# 21*st* South African Regional
# International Collegiate Programming Contest

26 October 2019

## Problem A — Silver Balloon
## Even Segments

### Problem Description

Bokamoso has a fetish for even numbers, loving them more than odd numbers. Whenever Bokamoso is presented with an array of numbers, Bokamoso tries to divide the array into an even number of contiguous non-empty subsegments where each subsegment begins and ends in an even number. For example, if Bokamoso is presented with the array $[2, 3, 4, 5, 6, 4, 1, 2, 8, 9, 10]$, they can divide the array in either of the following ways:

- $[2, 3, 4, 5, 6], [4, 1, 2, 8, 9, 10]$; or
- $[2, 3, 4, 5, 6, 4, 1, 2], [8, 9, 10]$.

Sometimes dividing the arrays makes Bokamoso an irritable person. You decide, as the recipient of this irritation, to write a program to determine whether an array can be divided or not.

### Input

The input consists of an arbitrary number of records, but no more than 35.

Each test record consists of two lines. The first line contains $N$, the length of the array, where $1 \leq N \leq 10^5$.

The second line contains $N$ space-separated integers $A_1, A_2, \cdots, A_N$, with $(-10^9 \leq A_i \leq 10^9)$, and represents the numbers of the array.

Input is terminated by $-1$ on a line of it's own.

### Output

For each test record, output either "yes" if the array can be divided, or "no" if it cannot be divided.

### Sample input

```
4
1 3 2 5
6
2 6 4 8 0 12
6
0 0 0 0 0 0
6
1 1 1 1 1 1
-1
```

### Sample output

```
no
yes
yes
no
```

### Resource Constraints

- **Time limit**: 1 second
- **Memory limit**: 256 MB

### Attribution

This problem was adapted from ECPC 2019.

# 21*st* South African Regional International Collegiate Programming Contest

26 October 2019

## Problem B — Pink Balloon Social Networking

### Problem Description

Your friend, Zark Muckerberg, has implemented a simple social network that consists of only $N$ users and $N-1$ friendship relations between them.

The relations between users in the network can be represented as a tree where nodes represent the users and edges represent the relations.

Each user has a name (consisting of only lowercase characters $a..z$) and an ID from 1 to $N$. ID's are unique but names may be duplicated. The service also allows users to update their names by adding one or more characters to their old name.

Zark wants to add a search engine that helps users to find other users.

If user $U$ writes the query string $S$ in the search box, the engine should return the ID of the closest user (according to the tree topology) that has $S$ as a prefix of his name, where the distance between two nodes is measured by the number of edges between them.

### Input

The input contains an arbitrary number of test records, but no more than 20.

Each test record consists of multiple lines where the first line contains two numbers, $N$ and $Q$ respectively, where $N$ represents the number of users on the system and $Q$ represents the number of user transactions to be processed ($1 \leq N, Q \leq 50000$).

The next $N$ lines represent the names of the users in the network where the $i^{th}$ line represents the name of the user with ID $i$. The sum of lengths of all names will not exceed 50000.

The next $N-1$ lines represents the relations between users, each line contains 2 integers $U$ and $V$ which means that there is an edge between the user with ID $U$ and user with ID $V$ ($1 \leq U, V \leq N$, $U \neq V$).

The next $Q$ lines contain two integers, $T$ and $I$ ($T \in \{1, 2\}$ and $1 \leq I \leq N$), followed by a single word $S$ (all characters of $S \in \{a, b, ..., z\}$). $T$ identifies the type of transaction, where 1 means user with ID $I$ requested to update his name by appending $S$ onto his name, and 2 means user with ID $I$ requested to search for $S$.

Input is terminated with $-1$ on a line by itself.

### Output

For each search in each test record output the ID of the found user, or $-1$ if no username starts with the provided search string.

If multiple users at the same distance match a search string, output the user with the lowest ID.

### Sample input

```
4 5
axc
ab
abcx
b
1 2
1 3
2 4
2 4 abcd
2 4 abc
2 4 b
1 2 cd
2 4 abcd
-1
```

### Sample output

```
-1
3
4
2
```

### Resource Constraints

- **Time limit**: 15 seconds
- **Memory limit**: 1024 MB

### Attribution

This problem was adapted from ECPC 2019.

# 21*st* South African Regional
# International Collegiate Programming Contest

26 October 2019

## Problem C — Purple Balloon Paladin

### Problem Description

Fluffy's biggest dream is to become a Paladin. In order to do that she needs to find the best possible way to turn arbitrary words, consisting purely of the lowercase characters $a..z$, into Palindromes.

