characters.

## Output

Print one integer $X$, the final position of the turtle.

---

```
input
2
FD 40
FD -30

output
10
```

```
input
5
FD 50
LOOP 3
FD 10
FD 25
END

output
155
```

```
input
9
FD 40
LOOP 10
FD 5
LOOP 7
FD -5
FD 7
END
FD 6
END


output
290

explanation
40 + 10 × {5 + 7 ∗ (−5 + 7) + 6}
```

---

# F: Fighting Fire With Fire

Moontech Pharmacueticals has successfully created an *anti-virus virus* that can be used to fight COVID-19. They intend to start injecting the new virus into people as soon as possible, but the astronomical cost of each dose means they want to *minimize* the number of total doses needed. The *antivirus* works like a regular virus, and can spread from one human to another, and is highly contagious. Given a population of $N$ people, and a list of $M$ *friendships* (people who will spend enough time with each other for the antivirus to spread), find the minimum number of people who need to be vaccinated to reach herd immunity (defined to be strictly greater than 80% of the population). Note that if A is a friend of B and B is a friend of C, injecting A with the antivirus will ensure that C also gets infected.

## Input

The first line of input contains the integer N (population) and M (number of friendships) ($1 \leq N, M \leq 10^5$, and individuals of the population are numbered $0 \ldots N-1$. The next $M$ lines contain two integers $u, v$ representing that $u$ is friends with $v$ and vice versa.

## Output

Print one integer, $E$, denoting the number of antivirus doses needed to infect strictly greater than 80% of the population.

---

```
input
10 8
0 1
1 8
5 7
8 0
6 9
6 7
9 5
2 3

output
3

explanation
Infecting any one of (0, 1, 8) will ensure all three of them are infected.  Similarly
for the sets (2, 3) and (5,6,7,9).  For example, we can infect 5, 1 and 2 to reach
a total of 9 people infected, which is strictly greater than 80%.
```

---

# G: Hitchcock and Scully

Hitchcock and Scully are trying to find new places to visit for lunch over the next $N$ days. Since they have only a limited amount of time in their lunch break, they make a list of $M$ restaurants in Brooklyn's $99^{th}$ precinct. Each restaurant will cost them $\$m_i$ for a lunch. Given a list of size $N$, representing the (total) amount of money they have in their pockets for each of the next $N$ days, calculate how many *options* they have for lunch each day. They can go to any restaurant they want on day j, provided $n_j \leq m_i$.

## Input

The first line of input contains space-separated $M$ ($1 \leq M \leq 10^5$) and $N$ ($1 \leq N \leq 10^5$). The second line contains $M$ space separated numbers $m_i$, representing the cost of dining at restaurant $i$. The third line contains $N$ space separated integers representing the money Hitchcock and Scully have each day. It is guaranteed that $1 \leq m_i, n_j \leq 10^9$.

## Output

Print $N$ space separated integers, representing the number of choices they have for each day.

---

input
```
10 5
8 9 6 5 4 3 23 9 10 1
24 9 12 3 1
```

output
```
10 8 9 2 1
```

explanation
```
On the first day with $24, they can
visit all ten restaurants.  On day
2, they can visit all restaurants
except the one costing $24.
```

input
```
5 3
60 40 90 45 120
13 44 90
```

output
```
0 1 4
```

explanation
```
If they have no options for a
certain day, print 0.
```

---

# H: H-Index

Given a graph of publications and citations, with each node representing a publication and each edge representing **one** citation, calculate the H-index of all the authors. We have $A$ authors (numbered $0 \ldots A-1$) and $P$ publications (numbered $0 \ldots P-1$). The value of h-index ($h$) of an author is the maximal possible value $x$, such that the number of papers ($x$) by the author have $x$ or more citations each. Each publication is written by exactly one author. There are a total of $C$ citations (edges) in the academic graph.

## Input

The first line of input contains the numbers $A$, $P$, $C$ in a space separated fashion such that $0 \leq A \leq P \leq 1000$. The second line contains $P$ numbers - the author for each of the publications. The next $C$ lines contain two space separated values indicating the publication numbers $(p_i, p_j)$ meaning that $p_i$ cited $p_j$.

## Output

Print $A$ space separated integers denoting the H-index of all of the authors.

---

```
input
3 5 7
0 0 1 1 2
1 0
2 0
3 0
4 0
2 1
3 1
1 4

output
2 0 1

explanation
We have three authors and five publications.  The number of citations for each publication
is:  4 2 0 0 1.  Author 0 has 2 publications with 4, 2 citations respectively.  So,
his H-index is 2.
```

---

# I: Okabe and the Toll Gates

The cities in Japan lie on a straight line numbered from 0 serially and adjacent cities are unit distance apart. Okabe Rintaro lives in city $U$ has been invited to give a talk on his Time Machine theory in city $V$. He plans to rent a car from his city to the destination. The road from city $U$ to $V$ has a few toll gates. Each toll gate has a gas station. All the gas stations surprisingly sell gas in fixed capacity containers (in litres). A litre of gas costs $K$ yen. The car Okabe rents runs according to the following mileage: $Z$ litres of gas lets him drive $AZ + B$ units. On his drive, Okabe plans to stop at every immediate toll gate (not anywhere in between), empty any gas currently in the tank and refill it again from the gas station at the toll gate. Help Okabe spend as minimum money as possible on the gas. It is guaranteed that cities $U$ and $V$ will always have tolls gates. Assume that the car had no gas before Okabe rented it. *Note that, the use of inbuilt qsort function cannot be made to solve this problem.*

## Input

The first line of input contains three space-separated N ($1 \leq N \leq 10^5$), M ($1 \leq M \leq N$) and L ($1 \leq L \leq 10^5$) denoting the number of cities in Japan, the number of cities that have toll gates and the number of different gas containers sold at each gas station. The next line contains five space-separated integers U ($0 \leq U \leq N - 1$), V ($U \leq V \leq N - 1$), A, B ($1 \leq A, B \leq 10^3$) and K ($1 \leq K \leq 10^3$), denoting the starting and destination city, mileage coefficients and the rate of gas respectively. The following line has M space-separated integers ($0 \leq T_i \leq N - 1$) denoting the cities that have toll gates. The last line of input contains L space-separated integers ($1 \leq C_i \leq 10^6$) denoting the various gas container sizes sold in the gas stations.

## Output

Print a single integer P, denoting the minimum money Okabe should spend on gas. If Okabe cannot make it to the city $V$ using the above strategy, print "NOT POSSIBLE".

---

```
input
100 15 7
11 92 2 3 10
52 81 76 36 5 23 50 90 17 46 3 82 11 92 83
1 8 11 7 2 5 9

output
330
```

---

# J. Okabe and Entropy

Okabe, having completed the course on Advanced Statistical Mechanics in his university, has an epiphany as to how to solve the problem of the parallel worldlines. He realizes that to transit from one worldline to another, he needs exactly $E_i$ energy (transit potential) to overcome the entropy between those two worldlines. After painstaking calculations, Okabe has finally figured out all the transit potentials. But to solve the final problem, he needs one extra information, that is the cost of the Minimum Spanning Tree across these worldlines. As he is dog-tired, he turns to you, Makise Kurisu, to help him find that cost. *Note: You can read up more about MSTs and how to find their cost here.*

## Input

The first line of input contains two space-separated integers N ($2 \leq N \leq 500$), M ($1 \leq M \leq \frac{N(N-1)}{2}$), denoting the number of worldlines and the number of transits possible between those worldines respectively. The following M lines contain three space-separated integers $U_i$, $V_i$ and $E_i$ ($0 \leq U_i, V_i \leq$ N-1, $1 \leq E_i \leq 10^5$) denoting an undirected transit between the worldlines $U_i$ and $V_i$ which has a transit potential of $E_i$. It is guaranteed that worldlines graph will be connected.

## Output

Print a single integer E, denoting the cost of the Minimum Spanning Tree of the worldlines graph.

---

input
```
7 10
0 1 1
0 2 5
2 3 1
3 4 3
4 5 4
5 0 3
6 1 3
2 6 2
3 6 2
6 4 3
6 5 2
0 6 2
```

output
```
11
```

---

# CS F211
# Data Structures and Algorithms
# Assignment - 4

Allowed languages: **C**

February 17, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

## Changelog

- Question B - testcase 2 has been corrected. The earlier answer was incorrect.
- Question C - clarification - all words shall consist of lowercase English alphabets only.
- Question D - constraints for $N$ added.
- Question F - Input format for corrected
- Question I - pseudo code corrected from $r = n - 1$ to $r = n$

# A: Sky of a Million Stars

A cave discovered on Mars appears to have a bunch of inscriptions in the wall, presumably made by an alien race. Linguists and anthropologists who are studying the images beamed back to Earth realised that they could piece together some of the information if they knew the order of that foreign alphabet. The alien race, surprisingly, uses a **subset** of the English alphabet [A-Z], but with a different alphabets order. Experts studying the inscriptions could discover $N$ rules about the alien alphabetical order. Each rule is represented as $c_1$ $c_2$, meaning letter $c_1$ comes before $c_2$ in the alien alphabet. Note that all alphabets that are a part of the alien alphabet **will definitely** appear atleast once in the rules. Print **one** valid order of the alien alphabet. If there are multiple alien alphabet orders possible, print any one. If the alien language is self-contradictory, print "ALIENS BE CRAZY" and exit.

## Input

The first line contains an integer $N$ ($1 \leq N \leq \binom{26}{2}$). The next $N$ lines contain two characters each (space-separated), representing one rule. It is guaranteed that all characters in rules are capital letters A-Z.

## Output

The order of the alien alphabet in one line, with no spaces. If no order is possible, print the string "ALIENS BE CRAZY".

---

input
```
3
X B
C X
C A
```

output
```
CAXB
```

explanation
Other possible answers are CXAB, CXBA

input
```
8
R E
A S
B A
E A
L M
A R
B X
X A
```

output
```
ALIENS BE CRAZY
```

explanation
No valid outputs possible.

---

# B: The Three Laws

*A robot may not injure a human being, or, through inaction, allow a human being to come to harm.*

In a bid to garner publicity for their new line of security and police robots, U.S. Robots is organising a RoboWars event that will be televised on international TV. The tournament has $N$ robots participating, each with strength $R_i$. When a robot with strength $R_i$ fights with a robot of strength $R_j$, the robot with higher strength wins, and destroys the weaker robot. The weaker robot is completely destroyed and eliminated from the tournament. The stronger robot (say, $R_i$) survives, but it's strength changes to $R_i(new) = abs(R_i - \alpha(R_i - R_j))$, where $0 \leq \alpha \leq 10$. If two robots are equally matched, they both die in each others hand. The tournament consists of a number of rounds, which continue till only one (or none) robots are left standing. The robots are initially placed in a straight line, and in the first round, the first robot fights the second, the third robot fights the fourth and so on. If a robot is leftover at the end of the line, and can't be paired it doesn't take part in the round, and recharges while everyone else fights, allowing it to increase its strength by $\beta$ in that round. Once the round is complete, dead robots are removed and the next round commences. Determine the position and final strength of the winner.

## Input

The first line contains three space separated integers - $N$ ($0 \leq N \leq 10^5$), $\alpha$ ($0 \leq \alpha \leq 10$) and $\beta$ ($500 \leq \beta \leq 10^3$). The next line contains $N$ space separated integers numbers, denoting the strengths of each of the robots in line. ($0 \leq R_i \leq 10^4 \; \forall i$).

## Output

Print two space separated numbers $i$ and $R_f$. $i$ is the position the final surviving robot (winner of the tournament) was standing in at the start of the tournament (the initial line is 1-indexed). $R_f$ is the strength of the winner after the final round. If no robots are left standing print $-1 - 1$.

---

```
input
5 5 30
70 75 80 80 20
output
-1 -1
explanation
After the first round, the robot
strengths are:  {50, 50} - which
leads to both of them dying in
the final round.  No robot left
standing..
```

```
input
7 8 40
70 60 50 120 100 50 40
output
4 9320
explanation
After round 1:  {10, 440, 300, 80}
After round 2:  {3000, 1460}
After round 3:  {9320}
```

---

# C: Talking to Myself

Mike likes playing a word game with himself, where he tries to transform a word, $S$ (length $M$), to a different word $E$ (length $M$). He begins by changing (replacing) one letter of the $S$, to form an intermediate word $W_1$. He then changes another letter in $W_1$ to form $W_2$. He does this to form a sequence of words $S, W_1, W_2 \ldots W_L, E$. The end word is special, since it must satisfy the property $E[i] \neq S[i]$, $\forall i$ such that $0 \leq i \leq M-1$. Additionally, he cannot arbitrarily replace any letter with another letter - all intermediate words ($W_i$) and the end word ($E$) must be real words, which means they must be in a dictionary $D$ (containing $N$ words). Determine if such an $E$ exists, and if it does find the *minimum* number of intermediate words required to get to an end word $E$ from the start word $S$. If it is impossible to do so, print $-1$. If there are multiple possible words E, choose the one that has a minimum number of intermediate steps. *Additional thinking: if you had to implement this for the entire English dictionary, would your approach work?*

## Input

The first line of the input contains integer $N$ such that $1 \leq N \leq 1000$. The second line of input contains the word $S$. The next $N$ lines contain one word each ($D_i$) such that $1 \leq len(D_i) \leq 12$ representing all words in the dictionary. All words consist of lowercase English letters only.

