A. Gasoline

time limit per test: 15 s
memory limit per test: 1024 MB

After the end of the truck drivers' strike, you and the rest of Nlogônia logistics specialists now have the task of planning the refueling of the gas stations in the city. For this, we collected information on stocks of R refineries and about the demands of P gas stations. In addition, there are contractual restrictions that some refineries cannot supply some gas stations; When a refinery can provide a station, the shorter route to transport fuel from one place to another is known.

The experts' task is to minimize the time all stations are supplied, satisfying their demands. The refineries have a sufficiently large amount of trucks, so that you can assume that each truck will need to make only one trip from a refinery to a gas station. The capacity of each truck is greater than the demand of any gas station, but it may be necessary to use more than one refinery.

Input

The first line of the input contains three integers P, R, C, respectively the number of gas stations, the number of refineries and the number of pairs of refineries and gas stations whose time will be given ($1 \leq P, R \leq 1000$; $1 \leq C \leq 20000$). The second line contains P integers D_i ($1 \leq D_i \leq 10^4$), representing the demands in liters of gasoline of the gas stations $i = 1, 2, \ldots, P$, in that order. The third line contains R integers E_i ($1 \leq E_i \leq 10^4$), representing stocks, in liters of gasoline, of refineries $i = 1, 2, \ldots, R$, in that order. Finally, the latest C lines describe course times, in minutes, between stations and refineries. Each of these rows contains three integers, I, J, T ($1 \leq I \leq P$; $1 \leq J \leq R$; $1 \leq T \leq 10^6$), where I is the ID of a post, J is the ID of a refinery and T is the time in the course of a refinery truck J to I. No pair (J, I) repeats. Not all pairs are informed; If a pair is not informed, contractual restrictions prevents the refinery from supplying the station.

Output

Print an integer that indicates the minimum time in minutes for all stations to be completely filled up. If this is not possible, print −1.

Examples

input
```
3 2 5
20 10 10
30 20
1 1 2
2 1 1
2 2 3
3 1 4
3 2 5
```
output
```
4
```

input
```
3 2 5
20 10 10
25 30
1 1 3
2 1 1
2 2 4
3 1 2
3 2 5
```
output
```
5
```

input
```
4 3 9
10 10 10 20
10 15 30
1 1 1
1 2 1
2 1 3
2 2 2
3 1 10
3 2 10
4 1 1
4 2 2
4 3 30
```
output
```
-1
```

input
```
1 2 2
40
30 10
1 1 100
1 2 200
```
output
```
200
```

B. Hacking Passwords

time limit per test: 1 second
memory limit per test: 256 MB

You are part of an elite team of hackers attempting to gain access to information that could be used to influence the next University of Melbourne Student Union (UMSU) election. It is known that UMSU has a policy on password security, such that all passwords used must satisfy the following rules:

1. A password must be at least six characters long.
2. A password must contain at least one lowercase letter and one uppercase letter.
3. A password must contain at least one numeric digit.
4. A password must contain at least one of the following punctuation characters: Full stop (.), Comma (,), Colon (:), Semi-colon (;), Question mark (?), Exclamation mark (!).

This makes finding their passwords via brute force extremely difficult. Luckily, you have managed to sneak a keylogger onto one of their machines and have tracked every string that they have typed. Given a list of strings recorded by the keylogger, report the ones that could potentially be a password as per the rules above.

Input

The first line of input contains a single integer n $(1 \leq n \leq 100)$, which is the number of strings recorded by the keylogger. The next n lines are the strings recorded by the keylogger. Each string is of length between 1 and 20, inclusive. Each string will contain only alphanumeric characters and the punctuation listed above.

Output

Display the number of strings recorded that could potentially be a password. Then display the strings in the same order that they appeared in the input. If the same string is recorded multiple times and is a potential password, it should be counted and displayed multiple times.

Examples

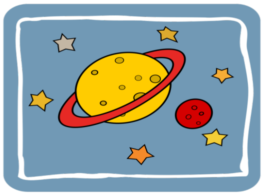
input
```
4
hello
world
Password1!
hunter2
```
output
```
1
Password1!
```

input
```
9
MelbourneRulez22;
google.com
Is
Melbourne
better
than
Monash?
JustLetMeIn42.
Bye
```
output
```
2
MelbourneRulez22;
JustLetMeIn42.
```

input
```
13
Password1!
how
to
avoid
getting
hacked
Password1!!!
I
think
I
was
hacked
Password1!
```
output
```
3
Password1!
Password1!!!
Password1!
```

C. Guardians of the Galaxy

time limit per test: 5.0 s
memory limit per test: 256 MB



Captain Picard has been given the task of protecting a large set of planets. The planets all have an (x, y) coordinate. R2D2 (an android) has positioned Picard's ship at a position so that the distance to the farthest planet in the galaxy is as small as possible.

Find the distance to the farthest planet in the galaxy. Note that the ship may be positioned on a planet.

Input

The first line of the input contains a single integer n $(1 \leq n \leq 200)$, which is the number of planets in the galaxy.

The next n lines describe the location of the planets. Each of these lines contain two integers x $(-10^4 \leq x \leq 10^4)$ and y $(-10^4 \leq y \leq 10^4)$, which represent the coordinates of the planet.

Output

Display the distance to the farthest planet. Your answer should have an absolute or relative error of less than 10^{-6}.