Fluffy is further restricted by the operations she can perform on a word in order to morph it into a Palindrome. In fact, as a single step she can change only a single character by adjusting it to an adjacent character. For example, "g" can be morphed to "f" or "h". Given the word "abcz" the following results are possible in a single step:

1. bbcz
2. zbcz
3. accz
4. aacz
5. bbbz
6. bbdz
7. bbca
8. bbcy

Fluffy would like to know what the minimum number of steps are in order to practice turning the words into palindromes.

### Input

The input contains an arbitrary number of test records, but no more than 50.

Each input record consists of a single word on a line by itself, consisting only of the characters $a..z$.

Input is terminated with $-1$ on a line by itself.

It is guaranteed that the sum of the lengths of all words in the input shall not exceed $2 \times 10^6$ characters.

### Output

For each test record, output a single line containing the least number of steps to turn that word into a Palindrome.

### Sample input

```
fluffy
pirate
king
anna
-1
```

### Sample output

```
24
31
9
0
```

### Resource Constraints

- **Time limit**: 1 seconds
- **Memory limit**: 256 MB

### Attribution

This problem was adapted from ECPC 2019.

# 21*st* South African Regional International Collegiate Programming Contest

26 October 2019

## Problem D — Yellow Balloon
## Fibo and Prime

### Problem Description

There are 2 members of the Intimidated Confused Programming Critters team, called Alice and Bob.

After the last session, the following discussion happened between them:

Alice: Do you know the Fibonacci sequence?

Bob: Of course, it is a sequence started by 2 ones and each number after that is computed by summing the previous 2 numbers (i.e. 1, 1, 2, 3, 5, 8, 13, ...). In addition, each number has an index where the first number in the sequence has index 1. As such we can formally define the Fibonacci sequence as:

$$F(n) = \begin{cases} 1, & n \in \{1, 2\} \\ F(n-1) + F(n-2), & n > 2 \end{cases}$$

Alice: Do you know the first 3 prime numbers?

Bob: Sure, they are 2, 3 and 5.

Alice: If I give you 2 indexes (say $A$ and $B$) for 2 numbers in the Fibonacci sequence and give you one of the first 3 prime numbers (say $P$), can you count how many numbers in the Fibonacci sequence between the indexes $A$ and $B$ (inclusive) is divisible by $P$?

Bob: Could you give me an example?

Alice: if $A = 4, B = 7$ and $P = 2$; we will have the following Fibonacci numbers 3, 5, 8, 13 (3 at index 4 ($A$), and 13 at index 7 ($B$)). There is only 1 number (which is 8) that is divisible by 2 ($P$).

Given $A$, $B$ and $P$, could you help Bob answer Alice's question.

### Input

The input contains an arbitrary number of test records, but no more than 100.

Each test record consist of three integers $A$, $B$ and $P$ on a single line such that $1 \leq A \leq B \leq 10^{15}$ and $P \in \{2, 3, 5\}$.

Input is terminated by $-1$ on it's own line.

### Output

For each test record, output a single line containing the count of Fibonacci numbers as from $F(A)$ to $F(B)$ which is divisable by $P$.

### Sample input

```
4 7 2
-1
```

### Sample output

```
1
```

### Resource Constraints

- **Time limit**: 1 seconds
- **Memory limit**: 256 MB

### Attribution

This problem was adapted from ECPC 2019.

# 21*st* South African Regional
# International Collegiate Programming Contest

26 October 2019

## Problem E — Red Balloon
## Palinmorse

### Problem Description

Palindromes are words that are written the same, even when the characters are reversed. For example, *racecar*.

An example of a word that is not a palindrome is *pulp*. Until you write it in Morse code that is.

Morse code replaces each character with a sequence comprising of dots and dashes. Historically these codes were used to transmit short messages over wires.

The full Morse code definition is given by:

| | | | | |
|---|---|---|---|---|
| a ·− | b −··· | c −·−· | d −·· | e · |
| f ··−· | g −−· | h ···· | i ·· | j ·−−− |
| k −·− | l ·−·· | m −− | n −· | o −−− |
| p ·−−· | q −−·− | r ·−· | s ··· | t − |
| u ··− | v ···− | w ·−− | x −··− | y −·−− |
| z −−·· | | | | |

The word *pulp* becomes "·−−· ··− ·−·· ·−−·" when written in Morse code.

If we ignore the spaces this is a palindrome since the sequence of dots (·) and dashes(−) remains the same when reversed.