## Output

Output a single integer, denoting the **minimum** number of intermediate words you'd need to generate before reaching an end word.

---

input 1
9
app
xi
ben
ape
apo
xu
ave
bpo
beo
ove
output 1
2

input 2
9
rome
tome
some
hell
hall
peep
xhud
yyee
yuii
iyll
output 2
-1

explanation 1
app -> ape -> ave -> ove. ove is an end word, by definition, and we need two intermediate words to get here. app -> apo -> bpo -> beo -> ben is also a valid path, with end word "ben", but this has 3 words.

---

# D: Wikipedia

*TIME Person of the Year, 2006*

Rohan began reading a Wikipedia article $S$ initially. In each Wikipedia article, there are links to other Wikipedia articles. Being the curious lad he is, one of the links **might** interest him, causing him to click on it and read the linked article. However, he doesn't pick what article to choose out of the links at random - for each link in an article, there is a certain probability ($p_{ij}$) that Rohan will click it. There is also a probability ($q_i$) that he will not click on any link at all, and just close the browser after reading the current article. This probability $q_j$ for a certain article, can be calculated as $q_j = 1 - \sum_i p_{ij}$, where $p_{ij}$ is summed over all outgoing links on that article. Some articles might not have any outgoing links, meaning he will shut down his browser after reading that last article. Given a set of links ($E$) and probabilities associated with clicking each link, calculate, for each article, the probability that it is the last article he reads. It is guaranteed that articles ($W_i$) and edges ($E$) form a tree rooted at $S$. Rank all articles from highest to lowest probabilities of it being the last article read.

## Input

The first line has integers $N$ ($1 \leq N \leq 10^5$), $E$, and $S$. The next $E$ lines contain three space separated numbers - $(W_i, W_j, p_{ij})$ - meaning that article $W_i$ has a link to $W_j$, and there is a $p_{ij}$ probability that this link will be clicked.

## Output

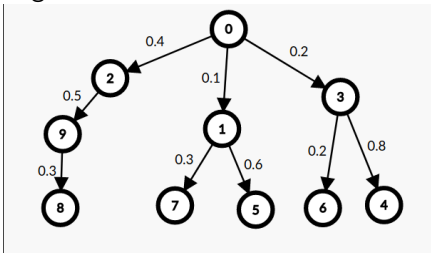
Print $N$ space separated numbers, the articles ranked in descending order of probabilities. If two articles have the same probability, order them in increasing order of article number.

---

input

```
10 9 0
0 1 0.1
0 2 0.4
0 3 0.2
1 7 0.3
1 5 0.6
2 9 0.5
9 8 0.3
3 4 0.8
3 6 0.2
```

figure




output

```
0 2 4 9 5 8 6 7 1 3
```

explanation

Probabilities for ending on articles $\{0..9\}$ are {0.3, 0.01, 0.2, 0, 0.16, 0.06, 0.04, 0.03, 0.06, 0.14} respectively. Article 0, which has the highest probability (0.3), is printed first. Articles 2, 4, 9 have the next three highest possibilities (0.2, 0.16, and 0.14). The next highest probability of 0.06 is shared by articles 5 and 8 - 5 must be printed before 8, as $5 < 8$.

# E: Connecting Nails

On a wooden plank, you have $M$ nails driven into the wood at points $S = \{(x_1, y_1) \dots (x_M, y_M))\}$, with each nail being one point. Nails are connected in this order: sweep a reference line from the X axis upwards (order of quadrants: Q1, Q2, Q3, Q4), completing a full turn while joining every subsequent point. Let $S_0$, $S_1$ be the first two vertices encountered while sweeping the line, and let $d_0$ be the Manhattan Distance between them. Calculate the sum of all pairwise Manhattan distances ($\sum_i d_i$) between all such pairs of points encountered as the reference line sweeps all four quadrants It is guaranteed that, for any two distinct points $I(x_i, y_i)$, $J(x_j, y_j)$, I, J and the origin $O(0, 0)$ are never collinear if $I$ and $J$ are in the same quadrant.

## Input

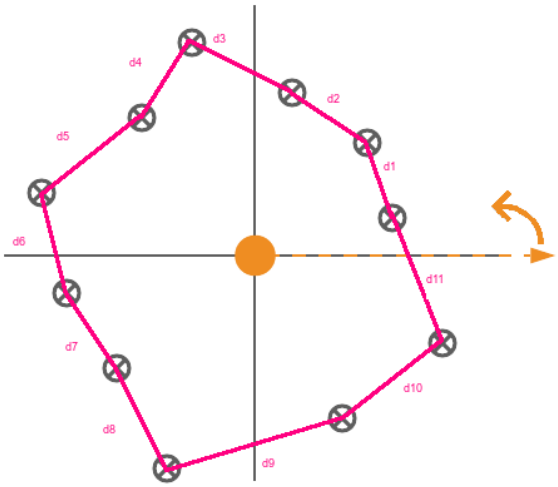
The first line contains the number $M$. The next $M$ ($3 \leq M \leq 10^5$) lines contain two space separated integers representing $x_i, y_i$ ($-1.5 * 10^5 \leq x_i, y_i \leq 1.5 * 10^5$).

## Output

Print one integer, $L$, the sum of all pairwise Manhattan distances.

---

figure




The answer to the above problem is the sum of all Manhattan distances of the points connected by red line segments, i.e. $\sum_i d_i$. Orange arrow indicates the beginning point, and the direction to rotate our reference line.

input
```
3
4120 0
-1000 9800
-3210 -6467
```

output
```
47194
```

input
```
4
-1 6
1 1
4 -3
-1 -2
```

output
```
28
```

explanation

Order of points: $(1, 1)$, $(-1, 6)$, $(-1, -2)$, $(4, -3)$, $(1, 1)$

---

# F: Polifia

Tim is a guitar player and wants to hold a concert. The venue can be represented by a cartesian co-ordinate plane. Tim has $N$ amplifiers where the $i^{th}$ amplifier has a power output of $a_i$ watts and is located at the position $(x_i, y_i)$. Each of these $N$ amplifiers has already been placed in the venue by the sound engineer and cannot be moved. There is something very peculiar about this particular venue: The people in the crowd can only stand along *some line parallel to* $y = x$. Note that this line is chosen by Tim and can be placed anywhere in the co-ordinate plane. Tim wants the audience to get an equal volume of sound from both sides so he wants to place the line such that the *sum of powers of amplifiers on both sides of the line is equal*. Help Tim find if placing such a line is possible. More formally, you must find if it is possible to choose a line $y = x + c$ where $c$ is a variable such that sum of powers of amplifiers on both sides of the line is equal.

## Input

The first line of input contains one integer $T$, the number of testcases. For each testcase, there will be $N + 2$ lines of input as described below. The first line of each testcase consists of $N$ $(1 \leq N \leq 10^5)$, the number of amplifiers. The next $N$ lines contain three integers each: $x_i$, $y_i$, and $a_i$ the co-ordinates of the $i^{th}$ amplifier $(-10^9 \leq x_i, y_i \leq 10^9)$ and the power of the $i^{th}$ amplifier $(-10^9 \leq a_i \leq 10^9)$

## Output

Print $T$ lines, with each line containing either YES or NO in all capitals. (please print in all capitals to avoid problems with Mooshak)

---

input

```
1
3
-1 1 3
-2 1 1
1 -1 4
```

output

```
YES
```

explanation

We have one testcase. Any line of the form $y = x + c \ \forall -2 < c < 2$ would divide the amplifiers such that both sides have a total power of 4 watts

---

# G: Tuul

Danny Carey has $N$ drum pads lined up in a row, where the $i^{th}$ drum pad has a frequency $a_i$. For the next song Danny needs all the drums pads to be arranged in a non-decreasing order of frequencies. In one swap, Danny can swap the positions of two adjacent drum pads. He needs to quickly tell his drum mechanic the minimum number of such swaps required to arrange the drum pads in a non-decreasing order of frequencies. Despite being a natural at polyrhythms, he was too slow at this task so he asked you for help.
Hint: Try and think of a divide and conquer strategy

## Input

The first line of input contains a single integer $N$ ($1 \leq N \leq 10^5$), the number of drum pads. The second line contains $N$ integers, where $a_i$ represents the frequency of the $i^{th}$ drum pad ($-10^9 \leq a_i \leq 10^9$).

## Output

output a single integer, the minimum number of swaps required to arrange the drums as described above.

---

```
input
5
1 20 6 4 5

output
5

explanation
first swap 20 with 6, then 20 with 4, then 20 with 5.  Then swap 6 with 4 and then
6 with 5.  With a total of 5 swaps, we get the arrangement 1 4 5 6 20
```

---

# H: Meshuguhh

Fredrik Thordendal wants to make an ironic song using only the major scale to confuse his fans. The major scale consists of 7 notes, represented by the upper case letters A to G (inclusive). He writes out a riff for this (in drop F tuning obviously) using these notes. Here, a riff is a sequence of notes where each note is a capital letter from the set of letters A to G (inclusive). The drummer Thomas noticed that the riff was way too long so he asked Frederik to choose the *smallest possible contiguous subsegment* from the riff that contains *all* the types of notes present in the riff, and use that for the song. Can you help Frederik find the length of such a subsegment? More formally, your task is to find the smallest subsegment such that the set of distinct notes in the subsegment should be equal to the set of distinct notes in the entire riff.

## Input

The first line contains a single integer $N$, the number of notes in the entire riff ($1 \leq N \leq 10^5$). The second line contains a string of characters representing the riff itself. It is guaranteed that the string only consists of upper case english letters from A to G (inclusive).

## Output

print a single integer, the length of the smallest possible contiguous subsegment that contains all the types of notes contained in the original riff

---

input
9
ABBGDCACD

output
5

explanation
We can take the subsegment from index 3 to index 7 (inclusive). This subsegment includes the notes B,G,D,C,A which contains all the notes present in the entire riff

---

# I: Melattica

Lars Ulrich isn't a very good drummer so he decided to use a Machine Learning model to write drum parts for him. But now the rest of the band likes the machine learning model better and decides to fire Lars. Now he has to use his machine learning skills in other ways. He develops a model that can check if a permutation is in increasing order or not but he's not sure if the model actually works. To check this, he decided to perform a test: he takes a number $x$ and expects it to be at position *pos* if the permutation is sorted. He then performs a binary search on the permutation and if the number $x$ is at position *pos* according to the binary search, then he says that the permutation is increasing. (Recall that a permutation of size $N$ is an array such that every number from 1 to $N$ occurs exactly once). The C code for the binary search he performs is given below

```
int l=0, r=n;
while(l<r) {
    mid = (l+r)/2;
    if(arr[mid]<=x) l = mid+1;
    else r = mid;
}
l>0 && a[l-1]==x? printf("Yes") : printf("No");
```

But what Lars doesn't know is that the permutation need not be necessarily sorted to find the number $x$ at position *pos*. Can you find how many such permutations exist such that the binary search algorithm finds $x$ at the position *pos*? Since the number can be large, print the answer modulo $10^9 + 7$
source for those interested: NSynth, using neural networks to create melodies

## Input

The first line contains three numbers, $N, x$, and *pos* ($1 \leq x \leq N \leq 1000$, $0 \leq pos \leq N - 1$).

## Output

Print a single integer, the number of possible permutations modulo $10^9 + 7$

---

input
4 1 2

output
6

explanation
the binary search works on (2,3,1,4) , (2,4,1,3), (3,2,1,4), (3,4,1,2), (4,2,1,3), (4,3,1,2)

---

# J. Zero Selects

You are giving the second round of inductions for the music club and the keyboardist plays a "arpeggio" of $N$ notes consecutively on his keyboard. You can tell from the sound that all the notes first increase to a certain point and then decrease till the end (Note that there can be 0 increasing or 0 decreasing notes). You are able to figure out the sequence of notes he plays and have stored that sequence in your memory where the $i^{th}$ note played is $a_i$. The keyboardist then asks you $q$ questions, where in the $i^{th}$ question he asks if he played the note $t_i$ **anywhere** in the sequence. You have to correctly answer YES or NO for all queries or you'll have to wait another semester to get into the club! To help you out with this refer to this link on ternary search

## Input

The first line contains 2 integers $N$ and $q$ ($1 \leq N, q \leq 10^5$), the number of notes played and the number of questions asked. The second line contains $N$ integers, representing the notes played. It is guaranteed that the notes first increase and then decrease as described above. The next $q$ lines contain one integer each: $t_i$ where $t_i$ represents the $i^{th}$ question ($0 \leq t_i \leq 10^9$).

## Output

output $q$ lines, each containing a single word YES or NO in all capitals. (Please use all capitals to avoid problems with mooshak)

---

```
input
6 4
1 3 7 5 4 2
2
5
6
8

output
YES
YES
NO
NO
```

---

# CS F211
# Data Structures and Algorithms
# Assignment - 5

Allowed languages: **C**

February 17, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# A: OS Scheduler

One of the most important tasks of an operating system is deciding which tasks gets to use the CPUs. Let us say that there are $T$ tasks in the waiting list that require the CPU's computational power at time $t = 0$. No additional tasks will be added to the wait list of tasks. Each task requires some CPU time, $E_i$, for completion and has a priority, $P_i$ (ranging from $0 - 10$, with 0 having the highest priority and 10 the least priority). The scheduling algorithm must work by (1) selecting *important tasks* first (as decided by $P_i$) and (2) in case of equal preference select the *shortest job first* (least time required). What are the indices of the first $k$ tasks the scheduling algorithm prioritises? *Note: Your solution* ***must*** *be in* $o(T + k \cdot log(T))$.