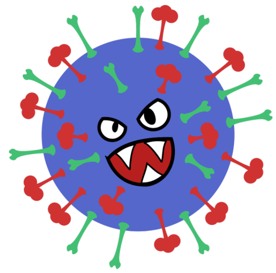
Examples

input
```
4
0 0
0 1
1 0
1 1
```
output
```
0.7071067812
```

input
```
3
10 10
11 20
5 -4
```
output
```
12.3693168769
```

D. Environmental Concern IV

time limit per test: 3 seconds
memory limit per test: 1024 MB



Each year at VCPC, we devote one problem to help save the environment. In 2014, we helped a group of environmentalists save a rainforest. In 2015, we saved a rare species of grass from extinction. And last year, we stopped the carbon emissions from all of Evil-Corp's buildings. This year, we need to help control the population of the highly infectious *globular dencrompticus*!

The globular dencrompticus' population grows very rapidly in a predictable way. On day 0, there are no globular dencrompticus. On day i, exactly a_i new globular dencrompticus are born. On day 1, there are exactly a_1 globular dencrompticus. On day 2, a_2 new globular dencrompticus are born (that is, there are now $a_1 + a_2$ globular dencrompticus total). On every other day i, $a_i = a_{i-1} + a_{i-2}$. For example, if $a_1 = 1$ and $a_2 = 3$, then the population on the first 5 days is 1, 4, 8, 15 and 26, respectively.

To help with this infestation, we have built a machine which can kill groups of m globular dencrompticus at a time. Note that the machine only works if exactly m globular dencrompticus are put into the machine. The value of m can be changed on the machine to any integer from L to R, inclusive. However, as soon as the machine has been used once, the value of m cannot be changed. So the goal is to choose a value for m, then use the machine as many times as possible so that the fewest number of globular dencrompticus remain.

The value of m can dramatically affect the number of surviving globular dencrompticus. For example, if $a_1 = 1$ and $a_2 = 1$, then on day 7, there would be 33 globular dencrompticus. If you set the machine to $m = 9$, then you could use the machine 3 times leaving 6 globular dencrompticus. However, if you set the machine to $m = 8$, then you could use the machine 4 times leaving only 1 globular dencrompticus.

Given the day in which you will use the machine and the acceptable range for m, determine the fewest number of globular dencrompticus that could survive after using the machine.

Input

The first line of input contains three integers a_1 $(0 \le a_1 \le 10^9)$, a_2 $(0 \le a_2 \le 10^9)$, which are the number of globular dencrompticus born on day one and two, and d $(1 \le d \le 10^{18})$, which is the day that we will use the machine.

The second line of input contains two integers L and R $(1 \le L \le R \le 10\,000)$, which are the bounds on which values of m can be used in the machine.

Output

Display the fewest number of globular dencrompticus that could survive.

Examples

input	output
1 1 7 8 9	1
1 3 4 100 100	15
5 17 40 1500 2000	0

E. Array Beauty

time limit per test: 5 seconds

memory limit per test: 256 megabytes

Let's call beauty of an array $b_1, b_2, \ldots, b_n$ $(n > 1)$ — $\min\limits_{1 \le i < j \le n} |b_i - b_j|$.

You're given an array $a_1, a_2, \ldots a_n$ and a number k. Calculate the sum of beauty over all subsequences of the array of length exactly k. As this number can be very large, output it modulo 998244353.

A sequence a is a subsequence of an array b if a can be obtained from b by deletion of several (possibly, zero or all) elements.

Input

The first line contains integers n, k ($2 \le k \le n \le 1000$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^5$).

Output

Output one integer — the sum of beauty over all subsequences of the array of length exactly k. As this number can be very large, output it modulo 998244353.

Examples

input	output
4 3 1 7 3 5	8

input	output
5 5 1 10 100 1000 10000	9

Note

In the first example, there are 4 subsequences of length 3 — $[1, 7, 3]$, $[1, 3, 5]$, $[7, 3, 5]$, $[1, 7, 5]$, each of which has beauty 2, so answer is 8.

In the second example, there is only one subsequence of length 5 — the whole array, which has the beauty equal to $|10 - 1| = 9$.

F. The Number of Products

time limit per test: 2.0 s
memory limit per test: 256 MB

You are given a sequence $a_1, a_2, \dots, a_n$ consisting of n integers.

You have to calculate three following values:

1. the number of pairs of indices (l, r) $(l \le r)$ such that $a_l \cdot a_{l+1} \dots a_{r-1} \cdot a_r$ is negative;
2. the number of pairs of indices (l, r) $(l \le r)$ such that $a_l \cdot a_{l+1} \dots a_{r-1} \cdot a_r$ is zero;
3. the number of pairs of indices (l, r) $(l \le r)$ such that $a_l \cdot a_{l+1} \dots a_{r-1} \cdot a_r$ is positive;

Input

The first line contains one integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of elements in the sequence.

The second line contains n integers $a_1, a_2, \dots, a_n$ $(-10^9 \le a_i \le 10^9)$ — the elements of the sequence.

Output

Print three integers — the number of subsegments with negative product, the number of subsegments with product equal to zero and the number of subsegments with positive product, respectively.

Examples

input	output
5 5 -3 3 -1 0	6 5 4

input	output
10 4 0 -4 3 1 2 -4 3 0 3	12 32 11

input	output
5 -1 -2 -3 -4 -5	9 0 6

Note

In the first example there are six subsegments having negative products: $(1, 2)$, $(1, 3)$, $(2, 2)$, $(2, 3)$, $(3, 4)$, $(4, 4)$, five subsegments having products equal to zero: $(1, 5)$, $(2, 5)$, $(3, 5)$, $(4, 5)$, $(5, 5)$, and four subsegments having positive products: $(1, 1)$, $(1, 4)$, $(2, 4)$, $(3, 3)$.