Buggy Keyboard

Gabrielle has an assignment for her Algorithmic Problem Solving course due in a couple of hours. She opened up her computer which she purchased recently and started working using her favorite editor when she discovered something strange. Generally, pressing the backspace key should erase a character to the left of the cursor. But on her new computer, pressing that key produced a character `<`. Since the assignment is due in couple of hours, she does not have time to call customer support or replace the computer so she decides to temporarily find a way around the problem with the help of a program.

Help Gabrielle write a program which takes the string written on her computer and outputs the string that Gabrielle actually intended to write. It can be assumed that she never needs to output the character `<`. Also you can be assured that she will never press backspace on an empty line.

Figure 1: Computer keyboard

Input

A string `s` containing text written on Gabrielle's computer. The length of `s` is less than 10^6 The string will contain only lower case letters, spaces, and the character `<`.

Output

A string containing text that was actually intended. Note that there should be no spaces or new line after the last character in the output

Example 1

```
Input:
a<bcd<

Output:
bc
```

Example 2

```
Input:
prind<tf

Output:
printf
```

Example 3

```
Input:
as<d<<

Output:
```

Note: there is no output in the last example.

Daily Prize

Trader Jane's has a new promotion to attract even more customers to already crowded supermarket.

To participate in a daily drawing of the prize a customer needs to fill out a form with their name, phone number and the amount that they paid for their groceries.

At the end of the day, Trader Jane's manager picks two forms from among the completed ones: - first is the one that has the highest amount paid - second is the one that has the lowest amount paid The person who paid the highest amount gets the prize equal to the difference between their bill and the lowest bill.

Given how busy Trader Jane's supermarket is, you can be certain that at the end of each day there are at least two bills in the pool to select from (usually there are many many more).

The selected forms are discarded, but the other ones remain in the pool for the next day, so each customer has a chance to be selected as the prize winner on the day of their purchase, or any day after.

Your task is to compute how much money Trader Jane's pays out in prizes.

Input

The input contains an integer `n`, `1 <= n <= 5,000` on the first line - this is the number of days in the promotion. Each of the next `n` lines contains a sequence of non-negative integers separated by whitespace. The first number on each line indicates the number of forms submitted on that day, `0 <= k <= 100,000`. The next `k` numbers specify the bill amounts on the new forms entered for the daily drawing on that day. Each amount is guaranteed to be no larger than `1,000,000`. The total amount of all bills is no larger than `1,000,000`.

Output

Print one number that is the sum of all the prizes that Trader Jane's pays out during the promotion followed by a newline.

Example 1

```
Input:
5
3 1 2 3
2 1 1
4 10 5 5 1
0
1 2

Output:
19
```

Example 2

```
Input:
2
2 1 2
2 1 2

Output:
2
```

Efficient Adding

You are tasked with writing a program that adds a sequence of numbers. But the added challenge is to do so efficiently!

The cost of adding two numbers `a` and `b` is equal to their sum `a+b`. For example: to add 1, 2, and 3, you can do it as follows:

1 + 2 = 3, cost of 3 3 + 3 = 6, cost of 6 Total cost = 9

or

2 + 3 = 5, cost of 5 1 + 5 = 6, cost of 6 Total cost = 11

or

1 + 3 = 4, cost of 4 2 + 4 = 6, cost of 6 Total cost = 10

Your goal is to add the numbers so that the cost is as small as possible.

Input

The first line of input contains a positive number `N` (`2 <= N <= 5000`) that tells you how many numbers there are to add.

The second line of input contains those `N` numbers `0 <= n_1, n_2, ..., n_N <= 100,000`.

Output

The minimum total cost of addition followed by a newline.

Example 1

```
Input:
3
1 2 3

Output:
9
```

Example 2

```
Input:
4
1 2 3 4

Output:
19
```

Ferry

Ferries used to carry cars across the river. In your village, there is still a ferry that can take up to `N` cars and needs `T` minutes to cross the river. A car may arrive at either river bank and wait to be carried to the opposite bank. The ferry operates continuously between the banks as long it is carrying at least one car or there is at least one car waiting on either side. Whenever the ferry arrives at one bank, it unloads cars carried and loads up to `N` cars waiting at that bank. When there are more than `N` cars waiting, they are loaded on the first-come-first-serve basis. If there is no car waiting on either bank, the ferry stops and waits until one car arrives. The ferry is initially on the left bank. You are asked to determine at what time each car arrives at the other bank.

Figure 1: Car Ferry

Input The first line of input contains three integers `N`, `T` and `M` (`1 <= N, T, M <= 10,000`). Each of the following `M` lines gives the arrival time of a car and the bank at which the car arrives (`left` or `right`). The cars are ordered by their arrival times (so the arrival times are non-decreasing) and the time spent on loading and unloading can be ignored.

Output For each car, you should print one line containing one number, the time at which the car is unloaded at the opposite bank.