## Input

The first line contains two integers $T$ ($1 \leq T \leq 10^5$) and $k$ ($1 \leq k \leq 10^2$). The next $N$ lines contain two space separated integers $E_i$ ($0 \leq E_i \leq 10^9$) and $P_i$ of the tasks (tasks are zero-indexed).

## Output

Print $k$ space separated integers representing the indices of first $k$ tasks that will be scheduled by the OS.

---

```
input
8 3
34 3
2167 5
10 0
23 0
325 3
45 6
646 7
353 7

output
2 3 0

explanation
Schedule all tasks with priority 0 and then schedule the 0th task (priority 3).
```

---

# B: Altered Carbon

In the Bay City Metropolis Hospital, $N$ patients are waiting to be treated outside the hospital today. Being the dystopian society it is, the hospital wishes to *maximise* it's earning potential today. Each patient $i$ has a certain *net wealth* $C_i$ that is stored in an encrypted form in alphanumeric string $D_i$ in their cortical stacks (an implant placed in the neck). The net worth $C_i$ is the sum of all of the *alphabet-separated numbers* in the string. For example, the string "as3442jhs2323kaj22kjyu3yhdjhshdkjrhsjk211df" contains five numbers in it, and the net worth for this string would be the sum of these five numbers, i.e. $sum(3442, 2323, 22, 3, 211) = 6001$. Given that the hospital can admit only $k$ ($1 \leq k << N$) patients, and wants to admit individuals with the highest net-worths, which ones should it select? What is the sum of these net worths? Since this number might be very large print it modulo $10^9 + 7$.

## Input

The first line contains two space separated integers: $N$ ($0 \leq N \leq 10^5$) and $k$ ($0 \leq k \leq 100$). The next $N$ lines contain one alphanumeric string each (of length $\leq 150$) with each line containing one $D_i$.

## Output

Print one integer representing the moduloed sums of net worths ($(\sum_i C_i)\%(10^9+7)$) of the selected individuals admitted in the hospital. Each string might contain alphabets (upper/lower case) and digits. It is guaranteed that each number inside the string will have at most 10 digits.

---

input
```
3 2
a21b45c90
a42b1d3FG6
98nahsg8
```

output
```
262
```

explanation
```
Person 0:  21 + 45 + 90 = 156
Person 1:  42 + 1 + 3 + 6 = 52
Person 2:  98 + 8 = 106
Select persons 0 and 2
```

input
```
5 3
a1ghe2
BGH54jk78
g3267372673hh7
jhj666SHH7
apes123strong23together9
```

output
```
267373487
```

explanation
```
The net worths are (2, 132,
3267372680, 673, 155).  The answer
is (3267372680 + 673 + 155)%(10^9 + 7).
```

---

# C: Largest Number

Given two numbers $N_1$, $N_2$ (as digit strings) find the *largest* number you can create by following these steps. To merge the numbers:

1. start with an empty string $M$.
2. At each step you can choose to:
   (a) append the first digit of $N_1$ to the back of M and delete the first digit in $N_1$ (or)
   (b) append the first digit of $N_2$ to the back of M and delete the first digit in $N_2$
3. Keep repeating this process till both $N_1$ and $N_2$ are empty

What is the largest possible number you can create by merging $N_1$ and $N_2$.

## Input

There will be two lines of input, with each line containing one very large number ($\leq$ 200 digits).

## Output

Output a single line, containing the maximal merged number.

---

input
7756453241
82715243

output
877564532724152431

---

input
965
453

output
965453

---

input
775
769

output
777695

---

# D: Valentine's Day

As an an unscrupulous store owner on a college campus, you decide to sell chocolates on Valentine's Eve in a bid to earn more money. Your supplier has $M$ different kinds of chocolates all of which have different qualities ($Q_i$), and you want to select *exactly two* varieties of chocolates to stock. Since you intend to sell chocolates at a grossly overpriced sales price, you decide that you want to *maximise* the *difference* in qualities of the two selected chocolates - to make it *seem* like the chocolate of higher quality is worth the price relative to the lesser, even when both of them aren't worth it. Given a list of qualities of each type of chocolates, select the types that *maximise* this difference of qualities of chocolates. What is this maximal difference? And how many ways can you achieve this?

## Input

The first line has the integer $M$ ($2 \leq N \leq 10^6$). The next line has $M$ space separated integers containing the *quality* of each type of chocolate ($0 \leq Q_i \leq 10^9$).

## Output

Print two space-separated integers $X$ and $Y$. $X$ is the the largest possible difference of qualities of the two selected chocolate types. $Y$ is the number of *ways* this maximum can be achieved.

---

```
input
2
9 278

output
269 1
```

---

```
input
3
19 23 24

output
5 1
```

---

```
input
5
24 15 19 15 24

output
9 4

explanation
There are 4 sets of indices that get the maxmimal difference of 9:  (0, 1),
(1, 4), (0, 3), (4, 0)
```

# E: A Foreshadowing of Array Operations

You are given an array $a$ of size $N$ in non-decreasing order. In one operation you can choose any two adjacent elements and insert the floor of their average between them. This operation would increase the length of the array by 1. Let the maximum difference between any pair adjacent numbers in the array be $m$. Using at most $k$ operations, find the minimum possible value of $m$ that can be achieved after these operations.

*As an extra challenge (will not be asked in the lab), try solving this if you can insert any integer such that the array remains non-decreasing.*

## Input

The first line contains two integers $N$ and $k$ ($1 \leq N, k \leq 10^5$). The second line contains $N$ integers representing the array $a$ ($1 \leq a_i \leq 10^9$).

## Output

Print one integer, the minimum possible maximum difference between any two adjacent integers after all operations.

---

```
input
5 2
10 13 15 16 17

output
2

explanation
you can add up to 2 numbers.  Adding 11 and 14 in the appropriate positions would give
a maximum difference as 2.  It is not possible to do better in this situation
```

---

# F: The Maxim

Migi the parasite is in a human host and trying to take over. To completely take over the host it must first take over all its organs which are numbered from 1 to $N$. You are given the pairs of organs that are connected to each other. Initially Migi is at organ 1. Migi can only take over a particular organ if it is not already occupied and it is connected to an occupied organ. The human host is fighting back pretty hard so Migi decides to do the following: **at every step choose the smallest numbered organ that is currently unoccupied and is connected to an occupied organ. Occupy the chosen organ**. Migi doesn't have much time so he asks you to tell him the exact order of organs to occupy including the starting organ.

## Input

The first line consists of two integers $N, M$ ($1 \leq N, M \leq 10^5$). The next $M$ lines contain two integers $u_i$ and $v_i$, indicating that organ $u_i$ and organ $v_i$ are connected ($1 \leq u_i, v_i \leq N$). It is guaranteed that all the organs form a connected graph.

## Output

Print a single line consisting of $N$ integers representing the order of organs Migi occupies (including the starting organ).

---

input

```
5 4
1 4
3 4
2 3
1 5
```

output

```
1 4 3 2 5
```

explanation

Initially Migi is at organ 1. Out of the connected organs 4 and 5, we choose 4 as it is the smallest. Now, 4 is connected to 3 and 1 is connected to 5, out of which 3 is the smallest so he occupies that. Now the remaining possible nodes are 2 and 5 so Migi goes to 2 first and then 5.

---

# G: Array Operations

You are given an array of size $N$ where the $i^{th}$ element is $a_i$. In one operation you will remove the smallest and the largest elements in the array and insert the absolute value of their difference into the end of the array. If multiple elements are smallest or largest pick any one. You are given $q$ queries, each of which consist of a single number $k_i$. For every $i$ from 1 to $q$, you have to find the sum of elements after performing $k_i$ operations on the **initial array** $a$.

## Input

The first line contains two integers $N$ and $q$ ($1 \leq N, q \leq 10^5$). The second line contains $N$ integers, representing the array $a$. The next $q$ lines contain one integer each, representing the query $k_i$ ($1 \leq k_i \leq N$),

## Output

Output $q$ lines, each line containing one integer representing the answer for the $i^{th}$ query.

---

input
```
5 2
3 2 1 4 5
1
2
```

output
```
13
9
```

explanation

after the first operation, the array becomes 3 2 4 4. After the second operation the array becomes 3 4 2.

---

# H: More Array Operations

After solving the previous problem, you wanted to challenge yourself some more so you decided to try a slightly different problem. You are given an array of size $N$ where the $i^{th}$ element is $a_i$. You have a new array $b$ which is initially empty. In one operation you remove the leftmost element in $a$ and add it somewhere in $b$. You do this $N$ times until the array $a$ is completely empty. After each step you have to print the median of all the elements in $b$. Median is defined in the following way: Let $M$ be the size of array $b$. If $M$ is even, then median is $b[\frac{M}{2}]$. Otherwise, the median is $\left\lfloor \frac{b[\frac{M-1}{2}]+b[\frac{M+1}{2}]}{2} \right\rfloor$. *Hint: Could you use heaps here?*

## Input

The first line contains a single integer $N$ representing the size of the array $a$. The second line contains $N$ integers where the $i^{th}$ integer is $a_i$.

## Output

output $N$ space separated integers where the $i^{th}$ integer is the median of $b$ after the $i^{th}$ operation

---

input
4
5 15 1 3

output
5 10 5 4

explanation
first operation $\rightarrow b = [5]$. Second operation $\rightarrow b = [5, 15]$ so median is 10. Third move $b = [5, 15, 1]$ and here median is 5. In the final operation $b = [51513]$ where median is 4

---

# I: Chocolates

Upon his return from the States, your father has brought you a lot of chocolates. On the first day, he arranges N bowls and adds a few chocolates to each bowl. Everyday (starting from day one), you decide to eat the chocolates from the bowls and select one bowl at random and eat *all* X chocolates (assuming that the bowl has X chocolates) from it. The next day, your dad replenishes the bowl (from which you ate the chocolates) with $\lfloor X/3 \rfloor$ chocolates. You now start thinking about the *maximum* number of chocolates you can eat in a span of few days and decide to write a simple program that will calculate the same for you.

## Input

The first line of input contains two space-separated integers N ($1 \leq N \leq 10^5$) and D ($1 \leq D \leq 10^5$) denoting the number of bowls and the number of days for which you want to make the calculation respectively. The following line contains N space-separated integers ($0 \leq A_i \leq 10^9$) denoting the number of chocolates put in each of bowls (the $i^{th}$ integer denotes the number of chocolates put in the $i^{th}$ bowl). Assume the bowls are numbered serially from 1 onwards.

## Output

Print a single integer Y, denoting the maximum number of chocolates you can eat by the end of D days. As the number can be large, print it to modulo $10^9 + 7$.

---

```
input
5 7
4 16 6 27 8

output
75

explanation
It can be observed that you eat 27 + 16 + 9 + 8 + 6 + 5 + 4 = 75 chocolates in 7 days.
```

---

# J. Dijkstra

Dijkstra's Single Source Shortest Path Algorithm is perhaps the most used procedure to find shortest paths in a weighted graph. The algorithm is greedy in nature and is optimally done with a min-priority queue (a heap). The pseudo-code for it is as follows:

1. Mark source vertex with 0 cost initially and rest all vertices with a cost of $\infty$.
2. With the cost of the source vertex, relax the cost of all its neighbours, i.e. `cost(v) = min(cost(`$v$`), cost(`$u$`) + W(`$u$`, `$v$`))` $\forall$ vertices $v$ that are neighbors of source vertex $u$.
3. Mark $u$ as visited and from those relaxed, choose the minimum cost vertex and make it the source vertex.
4. Go to step 2 and repeat till all vertices are visited.

Now given a weighted undirected connected graph G(V, E) and a source vertex $s \in$ G(V, E), your task is to find the shortest path from $s$ to all the remaining vertices in the graph. *Additional: Can you guess the time complexity of Dijkstra using heaps?*

## Input

The first line contains three space-separated integers N ($3 \leq$ N $\leq 500$), M (N-1 $\leq$ M $\leq \frac{N(N-1)}{2}$) and S ($0 \leq$ S $\leq$ N-1) denoting the number of vertices, edges and source vertex respectively. The following M lines contain three space separated integers $U_i$, $V_i$ and $W_i$ ($0 \leq U_i, V_i \leq$ N-1, $0 \leq W_i \leq 10^7$) denoting an edge of weight $W_i$ between vertices $U_i$ and $V_i$.

## Output

Print a sequence of N space-separated integers where the $i^{th}$ integer denotes the shortest path distance to the $i^{th}$ vertex from the given source vertex.

