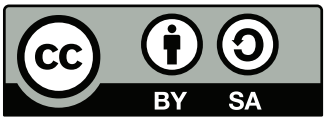
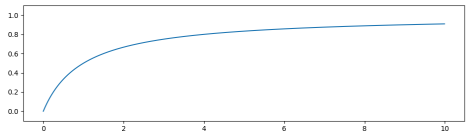
BAPC 2021 Preliminaries

Preliminaries for the 2021 Benelux Algorithm Programming Contest

Problems



A Almost Always



Today, your probability theory class covered probabilistic events that occur *with high probability*. As you are now explaining to your friend, this means that as some parameter n goes to infinity, the probability of this event goes to 1.

An example of an event that occurs with high probability is the following: given a list of n independent uniform random integers between 1 and $2 \cdot 10^9$, there are two elements such that one divides the other.

Your friend does not believe this, as he can easily come up with many examples where no number divides any other number: Just take any subset of $\{10^9 + 1, 10^9 + 2, \ldots, 2 \cdot 10^9\}$ and none of these will divide any other!

To convince your friend, you will show him that you can find two elements that divide each other for exactly 100 instances of the problem. You are confident this will succeed, since the probability of failure is in fact less than 10^{-25} per instance when $n = 5 \cdot 10^5$.

Input

The input consists of:

- One line with an integer n, the number of integers in the list.
- One line containing n independent random integers $a_1, \ldots, a_n$, drawn uniformly from $\{1, 2, \ldots, 2 \cdot 10^9\}$.

Your submission will be run on exactly 100 test cases, all of which will have $n = 5 \cdot 10^5$. The samples are smaller and for illustration only.

Each of your submissions will be run on new random test cases.

Output

Output two distinct integers i and j, indicating that a_i divides a_j.

Sample Input 1	Sample Output 1
10 4 9 8 10 5 28 3 62 9 17	5 4

Sample Input 2	Sample Output 2
10 7 48 17 43 97 26 85 10 26 9	3 7

B Buffered Buffet

CC-BY 2.0 By Mike Fleming on https://flic.kr/p/5NYZPq

Finally, you are allowed again to invite your friends over for a delicious buffet! However, you still need to take social distancing measures into account. To complicate matters further, your friends are coming from various different countries. Each of these countries has slightly different social distancing rules. For example, your friends from the US are required to sit at least 2 meters from another person, while your friends from France only need to sit 1 meter apart from others.

You would like for all your friends to sit around one large circular table. You want to position your friends at this table in such a way that everyone complies with the social distancing requirements of their respective country. To make matters a bit easier, the distance between two people sitting at the table is calculated along the circumference. What is the minimum circumference of the table such that everyone can comply with their social distancing requirement?

Input

The input consists of:

- One line containing an integer n ($2 \leq n \leq 10^5$), the number of people sitting at the table (yourself included).
- One line containing n integers $d_1, \ldots, d_n$ ($1 \leq d_i \leq 10^9$), how many meters each person needs to sit apart from others (measured along the circumference of the table).

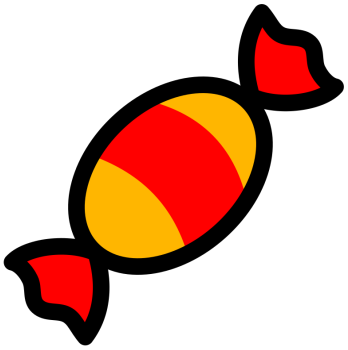
Output

Output the minimum circumference of the table in meters such that everyone can sit at the table while respecting their social distancing measures.

Sample Input 1	Sample Output 1
2 2 3	6

Sample Input 2	Sample Output 2
3 2 2 3	8

C Candy Contribution



CC0 1.0

While you were out travelling, you won the lottery. As it happened, the top prize of this lottery was not cash, but candies! Now you are stuck with a big pile of candies which you would like to take home. Fortunately, you have been able to acquire a truck, so now all you have to do is drive home.

Going from one country to another with such a big pile of candies in your truck is not allowed without paying some taxes. And because everybody likes candies, you are allowed to pay these taxes with candies.

After searching a bit on the internet, you have found a list that tells you exactly which borders you can cross with a truck and for each such border what percentage of tax you have to pay to cross it. You cannot pay with fractional candies and the candies are quite nice, so customs will always round up. You only have to pay taxes on the number of candies you bring across the border.