### Input

The input will consist of an arbitrary number of test cases, but no more than 1000.

Each test case will be a single word consisting of the characters from 'a' to 'z' as give in the Morse code definition above, without any spaces. Words will consist of at least one character, but no more than 100.

The end of input is indicated by a line containing only −1.

### Output

Output "`yes`" on a line if the word is a palindrome in Morse code, otherwise output "`no`".

### Sample input

```
racecar
pulp
-1
```

### Sample output

```
no
yes
```

### Resource Constraints

- **Time limit**: 1 seconds
- **Memory limit**: 256 MB

# 21*st* South African Regional
# International Collegiate Programming Contest

26 October 2019

## Problem F — Blue Balloon
## PrimeRose

### Problem Description

Your computer science teacher, Bruce Schneier, is going on about primes. Apparently they are one of the corner stones of cryptography.

You are tasked with finding all prime numbers in specific ranges. That is, for given boundaries, output the sequence of all prime numbers between these boundaries and inclusive of these boundaries, should they happen to be prime as well.

### Input

The input consists of an arbitrary number of test cases, but no more than 50. Each test case consists of a single line containing two numbers, $A$ and $B$ where $2 < A < B < 2^{32}$. The sequence $[A, B]$ shall contain at least one prime, but not more than 10000.

Input is terminated by $-1$ on its own line.

### Output

For each input pair $A$, $B$, output all prime numbers, separated by a space, on a single line.

### Sample input

```
5 11
20 30
-1
```

### Sample output

```
5 7 11
23 29
```

### Resource Constraints

- **Time limit**: 2 seconds
- **Memory limit**: 256 MB

# 21*st* South African Regional
# International Collegiate Programming Contest

26 October 2019

## Problem G — Green Balloon
## Moriarty's Password

### Problem Description

Sherlock Holmes has lifted Professor James Moriarty's cellular device from the inner breast pocket of his coat and is trying to guess the password to unlock the phone. Sherlock has ascertained that Moriarty's password consists of $N$ characters. Each character is either a lowercase or uppercase version of the 26 letters in the Modern English alphabet or it can be a digit.

Moriarty likes to mock Sherlock, so his password contains exactly $M$ instances of Sherlock's initials, namely *SH*. Sherlock, being the superior mind that he is, already knew this about Moriarty's password. He does, however, need your help in determining exactly how many possible passwords there are that would satisfy this condition in order to show Moriarty what a genius he is when he unlocks the phone using just three tries.

The number of possible passwords containing $M$ instances of Sherlock's initials can be quite large. You are therefore required to print the number of passwords $\bmod 10^9 + 7$.

### Input

The input file will consist of an arbitrary number of test cases but no more than 30.

Each test case will consist of a single line containing two integer values, $N$ and $M$ such that $1 \leq N \leq 10^4$ and $0 \leq M \leq \min\left(100, \left\lfloor \frac{N}{2} \right\rfloor\right)$.

For each test case $N$ indicates the length of Moriarty's password, and $M$ the number of times Sherlock's initials $SH$ appear in the password.

Input is terminated with $-1$ on it's own line.

### Output

For each test case, print the number of possible passwords you calculated, $\bmod 10^9 + 7$.

### Sample input

```
1 0
5 1
10000 100
10000 0
10000 1
200 100
3 1
-1
```

### Sample output

```
62
952940
131867479
716746027
165711702
1
124
```

### Resource Constraints

- **Time limit**: 1 second
- **Memory limit**: 256 MB

### Attribution

This problem was adapted from ECPC 2019.

# 21*st* South African Regional
# International Collegiate Programming Contest

26 October 2019

## Problem H — Orange Balloon
## Black Friday

### Problem Description

Black Friday is the last Friday of November, the day after Thanksgiving in the United States, and marks the beginning of the holiday shopping season. Many stores offer highly promoted sales on Black Friday, not only in the USA but across the world, and open their doors to bargain hunting shoppers soon after the clock has chimed midnight.

One of the more popular offers on Black Friday is the offer where you **Buy** $N$ and **Get** $M$ **Free**. This offer works as follows, for every $N$ items you buy (and pay for), you are allowed to take up to $M$ items free of charge, as long as their normal price is at most that of the cheapest item that's being paid for.

To illustrate how this works, consider the offer of **Buy 2 Get 1 Free** and seven items with prices of 10, 6, 8, 2, 1, 12 , and 5 *cents* respectively. In this scenario, you pay for the items costing 12, 10, 6, 5 and 1 *cents* and get the items costing 8 and 2 *cents* for free. You will therefore pay 34 *cents* for your 7 items instead of 44 *cents* if not on the Black Friday offer.