---

```
input
7 13 3
5 6 10
0 5 5
0 1 2
0 2 2
2 3 8
1 4 8
1 3 7
2 4 5
4 3 7
3 5 8
4 5 1
4 6 1
2 6 2
```

```
output
9 7 8 0 7 8 8

explanation
It can be observed by running
the algorithm that we get the
distances as mentioned above for
the vertices 0 to 6
```

---

# CS F211
# Data Structures and Algorithms
# Assignment - 6

Allowed languages: **C**

March 4, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# A: Disposition

Bob likes to watch the weather change so he wants to know today's forecast. He has a list of $N$ different numbers representing $a_i$, the air pressure at various times of the day. Bob can tell the weather using only these numbers, but to do so he first needs to sort them in non-decreasing order based on the value of $a_i \mod k$, given $k$. For example, if $k = 5$ then the number 11 would come before 7 in the sorted array because $11 \mod 5$ is less than $7 \mod 5$. If the value of $a_i \mod k$ is equal to the value of $a_j \mod k$ for some $i$ and $j$, then they must be sorted based on the values $a_i$ and $a_j$ (That is if $a_i < a_j$ then $a_i$ comes before $a_j$). Sort the array in this way for Bob.

## Input

The first line contains two integers $N$ and $k$ as described above ($1 \leq N \leq 10^5$, $1 \leq k \leq 10^5$). The second line of input contains $N$ integers where the $i^{th}$ number represents $a_i$ ($1 \leq a_i \leq 10^9$).

## Output

Print $N$ integers in the sorted order as described in the question.

---

```
input
5 2
1 2 3 4 5

output
2 4 1 3 5

explanation
2 and 4 must come before the rest since they have a value 0 mod 2 and 1, 3, and 5 have
a value 1 mod 2. Now 2 and 4 must be ordered based on their values and the same goes
for 1, 3, and 5.
```

---

# B: Reflection

It soon becomes night and Bob likes to talk to the moon before going to sleep. The moon tells him that the light is not her own but that of a thousand reflections passing over her. Bob noticed a row of $N$ pine trees that was illuminated by the light, the $i^{th}$ pine tree having a height $h_i$. He decided that he wants them to be sorted in non-decreasing order. To do this, he can divide the entire array of trees into contiguous segments such that each tree belongs to exactly one segment. He can then sort all the elements within each segment in non-decreasing order independently from the other segments. He must now choose segments such that after performing this operation, the entire array of trees is sorted in non-decreasing order. Bob doesn't want to waste energy on sorting so he wants to choose segments such that the number of segments chosen is as large as possible. Can you help Bob out?

## Input

The first line contains an integer $N$ representing the number of trees ($1 \leq N \leq 10^5$). The next line contains $N$ numbers representing the height of the $i^{th}$ tree $h_i$ ($1 \leq h_i \leq 10^9$).

## Output

Print a single integer, the largest possible number of segments you can choose satisfying the above conditions.

---

input
4
2 1 3 2

output
2

explanation
Here the optimal groups would be positions $1, 2$ in the first group and $3, 4$ in the second group. We can easily see that if we sort each segment individually the entire array will also be sorted.

---

# C: Triad

Since this is the finale of Bob's trilogy, he decides to keep quiet and let the work be done by a bunch of machines. Bob is actually a winemaker, but he's a little weird and uses watermelons instead of grapes. He plans to use these machines to crush watermelons to use in his wine. There are $N$ machines where the $i^{th}$ machine takes time $t_i$ to crush one watermelon. These machines can work simultaneously and independent of each other but one machine can only be crushing one watermelon at a time. You can assign a machine to begin crushing a watermelon at any time as long as the aforementioned rule is not violated. You can also use a machine more than once. Can you help Bob find the shortest amount of time needed to completely crush $k$ watermelons?

## Input

The first input line has two integers $N$ ($1 \leq N \leq 10^5$) and $k$ ($1 \leq k \leq 10^9$): the number of machines and watermelons. The next line has $N$ integers where $t_i$ ($1 \leq t_i \leq 10^9$) represents the time needed for the $i^{th}$ machine to crush one watermelon.

## Output

Output a single integer representing the shortest time needed to completely crush $k$ watermelons.

---

```
input
3 7
3 2 5

output
8


explanation
The optimal way in this case would be for machine 1 to crush two watermelons, machine
2 to crush 4 watermelons and machine 3 to crush 1 watermelon.  This would take a total
time of 8
```

---

# D: ANC Orders

The ANC burger stall, which serves only one dish - Jalapeño Poppers, is open for $N$ minutes every night. **Each minute**, $S_i$ tired and hungry students return from the library and attempt to order a plate of Jalapeño Poppers. To order food, the ANC employee takes each student's ID card and scans it with a barcode reader. The barcode scanner needs to be temporarily restarted in the $r_1^{th}, r_2^{th} ... r_k^{th}$ minute, and all students who arrive in these minutes will immediately leave (and will not come back) when they realise the barcode machine isn't working. The owner has a backup barcode scanner that works perfectly for exactly $X$ minutes, without requiring these maintenance cycles, and then shut down permanently. What would be the optimal time to activate this backup device, if the owner wants to serve as many students as possible?

## Input

The first line contains two integers $N$, $k$ and $X$ ($1 \leq k, X \leq N \leq 5 \times 10^6$), where $N$ is the number of minutes the shop remains open and $k$ is the number of minutes the barcode scanner doesn't work. The next line contains $N$ space separated integers $S_0, S_1, \ldots S_{N-1}$ denoting the number of students arriving in the zeroeth, first ... $(N-1)^{th}$ minute ($1 \leq S_i \leq 100$). The third line of input contains $k$ (sorted) space separated integers $r_1, r_2 ... r_k$ meaning that the barcode scanner doesn't work in these $k$ minutes. Note that time here is 0-indexed.

## Output

Print two space separated integers - (1) how many students the shop serves without the extra barcode machine and (2) the *maximal* number of students the shop could serve if it optimally uses the second barcode machine.

---

```
input
5 2 1
1 0 1 2 1
0 3

output
2 4

explanation
Since the backup barcode scanner can work for one minute exactly, we can choose to
use it at time, t = 0 or t = 3.  We choose t = 3 since more students are coming in
then.
```

---

# E: Anomaly Detection

Anomalies are points that don't fit the "trend" of the rest of the data. Anomaly detection is the identification of rare items and points like these, representing that differ from the majority of the data by a lot. Rahul, a Data Scient PS2 intern at Macrobook, is attempting to implement a simple anomaly detection algorithm developed by research scientists at the company. Since Rahul has never written C code before, he approaches you for help. The algorithm is explained as follows:

You are given $N$ data points each consisting of coordinates $(x_i, y_i)$, where both coordinates are not integers (guaranteed to be floats). It is guaranteed that all $N$ points will lie inside the square formed by the points (0, 0), (C,0), (A,D), (0,D), where C and D are two positive integers. We divide the X-axis into C intervals (or buckets) - namely x=0 to x=1, x=1 to x=2 and so on till x = C-1 to x = C. Similarly, the Y-axis is divided into D equally spaced buckets of unit size. These intervals form a 2D grid in the plane, with each point being located inside one X bucket and one Y bucket each - let us call these $B(x_i)$ and $B(y_i)$ respectively. The algorithm then assigns a normality score to each point $N_i = count(B(x_i)) * count(B(y_i))$, where $count(B)$ represents the number of points in that specific bucket. The $k$ points with the least normality scores are anomalies and should be outputted. If there is a tie in scores, resolve them by outputting the one with lesser index in the original array.

## Input

The first line consists of four space separated integers N ($1 \leq N \leq 10^7$) , C, D ($1 \leq C, D \leq 10^3$), k ($1 \leq k \leq N$). The next $N$ lines contain two space separated floats each representing $x_i$ and $y_i$ ($0 < x_i < C, 0 < y_i < D$) representing the $i^{th}$ point, with $i$ being zero-indexed.

## Output

Print $k$ space separated integers, representing the indices of the $k$ anomalous points in non- order of normality scores.

---

```
input
4 10 10 2
3.1 3.2
3.3 7.7
9.1 3.2
0.1 2.9

output
3 1

explanation
The normality scores of the four points are 4, 2, 2, 1 respectively.  The least score
is of index 3's so we output that.  For the next highest we notice a tie between the
1st and 2nd index elements.  We output the lesser of these indices, i.e.  1
```

---

# F: Array Operations (Again)

You are initially given an array $A$ and a set of $T$ instructions that you must evaluate sequentially, in order. Each instruction $T$ is a number $T_i$ - which means that you should rotate the array left by $T_i$ places. After executing each instruction, print the first and last element of the resulting array.

## Input

The first line consists of two space separated integers, $N$ and $T$ ($2 \leq T \leq 10^2$), where $N$ ($0 < N < 10^4$) is the size of the array $A$ ($0 \leq A_i \leq 10^9$). The second line contains $N$ space separated integers, the elements of $A$. The third line of input contains $T$ space separated integers representing the instructions ($0 \leq T_i \leq N$).

## Output

Output $T$ lines, each line containing two integers representing the first and last elements of the array after executing $i^{th}$ query.

---

input

```
5 2
3 2 1 4 5
1 2
```

output

```
2 3
4 1
```

explanation

after the first operation, the array becomes 2 1 4 5 3. After the second operation the array becomes 4 5 3 2 1.

---

# G: Maximal Descendent Distance

You are given a binary tree, consisting of **positive integer values** only, and each node is guaranteed to have a **unique** value. The binary tree is represented as an array $A$ of size $N$. $A[0]$ is the root of the tree and for a given node $i$, $(2i+1)$ and $(21+2)$ represent the left and right child respectively. If $A[i] = -1$ for any $i$, that means that the node $i$ does not exist. Given two numbers $a$ and $b$:

- find the location of both of these in the binary tree. If any one or both of these keys don't exist, print "-1".
- Identify two nodes $a'$ and $b'$, which are **leaf nodes** and are descendants of $a$ and $b$ respectively, such that the *hamming distance between values stored in nodes $a'$ and $b'$ respectively is minimized.*
- Print one integer, representing this minimal hamming distance.

*Note: The hamming distance between two integers is defined as the number of bits that are different in the binary representations of the number. eg. Hamming distance between 3 (011) and 4 (100) is 3.*

*Note 2: Every node is considered to be a descendent of itself.*

## Input

The first line consists of three space separated integer $N$ $(1 \leq N \leq 10^4)$, $a$, $b$. The next line consists of $N$ space separated integers $(A_i)$ representing the value of each node in the binary tree $(1 \leq A_i \leq 10^5$, or $A_i = -1)$.

## Output

Output one integer, the minimum hamming distance, as described in the problem.

---

```
input
7 2 3
1 2 3 -1 -1 4 5

output
2

explanation
The only possible leaf-descendent of 2 is 2.  The possible leaf-descendants of 3 are
4 and 5.  Hamming distance of (2, 4) = 2 and hamming distance of (2, 5) = 3.  Minimal
of these distances is the answer.
```

---

# H: Special String Activities

You are given a string consisting of alphabets and brackets "()[]". It is guaranteed that the brackets are balanced. You need to process the string by:

- Reverse substrings that are located within "()". For eg. $(aksj)$ becomes $jska$ after processing. (brackets removed).
- Increment all characters located within "[]". For eg. $[abcd]$ becomes $bcde$ after processing (brackets removed). The character $z$ would become $a$ after incrementing.
- Process the brackets from innermost bracket to outermost bracket.

Print the processed string.

## Input

The first line contains a string as described in the question above. It is guaranteed that the length of the string $\leq 10^4$.

## Output

Output one line, consisting of only lowercase English alphabets, denoting the processed string.

---

```
input
abcdegfg
output
abcdegfg

input
a(bc)deg[fg]
output
acbdeggh

input
a(bc[de]gf)g
output
afgfecbg

input
a[[[b]]]bujshs((dg))
output
aebujshsdg
```

---

# I: All Might vs The League of Villains

The league of villains was finally able to corner the great hero All Might. Underestimating his valor, they all decide to charge at him at once. As All Might's sidekick, you (Dave) see the locations of all the villains and All Might as points (with X and Y coordinates) on your radar. You know that the bottom-most point (the minimum y-coordinate) on the radar corresponds to that of All Might. You recommend to fend off the villains in a in a single counter clockwise sweep (see sample case for clarity). Help All Might defend himself against the league of villains in a counter clockwise manner. If more than one villain has the same orientation with respect to All Might, you suggest All Might to first fend off the villain who is closer to him (in terms of manhattan distance). Assume that All Might does not move from his point.

## Input

The first line of input contains a single integer N ($2 \leq N \leq 10^5$) denoting the number of points on your radar. Each of the following N lines contain three space-separated integers P, $X_P$, $Y_P$ ($1 \leq P \leq N$, $-10^9 \leq X_P$ , $Y_P \leq 10^9$) denoting an index number and its corresponding coordinates as you see on your radar. It is guaranteed the coordinates given will result in a unique bottom-most point.

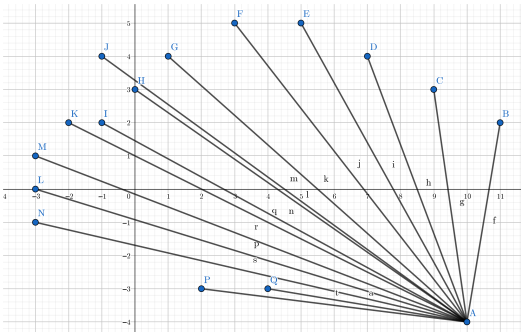
## Output

Print a sequence of N-1 integers (B), where $B_i$ denotes the index of the villain he fends off in the $i^{th}$ instant (in counter-clockwise direction). Figure for sample testcase-1 is given below.