Example 1

```
Input:
2 10 10
0 left
10 left
20 left
30 left
40 left
50 left
60 left
70 left
80 left
90 left

Output:
10
30
30
50
```

```
50
70
70
90
90
110
```

Example 2

```
Input:
2 10 3
10 right
25 left
40 left

Output:
30
40
60
```

Find Sums

You are given a multiset of N integers (multiset means that the repeated values are allowed) and a target value S. Your task is to find all distinct subsets of the given multiset for which the sum of all the elements in the subset is equal to S.

Input

The first line of the input contains two integers S (1 <= S <= 1,000) and N (1 <= N <= 12), indicating the target sum and the number of values in the multiset, respectively.

The second line contains N integers, all of which are between 1 and 100 - these are the elements of the multiset.

Output

First, print a line `Sums of S:` where S is the value given in the input. Then print one line for every subset satisfying the condition or a line containing `NONE` if there is no such subset.

For every subset, numbers are printed in decreasing order and separated by a plus sign (`+`). The subsets themselves are sorted lexicographically in decreasing order, i.e. they are sorted by their first integer, then the second integer in case of tie, and so on. Additionally, the subsets you print should not contain repetitions (i.e., you should never print two lines that are identical).

Example 1

```
Input:
4 6
4 3 2 2 1 1

Output:
Sums of 4:
4
3+1
2+2
2+1+1
```

Example 2

```
Input:
5 3
2 1 1

Output:
Sums of 5:
NONE
```

Example 3

```
Input:
400 10
150 100 60 60 50 50 50 30 20 20

Output:
Sums of 400:
150+100+60+60+30
150+100+60+50+20+20
150+100+50+50+50
150+100+50+50+30+20
150+60+60+50+50+30
150+60+50+50+50+20+20
100+60+60+50+50+50+30
```

How Many Words

In an introductory English class, Professor Umbridge assigns the students an impossible task of finding all the unique words in a ten-volume encyclopedia of magic. By her definition words are sequences of one or more consecutive alphabetic characters in upper or lower case. Their uniqueness should be determined in a case insensitive way, so words like “magic”, “Magic”, “MaGiC” should be treated as the same.

As an upperclassman in the school, you and your friends decide to help, by writing a program that will complete the task in a tiny fraction of the time and annoy Professor Umbridge.

Input

The input is a text with up to 5,000 lines. Each line has at most 200 characters. The input is terminated by EOF.

Output

A list of unique words that appear in the text, one per line. The output should be in alphabetical order and in lower case. You are guaranteed that the number of unique words in the text is no more than 5,000.

Example 1

```
Input:
Professor Umbridge, Room 3789, +44-7911 123456

Output:
professor
room
umbridge
```

Example 2

```
Input:
In an introductory English class taught by Professor Umbridge assigns the students
and impossible task for counting all the unique words in a ten-volume encyclopedia
of magic.
By her definition words are sequences of one or more consecutive alphabetic characters
in upper or lower case. Their uniqueness should be determined in a case insensitive way,
so  words like "magic", "Magic", "MaGiC" should be treated as the same.

Output:
a
all
alphabetic
an
and
are
as
assigns
be
by
case
characters
class
consecutive
counting
definition
determined
encyclopedia
english
for
```

her
impossible
in
insensitive
introductory
like
lower
magic
more
of
one
or
professor
same
sequences
should
so
students
task
taught
ten
the
their
treated
umbridge
unique
uniqueness
upper
volume
way
words

Lucky Draw

Each holiday season, the town of Dingle organizes a game for its citizens. People pick their lucky numbers out of a a lucky gold pot supervised by the town's judge. All numbers are in the range of 1 to 100, and the same number may occur on multiple tickets. After each person gets their number, they are required by law to keep it a secret.

When the time comes for the game, the mayor of Dingle arranges all citizens in a single line. The mayor will then eliminate the players two at a time until there are as few left as possible, since these folks will get the prize of 100 gold pieces from the mayor himself.

The mayor is allowed to remove two adjacent citizens from the line if the sum of their numbers is even. When that pair is removed, no other changes are made to the line. The mayor keeps removing pairs of disappointed citizens until there are no more adjacent people with numbers that add up to an even value.

The people who are left, if any, are the winners and end up going home with a pot of 100 gold pieces.

As an office clerk working for the mayor, your job is to figure out the smallest number of pots of gold he will have to hand out at the end of the day.

Input

The first line contains an integer `1 <= n <= 100000`, giving the number of people who participate in the game. The second line contains `n` integers indicating the ticket numbers of people standing in the line from first to last. Each lottery ticket value is in the range of 1 to 100.

Output

The number of pots of gold that the mayor will need to hand out at the end of the day assuming he eliminates as many people as possible.

Note: there should be no newline after the output.

Example 1

```
Input
10
10 11 12 13 14 15 16 17 18 19

Output
10
```

Because no adjacent pair of numbers adds up to an even value.

Example 2

```
Input
9
7 1 3 6 2 1 2 4 8

Output
1
```

Because (7,1), (6,2), (2,4), can be removed, leaving 3, 1, 8, and then (3,1) can be still removed, leaving 8 as the sole winner.

Note: there are alternative solutions that will also lead to 1 winner.