Your task is to write the algorithm to determine what you as customer will pay on Black Friday if you take $K$ items that are on a **Buy** $N$ **Get** $M$ **free** offer.

### Input

The input consists of an arbitrary number of test records, but not more than 50.

Each test record consists of two lines.

The first line contains three integers $N$, $M$ and $K$ such that $1 \leq N, M \leq 1000$ and $1 \leq K \leq 10^5$.

The second line contains $K$ integers $P_i$, the price of each item (in *cents*) bought where $P_i$ corresponds with the price of item $i$ for $1 \leq i \leq K$.

Input is terminated by $-1$ on it's own line.

### Output

For each test record, output the minimum price that you will pay for the specified offer and number of items.

### Sample input

```
2 1 7
10 6 8 2 1 12 5
1 1 3
12 4 9
1 1 4
6 8 10 2
-1
```

### Sample output

```
34
16
16
```

### Resource Constraints

- **Time limit**: 1 seconds
- **Memory limit**: 256 MB

### Attribution

This problem was adapted from ECPC 2019.

# 21*st* South African Regional International Collegiate Programming Contest

26 October 2019

## Problem I — Black Balloon
## MEXed sums

### Problem Description

Junior read about the MEX value of an array. He found the definition of MEX on Wikipedia to be the "Minimum EXcluded" value from a set.

Junior wants to calculate MEX values of all possible sub-arrays of a permutation $P$ of length $N$ to test his skills in calculating the MEX.

For example, given $N = 4$ and the specific permutation, $P = [2, 1, 0, 3]$, and the sub-array $[1, 0, 3]$ then the MEX for that sub-array is 2. Since we're working with permutations, our set list is the list of all non-negative integer numbers.

Please help Junior by calculating the sum of all MEX values of all possible sub-arrays of $P$.

### Input

The input consists of an arbitrary number of test records, but not more than 100.

Each test record consists of two lines.

The first line contains a single integer $N$ representing the length of the permutation, $1 \leq N \leq 10^5$.

The second line will contain $N$ space seperated distinct integers $P_i$ representing permutation, $0 \leq P_i < N$.

Input is terminated by $-1$ on it's own line.

### Output

For each test record, output a single line containing the sum of the MEX values of all possible sub-arrays of the provided permutation.

### Sample input

```
4
2 1 0 3
7
4 5 6 3 0 1 2
-1
```

### Sample output

```
13
37
```

### Resource Constraints

- **Time limit**: 1 seconds
- **Memory limit**: 256 MB

## Attribution

This problem was adapted from ECPC 2019.

# 21*st* South African Regional International Collegiate Programming Contest

26 October 2019

## Problem J — Gold Balloon Delivery

### Problem Description

A well known delivery company has approached you to automate part of their delivery process. The delivery company receives online orders and then determines which warehouses have the item in stock. You have been tasked to determine which of those warehouses are the closest to where the item is to be delivered. There is only one warehouse per city and the distance travelled within a city can be ignored. Given the cities, the lengths of the roads between them and which cities' warehouses have the item in stock, calculate which warehouse is the closest to the city to which the item is to be delivered.

### Input

The input consists of an arbitrary number of records, but no more than 10.

The first line of each record contains two integers, $C$ ($1 \leq C \leq 100$), the number of cities each numbered from 1 to $C$, and $R$ ($1 \leq R \leq 100$), the number of roads connecting the cities.

The next $R$ lines describe the roads. Each line contains three integers $A$, the index of the first city it connects, $B$, the index of the second city it connects ($1 \leq A, B \leq C$), and $L$ ($1 \leq L \leq 1000$), the length of the road. The roads can be travelled either way, the direction does not matter.

The next line contains one integer $W$ ($1 \leq W \leq C$), the number of cities with a warehouses with the item in stock. The following line contains $w$ integers representing the indices of the cities.

The last line contains one integer, the index of the city to which the item must be delivered.

The end of input is indicated by a line containing only the value $-1$.

### Output

You need to output the index of the city with the warehouse that is closest to the city to which the item must be delivered.

### Sample input

```
5 6
1 2 1
1 3 2
2 4 2
3 4 4
3 5 1
4 5 1
2
2 4
3
-1
```

### Sample output

```
4
```

### Resource Constraints

- **Time limit**: 5 seconds
- **Memory limit**: 256 MB