---

input

```
16
12 -3 -1
15 5 5
13 2 -3
14 3 5
1 11 2
2 -1 4
10 -2 2
5 7 4
4 9 3
7 -3 1
6 -3 0
8 10 -4
9 -1 2
16 1 4
3 4 -3
11 0 3
```




output

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

# J. Cube-Root

In this problem, you need to find the integral part of the cube-root of a given integer without using any inbuilt math functions. Further, your solution must run in o(N) time.

## Input

The only line of input contains a single integer N ($-10^9 \leq N \leq 10^9$) for which the cube-root has to be computed.

## Output

Print a single integer X denoting the integral part of the cube-root of the given integer N.

---

input
-125

output
-5

---

input
2348247

output
132

---

input
-617491

output
-85

---

# CS F211
# Data Structures and Algorithms
# Assignment - 7

Allowed languages: **C**

March 12, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# A: Another Amulet

Nathaniel has given Bartimaues another arduous task that involves breaking into the house of Simon Lovelace. He can't transform to sneak in otherwise Faquarl will notice, so the only way to enter is to break the security code to his house. The security code gives you $N$ numbers in the form of an array and to unlock it you must do the following: For each element in the array, output the index of the closest element on the left that is smaller than it. If no such element exists (that is, there is no element smaller than the current element to the left of the current element) then output $-1$. Note that the output must be 1-indexed. *Hint: See if you can use a stack to solve this in linear time.*

## Input

The first line contains a single integer $N$ as described above ($1 \leq N \leq 10^5$). The second line of input contains $N$ integers where the $i^{th}$ number represents $a_i$ ($1 \leq a_i \leq 10^9$).

## Output

Print $N$ space separated integers as described above.

---

```
input
7
2 5 3 7 8 1 9

output
-1 1 1 3 4 -1 6

explanation
for the first element, there is nothing to the left smaller than it, so the answer
for it is −1.  For the second element, the first element that is smaller than it is
in the first position, so the answer is 1.  For the third element, the first element
that is smaller than it to the left is also at position 1 so the answer for that is
also 1 and so on...
```

---

# B: The Eye

There's another Golem loose in London and Nathaniel has to take care of it since he's the head of Internal Affairs. Upon arriving at the scene, he sees that the Golem has smashed through a row of shops simply by breaking through the walls. Peculiarly, the heights of the holes in the walls are not the same. Suppose that there are $N$ walls in a row and the distance between each wall is 1 unit. Also assume that the width of each hole is 1 unit. The height of the hole in the $i^{th}$ wall is $a_i$. To estimate the dimensions of the Golem (which is more or less cuboidal in shape), Nathaniel needs to find the volume of the largest possible cuboid that can fit in this entire setup. More formally, find the volume of the largest cuboid (by volume) that can fit in any contiguous segment of these holes without exceeding the dimensions of the smallest hole in this subsegment. The holes are rectangular and all holes start from the ground level. *Hint: Try using the idea from the previous problem here*

## Input

The first line contains an integer $N$ representing the number of holes ($1 \leq N \leq 10^5$). The next line contains $N$ numbers representing the height of the $i^{th}$ hole $h_i$ ($1 \leq a_i \leq 10^9$).

## Output

Print a single integer, the volume of the largest possible cuboid (by volume) that can fit in the given set of holes

---

```
input
7
6 2 5 4 5 1 6

output
12


explanation
Here it is best to fit a cuboid of length 3 and height 4 on the indices 3, 4, 5 (1-indexed).
We can clearly see that it fits in this subsegment and it's total volume is 12.  It
is not possible to get a higher volume in this test case.
```

---

# C: Nouda's Minions

Now that Nathaniel and Bartimaeus have joined forces (literally), they have to work together to eradicate all the demons running around St. James Park in London. The park can be represented by a grid where each cell of the grid is occupied by either a 1 or a 0. 1 represents a demon in that cell and 0 represents a human. They have one burst of energy left in the Staff of Gladstone. A shot from this staff will eliminate all living things within some rectangle on the grid (you can choose the rectangle in any way, and a rectangle consists of multiple cells). Bartimaeus and Nathaniel want to kill as many demons as possible with this last shot so they need you to find the rectangle that contains the largest number of 1s and does not contain even a single 0. *Hint: Try using the idea from the previous problem here*

## Input

The first input line has two integers $N$ ($1 \leq N \leq 10^3$) and $M$ ($1 \leq M \leq 10^3$): the dimensions of the park The next $N$ lines contain $M$ integers each, all either 0 or 1.

## Output

Output a single integer representing the largest number of 1s in a rectangle that does not contain any 0s.

---

```
input
3 3
1 1 0
1 1 1
1 1 0

output
6


explanation
Here the largest such rectangle is from (1,1),(1,2),(2,1),(2,2),(3,1),(3,2) (1-indexed).
Any larger rectangle would contain 0s which is not allowed.
```

---

# D: Getting Rid of the Ring

Bartimaeus is trying to reach the ocean to throw away the ring before Ammet catches him. The ocean is too far away so he decides to throw it in a nearby lake instead. Let the Land be represented by a grid with 1s and 0s, where a 1 means that there's a lake in that cell and 0 is just plain land. Bartimaeus is a little confused about where he is right now so he asks you to find the nearest manhattan distance to **any** lake on the grid *for every point on the grid.*
*Take a look at the following concept to help you out*: Multi Source BFS

## Input

The first line contains two integers $N$ and $M$ ($1 \leq N, M \leq 10^5$). The next $N$ line contain $M$ integers each, all the of them being either 1 or 0.

## Output

Print $N$ lines of $M$ integers each, the manhattan distance of every cell to the nearest lake. If the cell itself is a lake then the nearest distance is 0.

---

```
input
4 4
0 0 0 1
0 1 0 0
0 0 1 1
0 0 0 0

output
2 1 1 0
1 0 1 1
2 1 0 0
3 2 1 1

explanation
for the cell (1,1)(1-indexed), the nearest lake is at cell (2,2).  For the cell (1,3)
the nearest lake is at the position (1,4), and so on.
```

---

# E: The Masked Biter

You are initially given an empty multiset of integers (Remember that a multiset is a collection of numbers without any particular ordering that may or may not have duplicate values. There are three types of operations that are performed:

- add $x$: this inserts the number $x$ into the multiset
- delete $x$: this removes exactly one occurrence of the element $x$ (it is guaranteed that at least one element exists in the multiset with the value $x$ before performing this operation).
- query $x$: You have to output the number of elements in the multiset that are **parity equivalent** with $x$.

To determine whether two numbers are parity equivalent or not, first make the number of digits of both numbers equal by appending zeroes to the left of the smaller number. Once they have an equal number of digits, they are parity equivalent iff the parity (whether the digit is even or odd) of every digit of one number is the same parity as the corresponding digit in the other number.

## Input

The first line contains a single integer $N$ representing the number of operations performed ($1 \leq N \leq 10^5$). The next $N$ lines each contain a string representing the type of operation and a number $x$ representing the number used in that operation ($1 \leq x \leq 10^9$).

## Output

For every "query" operation, output the answer for that query in a single line.

---

```
input
6
add 241
query 1
add 361
delete 241
query 2301

output
1
1

explanation
For the first query, the integer in the multiset at that time is 241.  Since the value
in the query matches it, the answer is 1.  For the second query, the only number that
matches the query is 361.  Hence the answer is 1
```

---

# F: Another String Game

Alice & Bob are having a friendly competition to see who knows more words. In this turn-based game, Alice begins by saying out loud a word, $S$. Bob responds by saying another word that begins with the last letter of the previous word. They continue this process in an alternating fashion till one of them is unable to come up with a new word, which is when the game ends. Given a sequence of words (one indexed) that they used in the game, you need to determine the winner. Rules:

- From the 2nd word onwards, each word *must* be unique, i.e. not used before. If anyone uses a word that's already been used before, they lose. Additionally, the $(i+1)^{th}$ word *must* begin with the last letter of the $i^{th}$ word ($\forall i \geq 2$) - otherwise, the person who said the $(i+1)^{th}$ word loses, and further words do not need to be checked.
- Words are one indexed, and the starting word (the first word) $S$ can be any word. If all words are valid, the person who said the last word wins. Find out who wins each game.

*Hint: Look up tries.*

## Input

The first line contains $T$ ($0 \leq T \leq 10$), the number of testcases. Each testcase begins with a line that contains one integer $N$ ($1 \leq N \leq 5 \times 10^5$), the number of words played in total before the game ends. The next line contains $N$ space-separated strings, the list of words used. (odd items in this list correspond to words Alice said, and even items correspond to Bob's words). All strings will be of length not greater than 10 and contain only English lowercase letters.

## Output

Output $T$ space separated strings, each of which is the winner of that round - "Alice" or "Bob".

---

```
input
2
3
apple orange enemy
6
app potato open nose even night
2
yay yay
output
Alice Bob Alice
explanation
Alice wins in the first game since the word "orange", which Bob uses is not a valid
word.  Bob wins in game 2 since he has the last word.  Alice wins in round 3 since
Bob reuses the word "yay".
```

---

# G: Linking Problems

You are given a linked list containing $N$ elements, and a series of $k$ numbers - $a_1, a_2, \ldots, a_k$ - such that $\sum_i a_i = N$. Process this linked list by: reversing the first $a_1$ elements in the linked list, and multiplying all the elements by $a_1$. Once this segment is processed, reverse the next $a_2$ elements and multiply them with $a_2$, and proceed to next segments in a similar fashion. Output the processed linked list. *Note: You must use a linked list to solve this problem.*

## Input

The first line contains $N$, the number of elements in your linked list, and $k$, the number of segments ($1 \leq k \leq N \leq 5 \times 10^5$). The next line contains $N$ space separated values denoting the values in the linked list (all values between $-10^3$ and $+10^3$). The third line of input contains $k$ numbers, $a_1, a_2, a_3 \ldots, a_k$.

## Output

Output one line with $N$ space separated integers, denoting the modified list.

---

```
input
7 2
1 2 3 -1 -1 4 5
3 4
output
9 6 3 20 16 -4 -4

input
5 5
1 2 3 4 5
1 1 1 1 1
output
1 2 3 4 5
```

---

# H: Engineering Graphics With Trees

You are given a binary tree, represented in the form of an array $A$, where $A[0]$ represents the root. Additionally, for any node $A[i]$, $A[2i+1]$ and $A[2i+2]$ are the left and right children respectively. All nodes have positive integer values only, and a -1 indicates that no node exists there.

Assume that:

- all nodes at the same *height* from the root are in the same horizontal line.
- an observer looking from the left can only see one node at each height, i.e. the leftmost node. Similarly, an observer on the right side of the binary tree can only see the rightmost node at each height.

Find the *left view* and the *right view* of the binary tree - sum up all the values that can be seen from the right and call this $R$. Similarly, sum up all the values seen from the left and call this $L$. Output $abs(L-R)$.

## Input

The first line contains an integer $N$ ($1 \leq N \leq 10^5$), the size of array $A$. The next line contains $N$ space separated integers denoting the value in each of the nodes. Also, $-1 \leq A_i \leq 10^5, \forall i$.

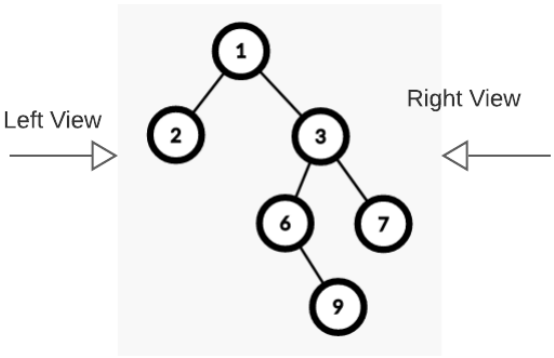
## Output

Output one line, consisting of only lowercase English alphabets, denoting the processed string.

---

```
input
9
1 2 3 -1 -1 6 7 -1 9

output
2
explanation
The nodes that can be seen from the
left are 1, 2, 6, 9.  So, L = 18.
Similarly, R = 20.  abs(L-R) = 2.
```




---

# I. Lazy Day at the Beach

The state of Florida wants to make its beaches safer from crocodiles by posting armed lifeguards at certain places on the beach. The state identified $N$ possible positions along the beach to potentially post guards - $x_1, x_2, x_3, \ldots x_N$ which are given in ascending order (guaranteed that $x_i < x_j$ if $i < j$). Placing a lifeguard at these positions will increase the "safety value" of the beach by $s_1, s_2, s_3, \ldots s_N$ respectively. Two guards cannot be posted within $X$ distance of each other, since they will end up chit-chatting, and ignore the safety of the beach - the distance between any two guards must be **strictly** greater than $L$. Calculate the maximal safety value possible, assuming you can hire as many guards as you want, provided no two guards you hire are positioned within $L$ distance of each other.

Hint: Let $A_k$ be the answer (maximal safety value possible) if we can use only the first $k$ positions on the beach $(x_1, x_2, \ldots, x_k)$. Clearly, $A_1 = s_1$. Now, let us say we know all values till $A_k$. To calculate $A_{k+1}$, we have two options: (1) place this $(k+1)^{th}$ lifeguard, and ignore all lifeguards that were placed from $(x_{k+1} - L)$ to $(x_{k+1} - 1)$ or (2) do not place lifeguard $A_{k+1}$. Select the maximal of these two options. For example, if $x_{k+1} - x_k > X$, adding the security guard at $x_{k+1}$ will not require removal of any previously placed guards.