What is the maximum number of candies you can bring home?

Input

The input consists of:

- One line containing two integers n ($2 \leq n \leq 1 \cdot 10^5$), the number of countries, and m ($1 \leq m \leq 2 \cdot 10^5$), the number of borders.
- One line containing three integers s ($1 \leq s \leq n$), the country where you won the lottery, t ($1 \leq t \leq n$, $t \neq s$), your home country and c ($1 \leq c \leq 10^9$) the number of candies you won in the lottery.
- Then follow m lines containing three integers u, v ($1 \leq u, v \leq n$, $u \neq v$) and p ($0 \leq p \leq 100$) where p is the percentage of tax you have to pay when travelling from country u to v or vice versa.

It is guaranteed you can drive home with your truck, and that each pair of countries is listed at most once.

Output

Output the maximum number of candies you can arrive home with.

Sample Input 1

```
4 4
1 4 1000
1 2 25
2 4 10
1 3 4
3 4 30
```

Sample Output 1

```
675
```

Sample Input 2

```
5 5
1 5 6
1 2 17
2 5 19
1 3 1
3 4 1
4 5 1
```

Sample Output 2

```
3
```

D Dickensian Dictionary

Flying Fingers. CC BY-NC-ND 2.0
By The Hamster Factor on Flickr

You are stuck in your job at the Boring Accountancy Platform Company; the entire day you have to code all kinds of programs that you do not care about. This involves a lot of tedious typing, which you do not want to do. To pass the time, you decide to interact with the words you type more playfully. In particular, you really enjoy it when you type a word with your left and right hand alternating. You dub these words *Dickensian*.

Your mind is quickly overwhelmed with Dickensian words, and even at home they still dictate your thoughts. You want to gather as many Dickensian words as possible and start coding. Given a word, you will need to decide if it is Dickensian or not.

The letters you can type with your left hand are "`qwertasdfgzxcvb`", and the letters you can type with your right hand are "`yuiophjklnm`".

Input

The input consists of:

- One line containing a string of length at least 2 and at most 20, consisting of lowercase characters `a-z`.

Output

Output "`yes`" if the input string is Dickensian, and "`no`" otherwise.

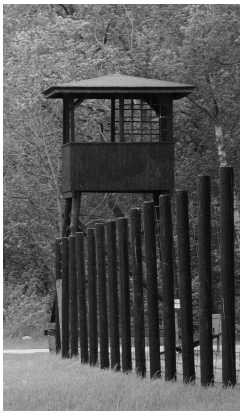
Sample Input 1	Sample Output 1
`dickensian`	`yes`

Sample Input 2	Sample Output 2
`dictionary`	`no`

Sample Input 3	Sample Output 3
`usual`	`yes`

Sample Input 4	Sample Output 4
`suspects`	`no`

E Entering Enemy Encampment



Watchtower by Ragnar Groot Koerkamp

A new two-player game simulates two countries fighting over a territory. This territory contains a number of strategic positions which are suitable for setting up war camps. The players take turns setting up a camp at one of these spots.

In this territory, there are also a number of trails that connect two positions each. Whenever a player sets up a new camp, this player will send a small group of soldiers over every adjacent trail that leads to an enemy encampment. Each of these groups of soldiers will raid the enemy camp. In other words, for every trail, the second player to capture one of the incident positions can launch at most one raid over this trail. When every strategic position has been claimed, the game ends. The player that performed the most raids wins the game.

Assuming both players play optimally, who wins?

Input

The input consists of:

- One line containing two integers n ($1 \leq n \leq 20$), the number of strategic positions, and m ($0 \leq m \leq \frac{n(n-1)}{2}$), the number of trails.
- m lines, each containing two integers a and b ($1 \leq a, b \leq n$, $a \neq b$), indicating that there is a trail between position a and position b.

There is at most one trail between every pair of strategic positions.

Output

If the player who goes first wins, output "`player 1`". If the player who goes second wins, output "`player 2`". If the players tie, output "`tie`".

