ACM ICPC SOUTH PACIFIC REGIONAL FINALS

NOVEMBER 25, 2017

Contest Problems

A: Andrew’s Amazing Architecture
B: Banner
C: Catan’s Longest Road
D: Drawn and Quartered
E: Extending Rock-Paper-Scissors
F: Fair Share
G: Grievous Loss of Data
H: Hiding Merlin
I : Injecting DNA
J : Judge’s Mistake
K: Kiwis vs Kangaroos II
L: Last Casino

This contest contains twelve problems over 28 pages. The top teams will advance to the World Finals. Good luck.

Definition 1

For problems that ask for a result modulo m:
If the correct answer to the problem is the integer b, then you should display the unique value a such that:

- $0 \leq a < m$
  and
- $(a - b)$ is a multiple of m.

Definition 2

A string $s_1 s_2 \cdots s_n$ is lexicographically smaller than $t_1 t_2 \cdots t_\ell$ if

- there exists $k \leq \min(n, \ell)$ such that $s_i = t_i$ for all $1 \leq i < k$ and $s_k < t_k$
  or
- $s_i = t_i$ for all $1 \leq i \leq \min(n, \ell)$ and $n < \ell$.

Definition 3

- Uppercase letters are the uppercase English letters $(A, B, \ldots, Z)$.
- Lowercase letters are the lowercase English letters $(a, b, \ldots, z)$.

Judging Team

Daniel Anderson, Darcy Best, Mike Cameron-Jones, Malcolm Corney, Zac Forman, Tim French, Andrew Gozzard, Walter Guttmann, Andrew Haigh, Richard Lobb, Evgeni Sergeev, Kevin Tran, Max Ward-Graham, Peter Whalan

Problem A
Andrew's Amazing Architecture
Time limit: 3 seconds

Aaron has a large supply of blocks and has challenged Andrew to build a structure out of them. All of the blocks are $k \times 1 \times 1$ for various values of k. The structure must be made up of n nonempty columns lined up in a sequence such that all blocks in column i have dimensions $h_i \times 1 \times 1$, and have a 1×1 face that is parallel to the ground. Moreover, the structure must be a *pyramid*. A pyramid must contain an *apex column* such that for each column j to the left of the apex, the height of column j is no more than the height of column $j + 1$ and for each column k to the right of the apex, the height of column k is no more than the height of column $k - 1$. For example, the left structure in Figure A.1 is not a pyramid since it does not have an apex column, while the right structure is a pyramid because the third column from the left is an apex column (as is the fourth column from the left).

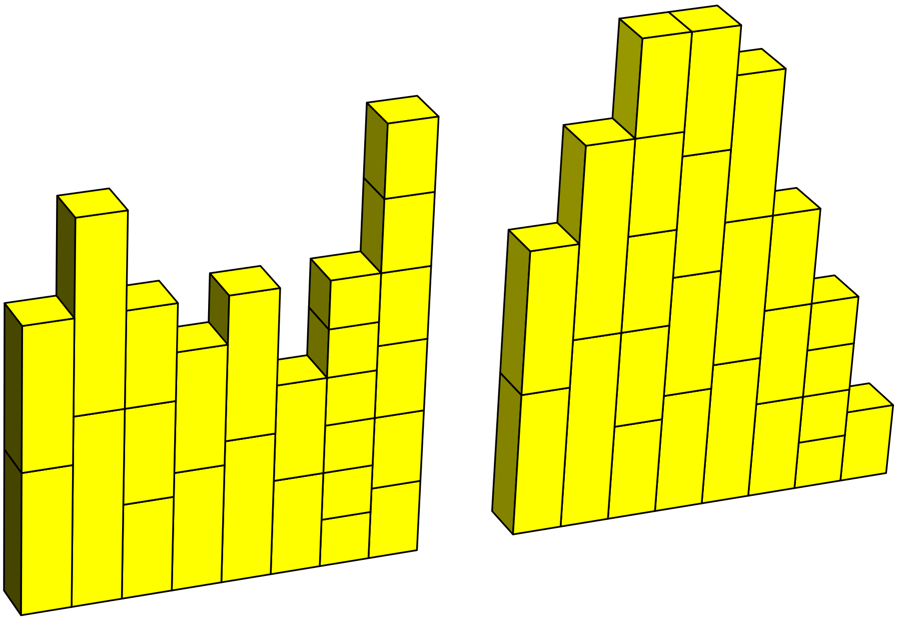


Figure A.1: (left) An example that is not a pyramid. (right) An example of a pyramid. In both cases, $n = 8$ with blocks of size $6, 8, 4, 5, 6, 4, 2, 3$ in the columns from left to right. This sequence is given in Sample Input 3.