## Input

The first line of input contains space separated integers $N$, $L$. $(1 \leq N \leq 10^4)$. The second line contains $N$ space separated integers, the possible positions $(1 \leq x_i, L \leq 10^5, \forall i)$. The third line contains $N$ space separated integers, the safety values for each of the positions $1 \leq s_i \leq 10^4$.

## Output

Print a single integer S denoting the maximal safety value that can be achieved.

---

```
input
5 5
6 7 12 13 14
5 6 5 3 1
output
10
```

---

```
input
4 2
6 9 12 14
5 6 3 7
output
18
```

---

# J. Merkle Tree

You are given $N$ unsigned positive integers as input in array $A$. Begin by multiplying the first two integers $A[0]$ and $A[1]$ - then *hash* the product (hash definition given below). Repeat this process for the next two elements, the next two elements and so on. If there is a single integer left over, hash it without multiplying it with anything else. Performing this process on the array $A$ of size $N$ gives us an array of hashes $B_1$ containing $N/2$ or $\frac{N-1}{2}+1$ hashes (if N is even or odd respectively). Now, process $B_1$ in a similar fashion to form $B_2$. Keep repeating the pairing and hashing process till we have only element (hash) left. Return this single value. The hash definition is given by,

```
def hash(x):
    x = x%65535
    x *= (x + 13);
    x = x%(65535);
    return x
```

Note: The value that you calculated is called the *merkle root or merkle hash*. This data structure, of a hierarchy of hashes, is one of the building blocks of blockchains. Additional reading.

## Input

The first line of input contains an integer $N$ ($0 \leq N \leq 10^3$). The next line contains $N$ space separated integers ($1 \leq A_i \leq 10^3$)

## Output

Print a single integer denoting the calculated single hash left at the end.

---

input

4

123 456 245 90

output

31905

explanation

Initially begin by multiplying $123 \times 456 = 56088$. Hash(56088) = 60933. Similarly, Hash($245 \times 90$) = 22845. Now, we have two hashes left - 60933 and 22845. The final answer is Hash($60933 * 22845$) = 31905

---

# CS F211
# Data Structures and Algorithms
# Assignment - 8

Allowed languages: **C**

March 26, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# Special Instructions

For binary-tree problems A, B, C, D, F you **must** use a pointer-based approach to solve these problems (syntax below). After inputting and constructing the tree,all subsequent operations must be performed **only on** the struct Node* objects. **Your code will be manually checked to ensure you did this, and you will be awarded zero score if any other approach is used.**

```
struct Node {
        int val;
        struct Node *left;
        struct Node *right;
};
```

Similarly, Question G also requires implementation of the solution using a specified data structure. Any other approaches will lead to a zero score.

# A: Binary Tree Blues - I

Given a binary tree, write a program to print the *zig-zag level order traversal*. To do this:

1. The *level* of a node is defined as the distance from the root node. Consequently, the root node has a level 0, and the children of the root would have level 1.
2. Start with level 0, & print all nodes in that level. Then print all nodes of level 1, and so on...
3. For all even levels, print the nodes of that level from left to right. For odd levels, print the nodes from right to left.

See the example and figure for clearer understanding.

## Input

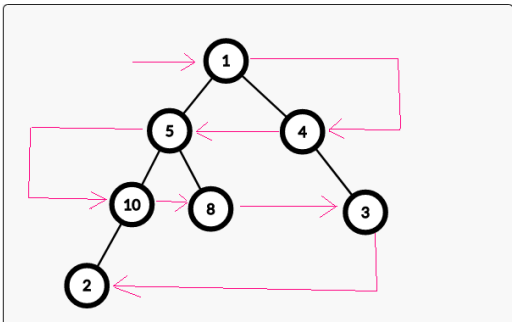
The first line contains a single integer $N$ ($1 \leq N \leq 10^3$), the number of nodes in the binary tree. The next line contains $N$ space-separated integers denoting the **values** of the $i^{th}$ node (zero-indexed). The $0^{th}$ element in this list is guaranteed to be the root node. The next $N-1$ lines contain two integers ($x$ and $y$ respectively) and a single character. This means that the $x^{th}$ element in the list of nodes is a parent of the $y^{th}$ element in the list of nodes, and the character (either 'L' or 'R') denotes whether it is the left child or right child. All values in the binary tree are in $[1, 10^9]$.

## Output

Print $N$ space separated integers representing the zig-zag level order traversal of the tree.

---

input
7
1 5 4 3 8 10 2
0 1 L
0 2 R
2 3 R
1 4 R
1 5 L
5 6 L
output
1 4 5 10 8 3 2
explanation
Follow the arrows.




---

# B: Binary Tree Blues - II

Given a binary tree with $N$ nodes, construct the tree and perform the following operations on it:

- Perform a **preorder** traversal and store it in an array $A$. Perform a **postorder** traversal and store it in array $B$. Perform an **inorder** traversal and store it in array $C$.
- Calculate $S = \sum_{i=0}^{N}\{abs(A_i - B_i) * C_i\}$ and since this number might be large, output it modulo $10^9 + 7$.

## Input

The first line contains a single integer $N$ ($1 \leq N \leq 10^3$), the number of nodes in the binary tree. The next line contains $N$ space-separated integers denoting the **values** of the $i^{th}$ node (zero-indexed). The $0^{th}$ element in this list is guaranteed to be the root node. The next $N - 1$ lines contain two integers ($x$ and $y$ respectively) and a single character. This means that the $x^{th}$ element in the list of nodes is a parent of the $y^{th}$ element in the list of nodes, and the character (either 'L' or 'R') denotes whether it is the left child or right child. All values in the binary tree are in $[1, 10^9]$.

## Output

Output the value $S$ as described in the question.

---

```
input
5
1 3 5 79 8
0 1 L
0 2 R
2 3 L
3 4 R
output
6484
explanation
Preorder traversal is [1, 3, 5, 79, 8].
Postorder traversal is [3, 8, 79, 5, 1].
Inorder traversal is [3, 1, 79, 8, 5]
```

---

# C: Binary Tree Blues - III

Perform an **inorder traversal** of a binary tree **without using recursion**. Your program **must** have exactly **one** function definition only - main() and no other functions. *Hint: You might need to use a stack. Additional thinking: can you do postorder, or pre-order traversals without using recursion?*

## Input

The first line contains a single integer $N$ ($1 \leq N \leq 10^3$), the number of nodes in the binary tree. The next line contains $N$ space-separated integers denoting the **values** of the $i^{th}$ node (zero-indexed). The $0^{th}$ element in this list is guaranteed to be the root node. The next $N-1$ lines contain two integers ($x$ and $y$ respectively) and a single character. This means that the $x^{th}$ element in the list of nodes is a parent of the $y^{th}$ element in the list of nodes, and the character (either 'L' or 'R') denotes whether it is the left child or right child. All values in the binary tree are in $[1, 10^9]$.

## Output

Output $N$ space separated integers represnting the inorder traversal of the given tree.

---

```
input
5 1
1 3 5 79 8
0 1 L
0 2 R
2 3 L
3 4 R
output
3 1 79 8 5
```

---

# D: Binary Tree Blues - IV

Given a binary tree with $N$ nodes perform $T$ operations on it. The format of each query is:

- "**DELETE A**": Find a node which has value A stored in it, and delete that node and all it's descendants.
- "**SWAP A B**": Find nodes that have values A and B in them respectively. Swap the subtrees rooted at A and B. Swap A (and all its children) with B (and all its children).

After completing all $T$ operations, print the **diameter** of the final binary tree. To calculate the diameter, identify the longest path (without cycles) between *any two leaves* in the binary tree. The path length between two nodes is the number of edges between them.

## Input

The first line contains space-separated integers $N$ ($1 \leq N \leq 10^3$) and $T$ ($0 \leq T \leq 10$). The next line contains $N$ space-separated **unique** integers denoting the **values** of the $i^{th}$ node (0-indexed). The $0^{th}$ element in this list is the root node. The next $N-1$ lines contain two integers ($x$ and $y$ respectively) and a single character, meaning that the $x^{th}$ element in the list of nodes is a parent of the $y^{th}$ element in the list of nodes, and the character ('L' or 'R') denotes whether it is the left or right child. The next $T$ lines contain one query each.

## Output

Print the diameter of the final tree.

---

input

```
8 2
2 6 5 1 3 4 99 128
0 1 L
0 2 R
1 3 L
2 4 L
4 5 R
5 6 L
6 7 R
DELETE 99
SWAP 1 3
```

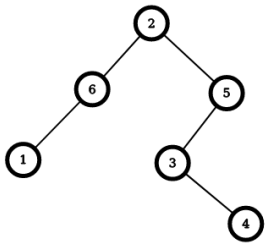
output

```
5
```

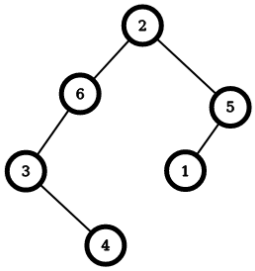
explanation

```
The diameter is from
node 4 to node 1.
```

After delete operation:




After swap operation:



# E: Binary Tree Blues - V

Given the inorder and preorder traversals of a binary tree, print its' postorder traversal.

## Input

The first line contains an integer $N$ representing the number of elements in the binary tree ($1 \leq N \leq 10^3$). The second line contains $N$ space separated integers representing the inorder traversal of the tree. The third line, which is similarly formatted, contains the preorder traversal of the tree.

## Output

Print $N$ space-separated integers representing the postorder traversal of the tree.

---

```
input
3
1 3 2
1 2 3

output
3 2 1
```

---

# F. Binary Tree Blues - VI: Binary Possibilities

You are given a Binary Search Tree and $q$ queries. Each query consists of two nodes. Find the Lowest Common Ancestor of these nodes. Let T be a rooted tree. The lowest common ancestor between two nodes $n_1$ and $n_2$ is defined as the node in T that has both $n_1$ and $n_2$ as descendants (where we allow a node to be a descendant of itself) and is farthest possible from the root while still satisfying the previous condition.

## Input

The first line contains two space-separated integers $N$ ($1 \leq N \leq 10^3$), the number of nodes in the BST and $q$ ($0 \leq q \leq 100$), the number of queries. The next line contains $N$ space-separated integers denoting the **values** of the $i^{th}$ node (zero-indexed). The $0^{th}$ element in this list is guaranteed to be the root node. The next $N-1$ lines contain two integers ($x$ and $y$ respectively) and a single character. This means that the $x^{th}$ element in the list of nodes is a parent of the $y^{th}$ element in the list of nodes, and the character (either 'L' or 'R') denotes whether it is the left child or right child. All values in the binary tree are in $[1, 10^9]$. The next $q$ lines each contain two integers $a$ and $b$ representing the nodes for which you have to find the least common ancestor.

## Output

Print $q$ space separated integers, one for each query print a single integer representing the node that is the least common ancestor of the two nodes given in the query.

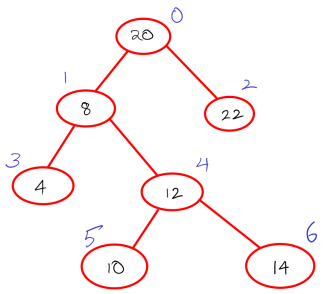
---

input

```
7 5
20 8 22 4 12 10 14
0 1 L
0 2 R
1 3 L
1 4 R
4 5 L
4 6 R
1 2
5 2
1 6
5 6
1 1
```

output

```
0 0 1 4 1
```




explanation

For the first case we can see that the LCA is 0. In the second case the LCA is also 0. In the third case, the LCA is 1 because the LCA can be one of the nodes themselves. The LCA in the fourth case is 4. In the last case, they are both the same node so the answer is 1 itself.

---

# G. Stack Overkill

Implement a simple stack that performs two operations: push and pop. push should push an element on to the top of the stack and pop removes the element on the top of the stack. There is one catch though. While implementing this stack, you must do so using **two queues**. You will have $N$ queries. All of them will be of three types:

1. "PUSH X ": push the number X on top of the stack
2. "POP": pop the topmost element from the stack.
3. "PRINT": print the value of the element on top of the stack

*Note that you have to use exactly two queues to implement this stack. Your code will be **manually checked** (by humans) to make sure you have done this. Implementing it in any other fashion will lead to a **zero score**.*

## Input

The first line of input contains an integer $N$ ($1 \leq N \leq 2 \times 10^5$). The next $N$ lines contain one of the above operations. If the operation is "PRINT" then it will be followed by a single integer "X" ($1 \leq X \leq 10^9$). It is guaranteed that the stack is not empty when the operation is "PRINT" or "POP".

## Output

For each "PRINT" function, print a single integer in a single line representing the element at the top of the stack at that point in time.

---

```
input
7
PUSH 3
PUSH 5
PRINT
POP
PRINT
PUSH 100
PRINT

output
5
3
100

explanation
It's pretty easy to follow the stack operations to see that the sample case is correct
```

---

# H: I Know You'll Google This

Given an array having both positive and negative integers. The task is to compute the length of the largest contiguous subarray with sum 0.