Sample Input 1	**Sample Output 1**

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

Sample Input 2	**Sample Output 2**
`2 1` `1 2`	`player 2`

Sample Input 3	**Sample Output 3**
5 7 3 4 2 1 4 5 1 4 1 3 2 3 2 4	player 1

F Fridge Distraction

Kevin has an excellent organisation system for his refrigerator: everything is arranged on one long shelf. At the moment, every item is organised in alphabetical order from "a" at the front to "z" at the back.

Kevin's very long fridge

When Kevin wants to get some item from the refrigerator, he simply pulls out everything in front of it, takes the item out, places all the other items back in their original order, and finally places the used item at the front when he is done. This takes a number of seconds equal to the total number of items taken out of the fridge.

You are planning a surprise birthday party for Kevin in which you and his other friends bought him a normal fridge. You will need to keep him busy for exactly the right amount of time – and you will do this by repeatedly asking him to take items out of the fridge.

Given the amount of time you need to waste, in seconds, suggest an order of as little items as possible to be taken out, such that Kevin will take exactly this long.

Input

The input consists of:

- One line containing n ($1 \leq n \leq 26$), the number of items in the refrigerator, and t ($1 \leq t \leq 10000$), the number of seconds you need to waste.

Output

Output the smallest possible number of items you should ask Kevin to take out of the fridge.

Next output one line containing, in order, the letters of all the items you will ask Kevin to take out of the fridge.

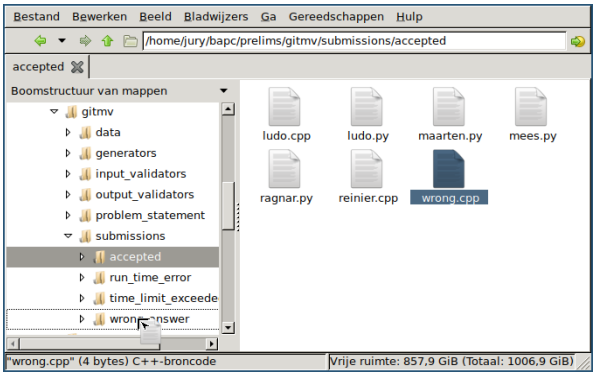
If there are multiple valid solutions, you may output any one of them.

Sample Input 1	Sample Output 1
4 7	2 c d

Sample Input 2	Sample Output 2
3 10	4 b a c b

G `git mv`

During development, you recently moved a file from one location to another. To keep your development team up to date with the change you made, you want to send them a short description of the change, without making use of any versioning software.



Both the source location and destination are valid Unix path names, that is, a nonempty string consisting of lowercase letters and "`/`" such that no "`/`" occurs at the begin or the end, nor does it contain two consecutive forward slashes.

You need to find the shortest string of the form "`A{B => C}D`" such that:

- The source location is "`ABD`" and the destination is "`ACD`", where double forward slashes should be read as one forward slash. For example, if a file is moved from "`a/c`" to "`a/b/c`", we can describe this movement by "`a/{ => b}/c`", meaning the source location was "`a/c`" and not "`a//c`".
- The string A is empty or ends with a forward slash, and similarly D is empty or starts with a forward slash.
- Both B and C do not start or end with a forward slash.

Input

The input consists of:

- One line containing the source location.
- One line containing the destination location.

Both lines will contain at most 10^6 characters, will not begin or end with a forward slash and will not contain any directory name twice. The two strings are guaranteed to be different.

Output

Output the shortest replacement string that transforms the source location to the destination, satisfying the above constraints.

Sample Input 1

```
www/public/passwords
private/passwords
```

Sample Output 1

```
{www/public => private}/passwords
```

Sample Input 2

```
home/linus/downloads/image
home/linus/pictures/recent/image
```

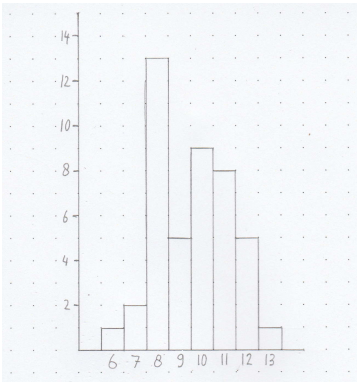
Sample Output 2

```
home/linus/{downloads => pictures/recent}/image
```

H Histogram

The term "histogram" was first introduced in 1895 by Karl Pearson. This is also the year that the portable electric drill was invented. It may not be a coincidence that we speak about "data drilling" today. Most things back then were not electric however, and histograms were still hand crafted. Nowadays, we like things electric and automated. The Bureaucratic Affiliation for Printing Charts needs to create a lot of histograms for a lot of data points, so they come to you for help to automate the printing of histograms.