Of course, just building a pyramid is easy, so Aaron has asked Andrew to find the pyramid with the smallest volume given a sequence of block sizes to use. Help Andrew by determining the smallest volume possible. You may assume that there is an unlimited supply of blocks of each size.

Input

The input starts with a line containing a single integer n ($1 \leq n \leq 200\,000$), which is the length of the sequence.

The second line describes the blocks. This line contains n integers $h_1, h_2, \ldots, h_n$ ($1 \leq h_i \leq 100\,000$), denoting that the blocks used in column i must be $h_i \times 1 \times 1$.

Output

Display the smallest volume of a pyramid.

Sample Input 1	Sample Output 1
1 1337	1337

Sample Input 2	Sample Output 2
3 99 15 11	125

Sample Input 3	Sample Output 3
8 6 8 4 5 6 4 2 3	49

Problem B
Banner
Time limit: 6 seconds

With a day to go until the cheerful parade in honour of Chairman Äčm's jubilee, your banner factory received a last-minute order! A very important order. You can vividly imagine your kids growing up in an orphanage if this banner is not ready on time. (Or perhaps that's just paranoia. Like Chairman Äčm famously said: "Our country is blessed with large numbers, and astonishing coincidences.")

Your factory has 26 machines, each of which is specialised for rapidly sewing a unique uppercase letter onto fabric. The banner is a string of uppercase letters.

At the beginning of the day, you will arrange the 26 machines into a line. Any nonempty substring of this line may be sewn onto the banner (there are $\frac{26 \cdot 27}{2} = 351$ nonempty substrings). Then the workers will make the banner. Each hour, they will start at the leftmost unsewn letter and sew the longest possible substring of the banner starting at that position. Each substring takes exactly one hour to sew and only one substring may be sewn at a time.

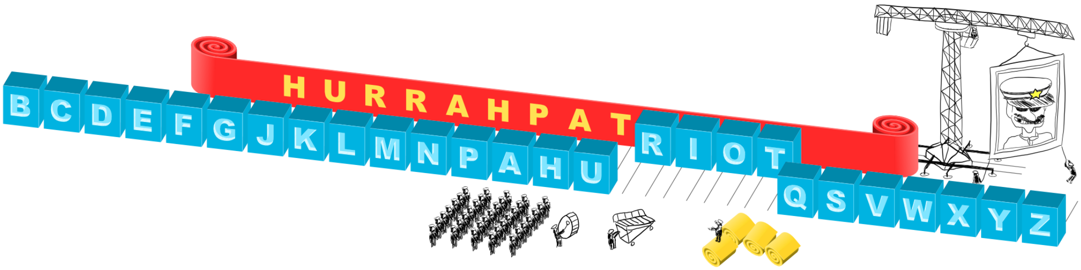


Figure B.1: One of the optimal permutations for Sample Input 2.

For example, consider the permutation in Figure B.1. If the string to be sewn is `HURRAHPATRIOT`, then the banner will be completed in 6 hours (`HUR` in the first hour, `R` in the second, `AH` in the third, `PA` in the fourth, `T` in the fifth and `RIOT` in the sixth).

A permutation of the machines is optimal if no other permutation can be used to complete the banner in fewer hours. How many permutations are optimal?

Input

The input consists of a single line containing the string to be sewn onto the banner. The string uses only uppercase letters and consists of at least 1 and at most 35 characters.

Output

Display the number of optimal permutations modulo each of Chairman Äčm's four favourite prime numbers: 1 000 000 007, 1 000 000 009, 1 000 000 021 and 1 000 000 033.