## Input

The first line contains one integer $N$, the sizes of array ($1 \leq N, \leq 2 \times 10^5$). The next line contains contains $N$ integers each integer representing $a_i$ ($-10^9 \leq a_i \leq 10^9$).

## Output

Output one integer, the length of the largest such subarray. If no such subarray exists, print 0.

---

```
input
8
15 -2 2 -8 1 7 10 23

output
5
explanation
The largest subarray is:  -2 2 -8 1 7
```

---

# I: Infuriatingly not Binary Search

You are given an array that was initially sorted, but now it has been cyclically rotated $k$ times to the right. ($k$ is unknown to you). Given $q$ queries where each query contains a number $x_i$, find whether $x_i$ exists in the array or not. **Your algorithm has to be O(1) space complexity (not including storing the original array). Your code will be checked for this**.
Also note that the array can contain duplicate numbers.

## Input

The first line contains the integer $N$ ($1 \leq N \leq 10^5$), the number of elements in the array. The next line contains all the elements in the array ($1 \leq a_i \leq 10^9$), and it is guaranteed that all $a_i$ are distinct. The next line contains an integer $q$ ($1 \leq q \leq 10^5$) representing the number of queries. Each of the next $q$ lines contains a single integer $x_i$, the number for which you have to check whether or not it is present in the array.

## Output

For each query output a single string, "YES" if the number exists in the array or "NO" if it does not. Note that you must print the strings without quotes and all in capital letters.

---

input
8
5 6 7 8 10 1 2 3
5
5
8
12
2
1000000000
output
YES
YES
NO
YES
NO
explanation
In the five queries, you can easily see whether or not the corresponding elements are present in the array or not

---

# J: Basic DP

Write an efficient program to find the sum of elements in a contiguous subarray within an array of numbers which has the largest sum.

## Input

The first line contains the integer $N$ ($1 \leq N \leq 10^5$), the number of elements in the array. The next line contains $N$ space separated integers, $a_i$ ($-10^9 \leq a_i \leq 10^9$) - representing the series of numbers.

## Output

Print a single integer representing the sum of the subarray with that largest sum.

---

```
input
8
-2 -3 4 -1 -2 1 5 -3

output
7
explanation
Here the largest contiguous subsegment is from indices 3 to 7 inclusive
```

---

# CS F211
# Data Structures and Algorithms
# Assignment - 9

Allowed languages: **C**

March 26, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# A: The Shortest Statement You'll Ever Find

You are given an array of $N$ integers and $q$ queries with the $i^{th}$ query containing the number $x_i$. For the $i^{th}$ query print the frequency of $x_i$ in the array. Implement this using a red black tree. *Note: Try to make your implementation as general as possible since subsequent problems also might require the usage of a Red-Black tree.*

## Input

The first line contains a single integer $N$ as described above ($1 \leq N \leq 10^5$). The second line of input contains $N$ integers where the $i^{th}$ number represents $a_i$ ($1 \leq a_i \leq 10^9$). The third line contains a single integer $q$. The next $q$ lines contain a single integer each, representing the $i^{th}$ query.

## Output

Print $q$ lines, the $i^{th}$ line containing the answer for the $i^{th}$ query.

---

input

```
7
2 5 1000000000 7 1000000000 5 5
3
5
1000000000
1
```

output

```
3
2
0
```

explanation

In the first query, the frequency of 5 is 3. In the second query, the frequency of 1000000000 is 2. In the third query, there is no 1 anywhere in the array, hence the frequency is 0

---

# B: The Best Part of Being a Scout

Levi is a complete neat freak and his squad has to clean up his entire room before he gets back from a scouting expedition. They're finished with almost everything and they need your help to clean up one last thing. There is an array of $N$ elements with the $i^{th}$ element being $a_i$. Levi thinks that the array is clean if $frequency(a_i) = a_i$ for all $i$. That is, the frequency of $a_i$ should be $a_i$. Note that an empty array is considered clean. Levi's array is a little messed up and isn't clean right now. Find the smallest number of elements to delete to make the array clean.

## Input

The first line contains an integer $N$ representing the number of elements in the array ($1 \leq N \leq 10^5$). The next line contains $N$ numbers representing the $i^{th}$ element of the array $a_i$ ($1 \leq a_i \leq 10^9$).

## Output

Print a single integer, the minimum number of elements to delete to make the array clean

---

input

```
5
2 4 1 4 2
```

output

```
2
```

explanation

There is no way to get $4$ elements with the value $4$, so delete all the elements with value $4$. Hence we would remove $2$ elements and the remaining array is $[2, 1, 2]$. This is a clean array since $2$ occurs $2$ times and $1$ occurs $1$ time.

---

# C: Sleepwalking

Dorothy Unsworth sucked you up into her dream world again and is willing to let you out only if you beat her at a game. You are given $N$ positive integers. In one operation, you can choose some even number $c$ and divide *all* elements in the array equal to $c$ by 2. The goal of the game is to perform operations until the array consists entirely of odd numbers. If you stay in the dream world too long you'll fall asleep and die so you have to find the minimum possible number of operations to make the array consist of only odd numbers.

## Input

The first input line has a single integer $N$ ($1 \leq N \leq 10^5$) The next line contains $N$ integers representing the array ($1 \leq a_i \leq 10^9$).

## Output

Output a single integer representing the smallest number of moves needed to make every element in the array odd.

---

input
6
40 6 40 3 20 1

output
4

explanation
Here the optimal sequence of moves is as follows: Choose $c$ as 40 and make all the 40s in the array equal to 20. Now we have: $20, 6, 20, 3, 20, 1$. Now if we choose $c$ as 20, we have $10, 6, 10, 3, 10, 1$. Next, take $c$ as 10. The array becomes $5, 6, 5, 3, 5, 1$. Lastly, take $c$ as 6 to get the final array as $5, 3, 5, 3, 5, 1$ where all the numbers are odd. This took a total of 4 steps and it is easy to see that this is the minimum number of steps.

---

# D: Captain and Vice Captain

Luffy made the biggest mistake of his life and sent Zoro out to get groceries. Now it's too late and Zoro went around in circles trying to find it although somehow he finally reached it. Luffy has gotten really good at using observation haki and using that, he's able to determine the exact path that Zoro takes. The path that Zoro takes can be imagined on the coordinate plane with his possible moves being up,right,left, or down by one unit. Assume his initial coordinate is $(0, 0)$. The path is represented by a string containing the letters 'L' (Left), 'R' (Right), 'U' (Up), and 'D' (Down). Luffy wants to save Zoro some time and to do that he can remove any non empty contiguous substring from the path *without changing the Zoro's final destination*. Since Luffy is lazy and doesn't actually care that much, he wants to find the smallest possible such substring by length. Help him find it by printing the indices of the starting and ending elements of the substring. If there is no such substring print $-1$. Print the indices with 1-indexing.

## Input

The first line contains a single integer $N$ representing the length of the string ($1 \leq N \leq 10^5$). The next line contains a string consisting of only the above mentioned 4 letters.

## Output

print two integers $l$ and $r$ representing the starting and ending indices of the substring in 1-indexing. If there is no such non empty substring then print $-1$.

---

```
input
10
LURDLLLRRD

output
7 8

explanation
You could also remove the substring from 1 to 4 and maintain the same destination,
but the size of that substring is not minimum.  Any other substring would change the
final destination
```

---

# E: Gojou is on Another Vacation

You have encountered a somewhat powerful spirit and it already has you captured in a maze that is in the form of a tree (with nodes and edges). To escape the maze you must first place a starting cursed energy bomb and an ending cursed energy bomb on two distinct nodes $u$ and $v$ respectively and then traverse the shortest path from $u$ to $v$. There is one condition though. There exist two special nodes $x$ and $y$ such that if you reach $y$ after reaching $x$ somewhere in the path from $u$ to $v$, then your plan fails. The special nodes $x$ and $y$ are given to you beforehand. Find all possible pairs $u, v$ where you can place the cursed energy bombs.

## Input

The first line contains 3 integers, $N, x, y$ where $N$ is the number of nodes and $x$ and $y$ are the special nodes ($1 \leq N \leq 10^5, 1 \leq x, y \leq N, x \neq y$). The next $N-1$ lines contain two integers each, $u_i$ and $v_i$ representing an edge between $u_i$ and $v_i$.

## Output

Print a single integer representing the number of pairs of nodes that you can place the bombs on.

---

input

```
3 1 3
1 2
1 3
```

output

```
4
```

explanation

In this example, you can go from 2 to 1, 1 to 2, 3 to 2, and 3 to 1. Any other path will pass through $x$ and $y$ in that order

---

# F: BITSian Standard Time (BST)

You are given a series of $N$ numbers. For each of these numbers, you will have to perform operations on a Binary Search Tree (BST).

1. begin with an empty BST.
2. read the next number in the series from input. Let us call this input $A_i$.
3. if $A_i$ is already present in the BST, ignore that input.
4. if $A_i$ is not present in the BST, insert it into the tree as a leaf at the appropriate position. Do not perform any rebalancing.

Print the *postorder traversal* of the formed BST. Note: Take a look at this article for more information on a BST. You can also use this visualiser to help you plot BSTs if you're having trouble.

## Input

The first line contains the integer $N$ ($0 \leq N \leq 10^4$), the number of elements in our series. The next line contains $N$ space separated integers - $A_i$ ($1 \leq A_i \leq 10^9$) - representing the series of numbers.

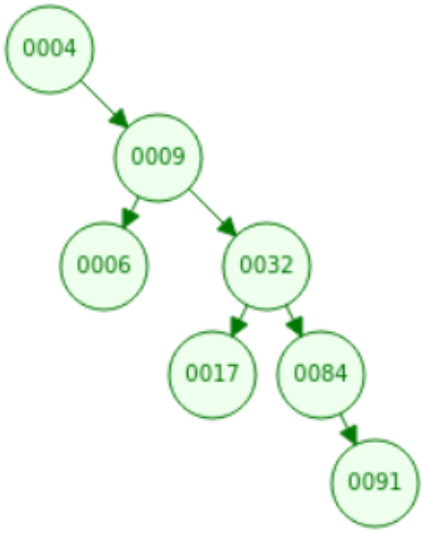
## Output

Print space separated integers denoting the *postorder traversal* of the BST formed after processing all elements. *Note*: The number of elements in the output is not necessarily equal to the number of elements in the input series. This is because repeated elements in the input are ignored.

---

```
input
8
4 9 4 32 17 84 6 91

output
6 17 91 84 32 9 4
explanation
The final BST formed is shown on
the right.  We perform a postorder
traversal on that.
```




---

# G. Bugs

There are $N$ bugs on a horizontal string of length $S$ units (from $x = 0$ to $x = S$). Each of the bugs is initially located at position $x_i$ ($0 \leq x_i \leq S$) and each of the bugs is moving either to the left ($-X$ direction) or the right ($+X$ direction) with the same speed of 1 unit/second. When a bug reaches positions $x = 0$ or $x = S$, they immediately fall off the string. When two bugs travelling in opposite directions end up at the same position, they collide *elastically*, i.e. after collision, both bugs start moving in directions opposite to what they were originally moving in, with the same 1 unit/second speed. It is guaranteed that each bug is initially at a unique position, i.e. no two bugs share the same initial position. How long will it take for all the bugs fall off the string?

## Input

The first line of input contains two integers $N$ and $S(0 \leq N \leq 5 \times 10^5, 1 \leq S \leq 10^9)$. The next $N$ lines contain two space space separated values - one integer and one character ("L" or "R"), representing the original position and original direction of the $i^{th}$ bug.

## Output

Print a single integer denoting the time at which all bugs have fallen off the string.

---

input

3 7
2 L
4 R
6 L

output

6

explanation

We have three bugs - $B_0$, $B_1$, and $B_2$. $B_0$ does not collide with any other bugs and falls off the left side of the table at $t = 2$. $B_1$ and $B_2$ collide at $t = 1$ at position $x = 5$. $B_1$ that starts going to the left and $B_2$ starts going to the right. $B_1$ falls off the left edge at $t = 6$, while $B_2$ falls off the right edge at $t = 3$.

---

# H: Tower Defense (Part I)

You are sitting on a tower at the point $(0, 0)$ in a 2D plane. There are $N$ enemies approaching the tower, with the locations of the enemies being $(x_1, y_1), (x_2, y_2) \ldots (x_N, y_N)$. You, a defender in the tower, have a laser beam that can shoot a laser in a straight line in any direction. One shot from the laser will obliterate all life on a line beginning at the tower $(0, 0)$ and extending in any direction. Calculate the minimum number of lazer shots required to eliminate all advancing enemies.

## Input

The first line contains the integer $N$ ($1 \leq N \leq 10^3$), the number of enemies. The next $N$ lines contain one coordinate each $(x_i, y_i)$ (one indexed) representing the position of the $i^{th}$ enemy. ($-10^8 \leq x_i, y_i \leq 10^8$).

## Output

Output one integer, the number of laser shots required.