Given a list of data points, print a histogram.



Manually drawn histogram for the number of problems in the BAPC and its preliminaries, 1997–2020. *If only we could automate this...*

Input

The input consists of:

- One line containing three integers n, s, and k ($1 \leq n, s, k \leq 1000$), the number of bins, the size per bin, and the number of data points, respectively.
- One line containing k integer data points x ($1 \leq x \leq n \cdot s$).

Output

Output the histogram. That is, write a rectangle of characters. The rectangle should be as wide as the number of bins. Each column of the rectangle should have

- a "−" at the bottom;
- on top of that, as many "`#`" as there are items in the bin;
- on top of that, as many "`.`" as are needed to fill out the rectangle.

The rectangle should be just high enough to display a "`#`" for each element in every bin, but no higher.

Sample Input 1	**Sample Output 1**

`3 1 6`
`1 2 3 2 1 2`
	`.#.`
	`##.`
	`###`
	`---`

Sample Input 2	**Sample Output 2**

`3 10 5`
`1 9 11 29 24`
	`#.#`
	`###`
	`---`

I Ice Growth

Ice growth is dependent on temperature. A rule of thumb is that every 5 degrees of frost (on average in a 24-hour period) contributes to 1 cm of ice growth, whilst every 5 degrees above zero removes 1 cm of ice. For example, if there are three days with average temperatures of -3, 1 and -7 degrees Celsius there will be a total of $3 - 1 + 7 = 9$ degrees of frost and thus 1.8 cm of ice growth at the end of day 3. Of course, the ice thickness cannot be negative. If there is enough ice, people can skate on it. The required ice thickness depends on the person, as different persons have different perceptions of safety.

There is currently no ice, but the weather report for the next n days has just come in, and a group of k people wants you to figure out how many of these days they can skate on the ice at the end of the day.

CC0, source: freesvg.org/girl-skating

Input

The input consists of:

- One line containing two integers, n ($1 \leq n \leq 10^5$) the number of days and k ($1 \leq k \leq 10^5$) the number of people.
- One line with n integers $a_1, \ldots, a_n$ ($-10^6 \leq a_i \leq 10^6$ for all i), the average temperature on day i.
- One line with k integers $b_1, \ldots, b_k$ ($1 \leq b_j \leq 10^6$ for all j), the required minimal ice thickness in cm before person j can skate on the ice.

Output

Output a line with k integers $c_1, \ldots, c_k$, where c_j is the number of days that person j can skate on the ice at the end of the day.

Sample Input 1	Sample Output 1
3 2 -3 1 -7 1 2	1 0

Sample Input 2	Sample Output 2
5 3 -5 -5 15 -5 -5 1 2 3	4 2 0

J Jack the Mole


CC BY-SA 3.0 By Kenneth Catania

The spy agency you are employed by[1] has sent you and some fellow spies on an important mission deep in the jungles of Luxembourg. Your colleagues and you are currently waiting on an abandoned and overgrown airstrip, about to be picked up by the finest spy plane the Dutch bureaucracy can buy.

However, an urgent communication from the HQ has just come in to inform you counterintelligence has learned that during the mission, an infamous counter-spy called Jack the Mole was planted among you! You do a headcount and to your horror realize that while you started the mission with $n-1$ people, there are now $n!$[2] Since you all take Covid regulations very seriously, you all wore masks, so it is not obvious who the imposter is.

Then you remember: the finest spy plane in question is thrown off balance rather easily, so the agency picked the $n-1$ agents so that it is possible to divide them into two groups of equal weight, but not necessarily of equal size, for the left and the right side of the plane respectively. However, it did not specify exactly which two groups (anonymity and all that). Of course you have a scale with you (doesn't any good spy?), so you can find which of the n people on the mission is potentially the mole.

Input

The input consists of:

- One line containing the number of spies present, n ($3 \leq n \leq 300$).
- One line containing n space separated integers $w_1, \ldots, w_n$ ($1 \leq w_i \leq 1000$), the ith of which gives the weight of the ith spy in kilograms[3].