Sample Input 1	Sample Output 1
`ABCDEFGHIJKLMNOPQRSTUVWXYZ`	`1 1 1 1`

Sample Input 2	Sample Output 2
`HURRAHPATRIOT`	`557316307 314026109 854284963 394543832`

Sample Input 3	Sample Output 3
`THISISNOTAHINT`	`731630315 402610621 428494515 454381913`

This page is intentionally left (almost) blank.

Problem C
Catan's Longest Road
Time limit: 4 seconds

In the game *Settlers of Catan*, several players compete to gain the most points by building items on a hexagonal grid. You and two friends are playing on the grid shown in Figure C.1. The board consists of seven regular hexagons which define 24 *intersections* (vertices of the hexagons) and 30 *lanes* (edges of the hexagons) connecting them.

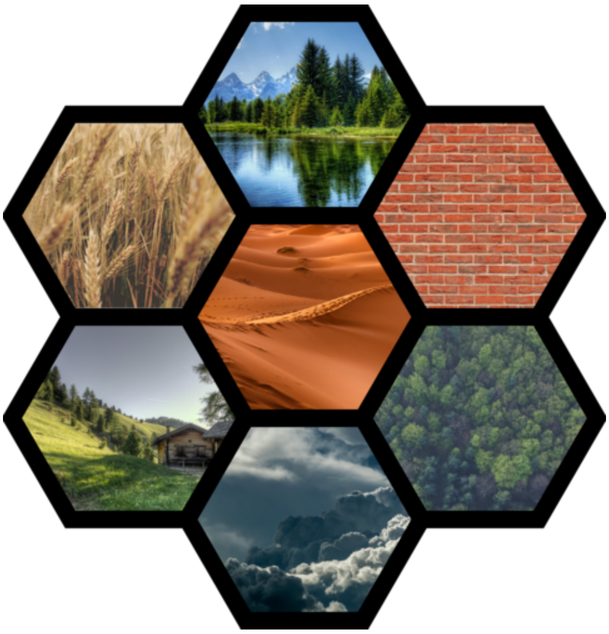


Figure C.1: The game board.

In the game, each player can build road segments on the lanes. Each road segment is owned by exactly one player. At most one road segment can be built on each lane. A player has a road of length k if there is a sequence of $k+1$ (not necessarily distinct) intersections $I_0, I_1, \ldots, I_k$ such that the player owns k distinct road segments that connect intersections I_j and I_{j+1} (for each $j \in \{0, 1, \ldots, k-1\}$).

There is a reward for the player who has the longest road. What is the length of each player's longest road?

Input

The input consists of a text representation of the game board. The game board consists of 19 lines and has size 19×31. Each intersection is represented by one of '>' or '<'. If a lane does not have a road segment on it, then all characters on that lane are one of '-', '/' or '\'. If a lane does have a road segment on it, then all characters on that lane are one of '1', '2' or '3', representing the player that owns the road segment on that lane. All other characters of the input are '.'.

The game board with no road segments is shown in the first sample input. It is guaranteed that any differences between that empty game board and the input are full lanes replaced with one of '1', '2' or '3'.

Output

Display the length of the longest road owned by each of the three players (Player 1 first, then Player 2, then Player 3).

Sample Input 1

```
............>-----<............
.........../.......\...........
........../.........\..........
...>-----<...........>-----<...
../.......\........./.......\..
./.........\......./.........\.
<...........>-----<...........>
.\........./.......\........./.
..\......./.........\......./..
...>-----<...........>-----<...
../.......\........./.......\..
./.........\......./.........\.
<...........>-----<...........>
.\........./.......\........./.
..\......./.........\......./..
...>-----<...........>-----<...
..........\........./..........
...........\......./...........
............>-----<............
```

Sample Output 1

```
0 0 0
```

Sample Input 2

```
............>-----<............
.........../.......\...........
........../.........\..........
...>11111<...........>-----<...
../.......1........./.......\..
./.........1......./.........\.
<...........>11111<...........>
.\.........2.......\........./.
..\.......2.........\......./..
...>22222<...........>-----<...
../.......\........./.......\..
./.........\......./.........\.
<...........>-----<...........>
.\........./.......\........./.
..\......./.........\......./..
...>-----<...........>-----<...
..........\........./..........
...........\......./...........
............>11111<............
```