---

input
```
2
1 0
-1 0
```

output
```
2
```

explanation

Two laser shots are needed since both points are on opposite sides of the tower.

input
```
5
3 4
0 9
-12 -36
9 12
-6 -18
```

output
```
3
```

explanation

(-12, -36) and (-6, -18) can be eliminated in one shot. Similarly for (3,4) and (9,12)

---

# I: Tower Defense (Part II)

You are sitting on a tower at the point $(0, 0)$ in a 2D plane. There are $N$ enemies approaching the tower, each of which is arriving from a certain angle, $A_i$, in a straight line towards the tower. You, the defender of the tower, have upgraded the laser allowing you to attack more enemies at once. Your new laser can attack everyone bounded between an arc of $k$ degrees and the centre of the circle. Assume that one blast from the laser beam can affect *all* enemies located within the range of angles $[i, (i + k)\%360]$ (both extremes inclusive), for any integral $i < 360$. What is the *minimum* number of laser blasts needed to kill all enemies?

*Additional thinking (not a part of this problem): Let us say the enemies are all wearing shields of varying strengths to protect themselves. Each $i^{th}$ enemy now has a different health $h_i$. An enemy with health $h$ requires $h$ hits from the laser before falling dead. How would you solve this extension to the problem?*

*Note: Notice that with $k = 0$ this problem reduces to the previous one.*

## Input

The first line contains the integer $N$ ($1 \leq N \leq 10^3$), the number of enemies and $k$ ($1 \leq k \leq 90$). The second line of input contains $N$ space separated integers denoting the angle of approaches, $A_i$ ($0 \leq A_i \leq 360$), of the advancing enemies.

## Output

Output one integer, the number of laser shots required.

---

input
3 90
45 60 75
output
1
explanation
One shot of the laser is enough to eliminate them all.

input
6 10
15 16 17 18 19 78
output
2
explanation
One shot of the laser is enough to eliminate them all.

input
5 10
114 114 114 355 5
output
2
explanation
Enemies approaching from 355, 5 can be eliminated with one shot.

input
3 45
13 58 178

output
2
explanation
Enemies approaching from 13 and 45 can be eliminated in one shot.

---

# J. Inverting a Binary Tree

You are given a binary tree in the form of an array $A$, where $A[0]$ is the root. For any node $A[i]$, the left and right children are located in $A[2i+1]$ and $A[2i+2]$ respectively. All values in the binary tree are guaranteed to be positive integers. If a node does not exist at a certain index, the value $-1$ will be located in the array. Invert the binary tree by:

1. Start at the root node. Reverse the digits in the value at this node. For eg, if the value in root node is 123, change it to 321. Then, swap the positions of the entire right subtree with the entire left subtree.

2. Once the root node is processed, go to each of their children, and repeat this process of reversing the value in the node and swapping the positions of its children subtrees.

3. repeat this step all the way down to leaves. For each leaf, after reversing the value in each of the leaves, swapping its children (NULL and NULL) causes no change in the tree.

Output the inorder traversal of the final tree.*Note: after reversing the digits of a number, you need to preserve leading zeroes. i.e. reverse of 980 is 089.*

## Input

The first line contains $N$ ($1 \leq N \leq 10^4$), the number of elements in array $A$. The second line contains N space separated integers representing the binary tree ($A_i \leq 10^9$) in array format.

## Output

Print space separated integers denoting the inorder traversal of the final tree.

---

```
input
7
123 45 92 70 60 -1 5

output
5 29 321 06 54 07

explanation
We notice that our input has 7 elements (one empty element denoted by -1).  However,
the number of integers in the output is equal to the number of elements in the tree,
6.
```

---

# CS F211
# Data Structures and Algorithms
# Assignment - 9

Allowed languages: **C**

March 31, 2021

## General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. You can install Windows Subsystem Linux (WSL) or MinGW 7.3.0, if you are Windows user to compile and run your programs. Alternatively, you can run and test your codes on Online GDB. If you are using WSL or Linux to run your programs, make sure that the gcc version is `gcc 5.4.1 c99`.

# A: Erwin's Gambling Addiction

In a much more mellow world, Erwin's gambling tendencies have got him in quite a pinch and he needs your help to win his money back. He will give you a number $N$ and a six sided unbiased die. You have at most $N$ rolls. You have to tell him how many distinct sequences of rolls exist such that the sum of the numbers on each roll adds up to $N$.
For example if $N = 3$:

1. $1 + 1 + 1$
2. $2 + 1$
3. $1 + 2$
4. $3$

Hence there are 4 ways. Since the answer can be very large, print the answer modulo $10^9 + 7$

## Input

The first and only line contains a single integer $N$ ($1 \leq N \leq 10^5$).

## Output

Print a single integer, the number of ways to get a sum $N$ modulo $10^9 + 7$

---

```
input
3

output
4

explanation
The explanation is given in the question
```

---

# B: Another Short Question

You are given a number $N$ in the base 10 format. In one operation, you can choose any digit that appears in the number $N$, and subtract it from $N$. Find the minimum number of operations required to make $N$ equal to 0.

## Input

The first and only line contains a single integer $N$ ($1 \leq N \leq 10^5$).

## Output

Print a single integer, the minimum number of operations required to make $N$ equal to 0.

---

input
27

output
5

explanation
The operations would go as follows: $27 \to 20 \to 18 \to 10 \to 9 \to 0$

---

# C: Go D.

Usopp has met another overpowered enemy and is running away again. The layout of the castle he's in is an $N \times N$ grid consisting of empty cells and blocked cells. He is currently at the top left corner of the grid and he has to make his way down to the bottom right corner by only walking on empty cells and by performing two moves, either *right* or *down*. However he must not move to any blocked cell on the grid. Empty cells are represented by '.' and blocked cells are represented by '*'. Find the number of distinct ways to reach the bottom right corner by using only those two moves. Since the number can be large, output the answer modulo $10^9 + 7$

## Input

The first line contains a single integer $N$ ($1 \leq N \leq 10^3$), the number of rows and columns in the grid. The next $N$ lines contain $N$ characters each, describing the board.

## Output

Output a single integer representing the number of distinct paths to get to the bottom right corner.

---

input

```
4
....
.*..
...*
*...
```

output

```
3
```

explanation

The three different sequences of moves are given as follows: $(D \to D \to R \to D \to R \to R)$, $(D \to D \to R \to R \to D \to R)$, $(R \to R \to D \to D \to D \to R)$.

---

# D: Knapsack

You have a Knapsack with you that can handle a maximum weight of $W$. You also have $N$ items, the $i^{th}$ integer having a weight $w_i$ and cost $c_i$. You now have to put items in the knapsack such that the total weight of all the items does not exceed $W$ and the total cost is maximised. Note that you can only use each item at most once.

## Input

The first line contains two integers $N$ and $W$ ($1 \leq N \leq 100$, $1 \leq W \leq 10^4$). The next line contains $N$ integers representing the weight of each element, $w_i$ ($1 \leq w_i \leq W$). The third line also contains $N$ integers, representing the cost of each element, $c_i$ ($1 \leq c_i \leq 10^3$).

## Output

Print a single integer representing the maximum cost possible satisfying the above conditions.

---

input

```
3 8
3 4 5
30 40 60
```

output

```
90
```

explanation

If you take the first and third element, you will get a total cost of $60 + 30 = 90$ which is the maximum possible without exceeding the weight limit of $8$.

---

# E: A Useless Timeskip

Since Naruto only knows rasengan variations, he decided to learn some new jutsu to increase his power. He decided to learn three new jutsu, A,B, and C. Each day he chooses one of the jutsu and trains only that jutsu. Since he gets bored easily, he cannot choose the same jutsu on two consecutive days. One thing about this training is that it does not increase his power level uniformly. More specifically, on the $i^{th}$ day, training the jutsu A increases his power level by $a_i$, jutsu B increases his power level by $b_i$, jutsu C increases his power level by $c_i$. He only has $N$ days to train so he wants you to help him find out what his maximum power level can be after $N$ days of training according to the above restrictions. Assume his inital power level before beginning training is 0.

## Input

The first line contains one integer $N$ ($1 \leq N \leq 10^5$). The second line contains $N$ integers representing the increase in power level if he trains jutsu A on the $i^{th}$ day, $a_i$ ($1 \leq a_i \leq 10^4$). The third line contains $N$ integers representing the increase in power level if he trains jutsu B on the $i^{th}$ day, $b_i$ ($1 \leq b_i \leq 10^4$). The fourth and last line contains $N$ integers representing the increase in power level if he trains jutsu C on the $i^{th}$ day, $c_i$ ($1 \leq c_i \leq 10^4$).

## Output

Print a single integer, the maximum possible power level after training for $N$ days.

---

input

```
3
10 20 30
40 50 60
70 80 90
```

output

```
210
```

explanation

If he trains C on the first day, B on the second day and C again on the third day, his overall power level will be 210 which is the highest possible in this case.

---

# F: Tree Overkill

You are given an k-ary (k is unknown to you) tree and two vertices. You simply have to output the distance between these two vertices. There is one catch. You have to store the trees using pointers and Left-Child-Right-Sibling notation. The definition for a node should look like this or something similar:

```
struct Node {
        int val;
        struct Node *parent
        struct Node *left_child;
        struct Node *next_sibling;
};
```

It is guaranteed that the root node is node 1.

## Input

The first line contains three integers $N, u, v$ ($1 \leq u, v \leq N \leq 10^3$), the number of vertices in the tree. The next $N-1$ lines contain two space separated integers, $x$ and $y$ where $y$ is a child of $x$.

## Output

output a single integer, the distance between the vertices $u$ and $v$.

---

input
4 3 4
1 2
2 3
1 4

output
3

explanation
If you draw the graph, you can see that the distance between vertices 3 and 4 is 3

---

# G. Longest Common Subsequence

You are given two strings $s$ and $t$. You have to find the length of the longest subsequence that is present in both $s$ and $t$.
Note that a subsequence of a string x is the string obtained by removing zero or more characters from x and concatenating the remaining characters without changing the order. For example, "*rishiplayingdressup*" and "*geethplayingdress*" are both subsequences of the string "*rishiandgeethplayingdressup*", but the string "*geethandrishidress*" is not because the order of the letters is not maintained.

## Input

The first line contains two numbers $N$ and $M$, the length of strings $s$ and $t$ respectively ($1 \leq N, M \leq 10^3$). The second and third lines contain a single string each, consisting of only lower case English letters.

## Output

Print a single integer representing the length of the longest common subsequence in the two strings.

---

input
4 5
axyb
abyxb

output
3

explanation
Two subsequences that have a length 3 in this case are $axb$ and $ayb$ which are the longest possible.

---

# H: Nikee

You just bought a new pair of special shoes that allow you to jump abnormal heights. You want to test this out so you go to a street in the city that has $N$ buildings all adjacent to each other (meaning there's no space in between buildings). You want to get from the leftmost building to the rightmost building by jumping across the tops of the buildings. You start from the leftmost building and at any point in time, if you are currently on the $i^{th}$ building, you can jump to either the $i+1^{th}$ building or the $i+2^{th}$ building (as long as $i+1$ and $i+2$ are less than $N$). The company that made the shoes is very greedy so they charge you money through paytm for every jump. More specifically, if the difference in heights between your initial position and final position is $h$, then you have to pay an amount of $h$. Find the minimum cost required to reach the last building. In summary, from position $i$ you can go to position $i+1$ or $i+2$ and the cost required to get there is the absolute difference in heights between the current position and the position you go to.

## Input

The first line contains the integer $N$ ($1 \leq N \leq 10^5$), the number of buildings. The next line contains $N$ integers, $h_i$ being the height of the $i^{th}$ building ($1 \leq h_i \leq 10^4$)

## Output

Output one integer, the minimum cost possible.

---

```
input
6
30 10 60 10 60 50

output
40

explanation
The optimal path to take here would be the indices 1 → 3 → 5 → 6 where the costs
would be |30 − 60| + |60 − 60| + |60 − 50| = 40
```

---

# I: Coin Minimization

You live in a country where there are $N$ types of coins, each worth a different value. The values of the coins are known to you. You need to pay someone only in coins so you want to make a total of $x$ using only these coins. Note that you can use multiple coins of the same type. If it is not possible to get the desired sum, print $-1$.

## Input

The first line contains two integers $N$ and $x$ ($1 \leq N \leq 100$, $1 \leq x \leq 10^4$), the number of types of coins and the sum you need to achieve respectively. The second line contains $N$ integers, $c_i$ representing the value of the $i^{th}$ coin ($1 \leq c_i \leq 100$).

## Output

Output one integer the minimum number of coins required or $-1$ if it is not possible to get the sum $x$.

---

input

3 11

1 5 7

output

3

explanation

here an optimal solution would be to take $5+5+1$. You could take $7+1+1+1+1$ $5+1+1+1+1+1+1$ or something similar but that would not be the minimum number of coins possible.

---

# J. Finally Some Non-Pointer Trees

You are given a tree with $N$ nodes. You have to find the length of the longest path in the tree. If you are using dfs or bfs to solve this, you may only traverse the tree once. (If you google this question you'll mostly get an answer with 2 dfs's/bfs's). Note that you don't have to use pointers in this problem.

## Input

The first line contains $N$ ($1 \leq N \leq 10^5$), the number of nodes in the tree. The next $N-1$ lines contain two integers each, $u$ and $v$ where there is an undirected edge between the node $u$ and the node $v$.

## Output

Print a single integer, the length of the longest path in the tree.

---

```
input
5
1 2
2 3
2 4
4 5

output
3

explanation
In this graph there are 2 paths with length 3, 1 → 2 → 4 → 5 and 3 → 2 → 4 → 5
```

---