Output

Output a single number k, the number of spies that are potentially the mole, followed by k integers, the suspects, in increasing order.

Sample Input 1	Sample Output 1
3 1 1 2	1 3

Sample Input 2	Sample Output 2
3 1 1 1	3 1 2 3

[1] When your family members ask, please lie to them and say you work as a computer programmer.
[2] Exclamation point, not factorial.
[3] If those ranges seem odd, recall spies can be very muscular or very nimble.

Sample Input 3

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

Sample Output 3

```
3
1 4 5
```

Sample Input 4

```
5
1 1 1 3 1
```

Sample Output 4

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

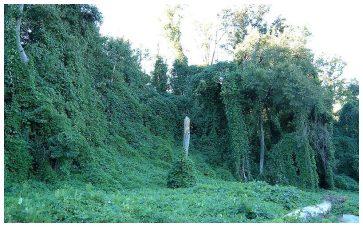
Sample Input 5

```
4
2 1 5 3
```

Sample Output 5

```
2
2 3
```

K Kudzu Kniving



Kudzu overgrowing trees
By Scott Ehardt on Wikipedia

You can't deny it anymore: the kudzu vines in your garden have grown out of control. Some years ago, you planted a single seedling which you received as a gift from your Mathematics teacher. You vaguely remember her explaining how it grows:

- It started as a single root vertex, labelled 0.
- Every year, it grows some new vertices and edges: if at the start of the year it has n vertices, then during this year, it will grow a new edge and vertex from each of the n vertices. If an old vertex had index v, then the new vertex which grows from it will have index $v + n$.
- It can be shown that after i years, your kudzu plant has exactly 2^i vertices, numbered 0 to $2^i - 1$.

Today, it is time to get out your machete knife and remove a number of branches (subtrees) from the kudzu, one by one. You plan on pruning the tree in a fancy shape, which will probably not stay intact for a long time given how fast your kudzu is growing, but at least you promise yourself to keep pruning every year. After deciding which branches you want to cut off today, you call up the Branching And Pruning Company to ask if they can dispose of the plant waste. They want to know exactly how much they need to clean up. You feel like you should be able to compute that, but how?

When a subtree rooted at some vertex v is removed, this means that vertex v will be removed, together with all vertices which have grown from it (and the vertices which have grown from those, and so on). Figure K.1 shows this process for the second sample case.

Given the indices of the roots of the subtrees which you will remove, compute the number of vertices which will be removed for each of these removed subtrees. Since these numbers may be large, you should find them modulo $10^9 + 7$.

Input

The input consists of:

- One line containing two integers a ($0 \leq a \leq 10^6$), the age of the tree in years, and m ($1 \leq m \leq 10^5$), the number of subtrees which will be removed.
- m lines, each with an integer v ($0 \leq v \leq 10^9$), the index of a vertex to be removed from the tree. It is guaranteed that v will not yet have been removed.

Output

Output m lines. The ith line should contain the number of vertices removed in the ith removal, modulo $10^9 + 7$.

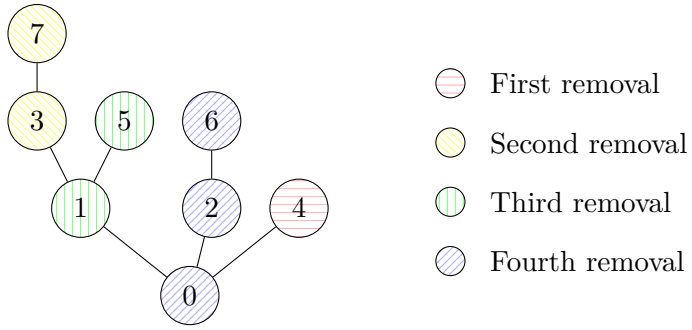


Figure K.1: The tree of the second sample case. The different colours indicate in which removal the vertices are removed.

Sample Input 1

```
4 1
0
```

Sample Output 1

```
16
```

Sample Input 2

```
3 4
4
3
1
0
```

Sample Output 2

```
1
2
2
3
```

Sample Input 3

```
5 5
6
3
1
18
2
```

Sample Output 3

```
4
8
8
1
3
```

Sample Input 4

```
42 1
0
```

Sample Output 4

```
46480318
```