Sample Output 2

```
3 2 0
```

Sample Input 3

```
............>-----<............
.........../.......2...........
........../.........2..........
...>-----<...........>-----<...
../.......\........./.......\..
./.........\......./.........\.
<...........>-----<...........>
.\........./.......\........./.
..\......./.........\......./..
...>-----<...........>-----<...
../.......\........./.......\..
./.........\......./.........\.
<...........>-----<...........>
.\.........1.......\........./.
..\.......1.........\......./..
...>-----<...........>-----<...
..........\.........3..........
...........\.......3...........
............>-----<............
```

Sample Output 3

```
1 1 1
```

Sample Input 4

```
............>11111<............
...........1.......1...........
..........1.........1..........
...>11111<...........>11111<...
..1.......1.........1.......1..
.1.........1.......1.........1.
<...........>11111<...........>
.1.........1.......1.........1.
..1.......1.........1.......1..
...>11111<...........>11111<...
..1.......1.........1.......1..
.1.........1.......1.........1.
<...........>11111<...........>
.1.........1.......1.........1.
..1.......1.........1.......1..
...>11111<...........>11111<...
..........1.........1..........
...........1.......1...........
............>11111<............
```

Sample Output 4

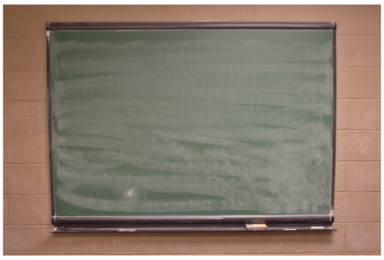
```
25 0 0
```

This page is intentionally left (almost) blank.

Problem D
Drawn and Quartered
Time limit: 2 seconds

You are playing a game which involves drawing and redrawing a string on a blackboard. You start with a string S of length N and perform a switcheroo on the string exactly K times. A switcheroo involves breaking S into quarters, and then moving the middle two quarters to the end of S without changing their relative order to each other. For example, say you start with `aabbccdd`. After a single switcheroo, the string would become `aaddbbcc`. After another switcheroo, you would have `aaccddbb`, and so on.

Given some starting string S and the number of times you should perform a switcheroo, what is the final state of the string?


Source: Pexels

Input

The input starts with a line containing two integers N ($4 \leq N \leq 100\,000$), which is the length of the string, and K ($1 \leq K \leq 10^{18}$), which is the number of times you should perform a switcheroo to S. It is guaranteed that N is a multiple of 4.

The second line contains S. The string S contains only lowercase letters and consists of exactly N characters.

Output

Display the string after performing K switcheroos.

Sample Input 1	Sample Output 1
4 2 abcd	acdb

Sample Input 2	Sample Output 2
8 1 abcdefgh	abghcdef

Sample Input 3	Sample Output 3
20 26 southpacificregional	southicregionalpacif

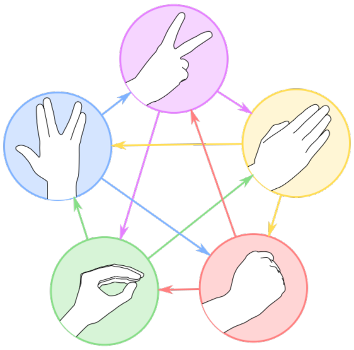
This page is intentionally left (almost) blank.

Problem E
Extending Rock-Paper-Scissors
Time limit: 2 seconds

Rock-paper-scissors is a game played between two players, where each player chooses one of three elements: rock, paper or scissors. The rules are simple: rock-beats-scissors, scissors-beats-paper and paper-beats-rock. If the players choose the same element, then they tie. On the television show *The Big Bang Theory*, Sheldon extended rock-paper-scissors to include two extra elements: Lizard and Spock.

The rules are scissors-cuts-paper, paper-covers-rock, rock-crushes-lizard, lizard-poisons-Spock, Spock-smashes-scissors, scissors-decapitates-lizard, lizard-eats-paper, paper-disproves-Spock, Spock-vaporizes-rock and rock-crushes-scissors.

Sheldon's setup is valid since each element beats exactly half of the other elements and loses to the remaining half. Whenever the number of elements is odd, it is possible to find a game that satisfies these criteria. Extend the game to n elements.



Source: Wikipedia

Input

The input consists of a single line with one integer n ($3 \le n \le 99$), which is the number of elements. It is guaranteed that n is odd.

Output

Display one valid extension of rock-paper-scissors to n elements. Display exactly $\binom{n}{2} = \frac{n(n-1)}{2}$ lines containing two integers each, b and c ($b \neq c$), indicating that element b beats element c. The elements are numbered $1, 2, \ldots, n$.

For each pair of distinct elements, x and y, exactly one of 'x y' or 'y x' should be displayed. The lines may be displayed in any order. If there are multiple solutions, any one will be accepted.

Sample Input 1	Sample Output 1

3	1 2
	2 3
	3 1

Sample Input 2	Sample Output 2

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

This page is intentionally left (almost) blank.

Problem F
Fair Share
Time limit: 5 seconds

Source: Wayne England, Flickr CC BY 2.0

Scientists have discovered vast water sources in the Great Victoria Desert. Representatives from Western Australia and South Australia—the two states sharing the desert—have agreed to divide the water sources in a fair way. They propose to map the water sources, to draw a straight line through the centre of the map and to assign each state the sources on one side of the line. The centre of the map is located at $(0, 0)$.

The benefit for a state is the sum of the values of the water sources in its half minus the sum of the costs for infrastructure development in the same half. The costs are also given on the map. The state representatives ask you to compute a fair division, which minimises the difference between the benefits for the two states. The dividing line must not pass directly through any of the locations in question.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$), which is the number of water sources and infrastructure locations. Each of the next n lines contains three integers x_i, y_i and v_i ($-10^9 \leq x_i, y_i, v_i \leq 10^9$). They represent the location (x_i, y_i) of a water source with value v_i (if $v_i \geq 0$) or an infrastructure location whose development costs $-v_i$ (if $v_i < 0$). Locations are pairwise distinct and different from $(0, 0)$.

Output

Display the minimum absolute difference between the benefits for the two states.

Sample Input 1	Sample Output 1

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

Sample Input 2	Sample Output 2

3
-1 -1 100
0 1 101
1 -1 102
	99

Sample Input 3	Sample Output 3

5
1 1 5
1 0 5
1 -1 5
-1 0 15
-1 1 -2
	2

This page is intentionally left (almost) blank.

Problem G
Grievous Loss of Data
Time limit: 6 seconds

Something has gone very wrong with the lecture scheduling system at the University of Competitive Programming (UCP), which has resulted in a large loss of data. The lecture scheduling system's job is to assign lecture halls to lectures. Unfortunately, the algorithm that was implemented to do this task has gone completely haywire, and has started displaying a very convoluted error message. No one has been able to decipher the error message, but everyone agrees it seems to be some sort of very long, colourful, and immersive proof. Luckily, some of the information has been recovered.

From the recovered data, we have determined that there are N lectures today, each of which takes up a single, contiguous interval of time. Some of these lectures may clash, which means that their time intervals intersect. Unfortunately, the information describing which interval of time each lecture requires has been lost! Fortunately, we have been able to determine which pairs of lectures clash.

If two lectures clash, they need to be assigned to different lecture halls. At the UCP, there is a limited number of lecture halls, and all the lectures must be assigned a hall. People will start arriving for the lectures soon. Can you help to find the minimum number of lecture halls needed such that every lecture can be assigned to a hall?

Input

The first line of input contains two integers N ($1 \leq N \leq 200\,000$), which is the number of lectures, and M ($0 \leq M \leq 200\,000$), which is the number of clashes.

The next M lines describe the clashes. Each of these lines contains two integers u and v ($1 \leq u < v \leq N$), which describe a pair of lectures that clash. It is guaranteed that there is a set of N lectures that have exactly the M clashes given. Each clash is unique and will appear exactly once in the input.

Output

Display the minimum number of lecture halls required.

Sample Input 1	Sample Output 1
3 0	1

Sample Input 2	Sample Output 2

5 6
1 2
2 3
1 4
1 5
2 4
2 5
	3

Sample Input 3	Sample Output 3

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

This page is intentionally left (almost) blank.

Problem H
Hiding Merlin
Time limit: 4 seconds

Source: 4teachers

The war is coming to an end. King Arthur and his loyal servants have grouped together to eradicate Mordred and his minions. If all goes well, the war will be over by tomorrow. The only thing that now worries King Arthur is the news that an assassin is out to get Merlin.

King Arthur has decided to hide Merlin in one of the billion houses in Camlann. The houses in Camlann are numbered 1, 2, ..., 999 999 999, 1 000 000 000. Because King Arthur does not want to forget which house he has hid Merlin in, he would like to write it down. However, King Arthur is worried about security, so he is going to encrypt the house number. But since this is the fifth century AD, the encryption that he will use is quite primitive. He first writes the number down as a sum of positive square numbers, then concatenates those squares together and writes down that string.

For example, if the house number is 46, then he could write down `3691` since $46 = 36+9+1 = 6^2+3^2+1^2$. King Arthur could also write down `1369` ($46 = 1+36+9$) or `1619416` ($46 = 16+1+9+4+16$). King Arthur writes each of the squares with no leading zeroes.

What is the smallest house number in Camlann that is consistent with the encrypted house number that King Arthur wrote down?

Input

The input consists of a single line containing a string, which is the encrypted house number. The encrypted house number contains only digits (`0`, `1`, ..., `9`) and has length at least 1 and at most 100 000.

Output

Display the smallest house number in Camlann that is consistent with the encrypted house number. If the encrypted house number could not have been obtained by King Arthur's encryption scheme, display `-1` instead.

Sample Input 1	Sample Output 1
3691	46

Sample Input 2	Sample Output 2
2	-1

Sample Input 3	Sample Output 3
7562536	75661

Sample Input 4	Sample Output 4
1031887129	-1

This page is intentionally left (almost) blank.

Problem I
Injecting DNA
Time limit: 5 seconds

Source: Pixabay

Last year, the Mountain Pine Beetles (*Dendroctonus ponderosae*) devastated the local forests. While studying the beetles, Debbie has made a big scientific discovery!

Human DNA is made up of four primary nucleobases: adenine (A), cytosine (C), guanine (G) and thymine (T). Beetle DNA is made up of twenty-six primary nucleobases, represented by the uppercase letters of the alphabet. A strand of DNA is a sequence of nucleobases, and can be represented as a string of corresponding characters. The amazing discovery that Debbie made is that certain strands of beetle DNA can be inserted into human DNA in such a way that the human becomes immune to the common cold!

Unfortunately, there is a drawback. The beetle DNA is somewhat toxic to humans. After months of studying, Debbie has concluded that the toxicity of a strand of DNA is related to the suffixes of the strand (note that we are only interested in nonempty suffixes). You must first take all suffixes of the strand and sort them in lexicographical order. Two suffixes are *out of order* if the longer suffix appears before the shorter suffix in this sorted order. The toxicity of any strand is 1 unit more than the number of (unordered) pairs of suffixes that are out of order. For example, consider this strand of beetle DNA: `BUBBLE`. Here are the suffixes of the strand in sorted order:

```
BBLE   BLE   BUBBLE   E   LE   UBBLE
```

There are eight pairs of suffixes that are out of order, so the toxicity of `BUBBLE` is 9 $(8 + 1)$. The out of order suffix pairs are (`BLE`, `E`), (`BLE`, `LE`), (`BBLE`, `E`), (`BBLE`, `LE`), (`BBLE`, `BLE`), (`BUBBLE`, `E`), (`BUBBLE`, `LE`) and (`BUBBLE`, `UBBLE`).

There is a trade-off because the longer the strand is, the better it fights the common cold. Debbie has come up with a formula for the *effectiveness* of a strand of DNA. Let S be a DNA strand of length n.

$$\text{effectiveness}(S) = \begin{cases} \dfrac{n(n-1)}{\text{toxicity}(S)} & \text{if toxicity}(S) \le 10^8, \\ 0 & \text{otherwise.} \end{cases}$$

Debbie has extracted a piece of beetle DNA and has determined that any suffix of this strand is capable of fighting the common cold. Help Debbie decide which suffix of the strand has the highest effectiveness. If multiple suffixes have the same effectiveness, she will choose the longer one.

Input

The input consists of a single line containing one string, which is the strand of DNA that Debbie has chosen. The string contains only uppercase letters and has length at least 1 and at most 200 000.

Output

Display the length of the strand of DNA that Debbie will choose.

Sample Input 1	Sample Output 1
`ALGORITHM`	`6`

Sample Input 2	Sample Output 2
`SUFFIXARRAY`	`11`

Sample Input 3	Sample Output 3
`SOUTHPACIFICREGIONALFINALS`	`26`

This page is intentionally left (almost) blank.

Problem J
Judge's Mistake
Time limit: 4 seconds

Setting problems for programming competitions takes a long time. Several months ago, the judges sent out a call for problems and Kevin replied with the problem in Figure J.1.

There are C cities in Australia, connected by a network of R two-way roads. There may be several roads connecting the same pair of cities and there may be roads that connect a city to itself. It is possible to reach any city from any other city via a sequence of connected roads. Each road has a maintenance cost in dollars. Due to budget cuts, the government is planning to close a set of roads. The maintenance cost of the remaining roads must be as small as possible and it must still be possible to travel between any pair of cities via some sequence of roads.

What is the cost of the cheapest possible plan?

Source: Pexels

Input

The input starts with a line containing two integers C ($2 \leq C \leq 100\,000$), which is the number of cities, and R ($1 \leq R \leq 100\,000$), which is the number of roads. The next R lines describe the roads. Each of these lines contains three integers u ($1 \leq u \leq C$), v ($1 \leq v \leq C$) and w ($1 \leq w \leq 100\,000$), which indicates that there is a road between u and v with maintenance cost w.

Output

Display the cost of the cheapest maintenance plan.

Sample Input 1	Sample Output 1

3 2
1 3 1
2 3 2
	3

Sample Input 2	Sample Output 2

3 3
1 2 3
2 3 2
3 1 1
	3

Sample Input 3	Sample Output 3

2 2
1 2 1
2 1 2
	1

Figure J.1: The original problem that Kevin proposed.

This problem was going to be used at the Divisionals contest until we (the judges) messed up. While writing our solutions, one of us accidentally sorted the integers in the official input. We were able to determine the values of C and R, but we are unable to determine the order of the remaining $3R$ integers.

Rather than attempting to reconstruct the original data, we will ask a slightly different question. These $3R$ integers could correspond to many different road networks. Out of all possible road networks that the data could represent, what is the smallest output for the original problem? (That is, what is the cheapest maintenance plan over all possible road networks that may be represented?)

Input

The input starts with a line containing two integers C ($2 \leq C \leq 100\,000$), which is the number of cities, and R ($1 \leq R \leq 100\,000$), which is the number of roads.

The second line contains $3R$ integers, which are the u, v and w values from the original problem in Figure J.1. Each of these values is between 1 and 100 000 inclusive. These integers are in nondecreasing order.

It is guaranteed that the input corresponds to at least one possible road network that satisfies the constraints from the original problem.

Output

Display the cost of the cheapest maintenance plan over all possible road networks that may be represented.

Sample Input 1

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

Sample Output 1

```
3
```

Sample Input 2

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

Sample Output 2

```
2
```

Sample Input 3

```
2 2
1 1 1 2 2 2
```

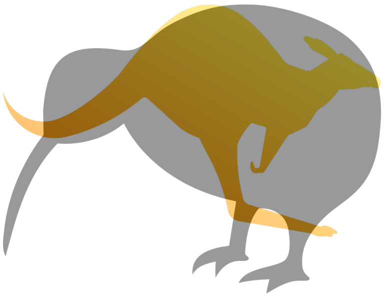
Sample Output 3

```
1
```

Problem K
Kiwis vs Kangaroos II
Time limit: 3 seconds

Last year's feud between the Kiwis and the Kangaroos is still ongoing. Somehow, the word game did not seem to squash the rivalry that has formed between them! Again, they have turned to you to settle this.

You have decided that you will hold a head-to-head programming tournament. Each country, Australia and New Zealand, will send some number of programmers (not necessarily the same number) to compete in the tournament. The programmers from Australia are called 'kangaroos', and the programmers from New Zealand are called 'kiwis'. You have set up n stadiums to hold the tournament. The tournament will take place in n separate rounds.



Source: Pixabay

In each round, n different kangaroos will battle against n different kiwis, with one kangaroo battling one kiwi in each stadium (n battles per round, so n^2 battles in total). To keep things interesting for the spectators, no programmer may battle in any given stadium more than once, though they may battle against the same opponent multiple times in different rounds.

The king of the kangaroos has nominated m kangaroos. The ith kangaroo must fight in exactly t_i different battles. Similarly, the queen of the kiwis has nominated k kiwis. The ith kiwi must fight in exactly s_i battles.

Find a valid tournament schedule that satisfies the above constraints.

Input

The first line of input contains three integers n ($1 \leq n \leq 200$), which is the number of stadiums and rounds, m ($n \leq m \leq n^2$), which is the number of kangaroos, and k ($n \leq k \leq n^2$), which is the number of kiwis.

The second line contains m integers $t_1, \ldots, t_m$ ($1 \leq t_i \leq n$), which are the number of battles each of the kangaroos should compete in. The third line contains k integers $s_1, \ldots, s_k$ ($1 \leq s_i \leq n$), which are the number of battles each of the kiwis should compete in. It is guaranteed that $t_1 + \cdots + t_m = n^2$ and $s_1 + \cdots + s_k = n^2$.

Output

Display a valid schedule.

The schedule should be displayed over n lines. The ith line is the schedule for round i. Each line must contain n battles. Each battle must be of the form avb, where a is the kangaroo in the battle and b is the kiwi in the battle (v is just the character 'v'). The kangaroos are numbered $1, \ldots, m$ and the kiwis are numbered $1, \ldots, k$. The first battle listed on each line is the battle in stadium 1, the second battle listed on each line is the battle in stadium 2, and so on. See the sample output for clarity.

If there are multiple solutions, any one will be considered correct. It is guaranteed that at least one valid schedule exists.

Sample Input 1

```
2 2 2
2 2
2 2
```

Sample Output 1

```
1v2 2v1
2v1 1v2
```

Sample Input 2

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

Sample Output 2

```
1v1 2v2
3v2 4v3
```

Sample Input 3

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

Sample Output 3

```
3v2 4v1 5v6 1v3
1v3 3v4 2v1 5v2
4v1 5v2 1v4 2v5
2v4 1v3 3v5 4v1
```

Problem L
Last Casino
Time limit: 2 seconds

Source: Pexels

The movie *The Last Casino* is about a professor who exploits probabilities to beat the casino and win often at blackjack. This has terrified Lucy, the owner of the new casino in town, *The Little Epsilon* (TLE), so she has banned blackjack in her casino.

To bring people into TLE, she has invented a new game called bins-and-balls, which is a game played by two players, say Alice and Britney. There are n bins; each of which is either red, black or white. There are B balls. Each ball will be independently and randomly placed into one of the n bins with equal probability of landing in each of the bins. If all of the balls end up in red bins, then Alice wins. If there is at least one ball in a white bin and at least one ball in a black bin, then Britney wins. In every other situation, the casino wins.

Because the game is so new, the government has put some restrictions on the game:

1. The casino can choose the number of bins.
2. The casino must determine how many balls to use (some integer between 1 and 10^6 inclusive) and how to paint the bins so that the absolute difference between the probability of Alice winning and the probability of Britney winning is minimised.
3. If there are multiple choices for restriction 2, the casino must choose one that minimises the casino's probability of winning.

Restriction 2 is to ensure that the game is fair for the players, while restriction 3 is so that the casino does not have too large an advantage. Lucy has already decided on the number of bins to use, but she needs your help with the government's restrictions. Determine what colour to paint each of the bins and how many balls to use so that the government's restrictions are satisfied.

Input

The input consists of a single line with one integer n ($1 \leq n \leq 1\,000\,000$), which is the number of bins the casino will use.

Output

Display the number of balls to use. Then display two values: the number of bins to colour black and the number of bins to colour white. The remaining bins will be coloured red. If there are multiple solutions, any one will be accepted.

Sample Input 1	Sample Output 1
5	2 1 2

This page is intentionally left (almost) blank.