A. Vitaliy and Pie

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After a hard day Vitaly got very hungry and he wants to eat his favorite potato pie. But it's not that simple. Vitaly is in the first room of the house with n room located in a line and numbered starting from one from left to right. You can go from the first room to the second room, from the second room to the third room and so on — you can go from the $(n - 1)$-th room to the n-th room. Thus, you can go to room x only from room $x - 1$.

The potato pie is located in the n-th room and Vitaly needs to go there.

Each pair of consecutive rooms has a door between them. In order to go to room x from room $x - 1$, you need to open the door between the rooms with the corresponding key.

In total the house has several types of doors (represented by uppercase Latin letters) and several types of keys (represented by lowercase Latin letters). The key of type t can open the door of type T if and only if t and T are the same letter, written in different cases. For example, key `f` can open door `F`.

Each of the first $n - 1$ rooms contains exactly one key of some type that Vitaly can use to get to next rooms. Once the door is open with some key, Vitaly won't get the key from the keyhole but he will immediately run into the next room. In other words, each key can open no more than one door.

Vitaly realizes that he may end up in some room without the key that opens the door to the next room. Before the start his run for the potato pie Vitaly can buy any number of keys of any type that is guaranteed to get to room n.

Given the plan of the house, Vitaly wants to know what is the minimum number of keys he needs to buy to surely get to the room n, which has a delicious potato pie. Write a program that will help Vitaly find out this number.

Input

The first line of the input contains a positive integer n $(2 \le n \le 10^5)$ — the number of rooms in the house.

The second line of the input contains string s of length $2 \cdot n - 2$. Let's number the elements of the string from left to right, starting from one.

The odd positions in the given string s contain lowercase Latin letters — the types of the keys that lie in the corresponding rooms. Thus, each odd position i of the given string s contains a lowercase Latin letter — the type of the key that lies in room number $(i + 1) / 2$.

The even positions in the given string contain uppercase Latin letters — the types of doors between the rooms. Thus, each even position i of the given string s contains an uppercase letter — the type of the door that leads from room $i / 2$ to room $i / 2 + 1$.

Output

Print the only integer — the minimum number of keys that Vitaly needs to buy to surely get from room one to room n.

Sample test(s)

input

```
3
aAbB
```

output

```
0
```

input

```
4
aBaCaB
```

output

```
3
```

input

```
5
xYyXzZaZ
```

output

```
2
```

B. Pasha and String

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Pasha got a very beautiful string s for his birthday, the string consists of lowercase Latin letters. The letters in the string are numbered from 1 to $|s|$ from left to right, where $|s|$ is the length of the given string.

Pasha didn't like his present very much so he decided to change it. After his birthday Pasha spent m days performing the following transformations on his string — each day he chose integer a_i and **reversed** a piece of string (a segment) from position a_i to position $|s| - a_i + 1$. It is guaranteed that $2 \cdot a_i \le |s|$.

You face the following task: determine what Pasha's string will look like after m days.

Input

The first line of the input contains Pasha's string s of length from 2 to $2 \cdot 10^5$ characters, consisting of lowercase Latin letters.

The second line contains a single integer m $(1 \le m \le 10^5)$ — the number of days when Pasha changed his string.

The third line contains m space-separated elements a_i $(1 \le a_i; 2 \cdot a_i \le |s|)$ — the position from which Pasha started transforming the string on the i-th day.

Output

In the first line of the output print what Pasha's string s will look like after m days.

Sample test(s)

input

```
abcdef
1
2
```

output

```
aedcbf
```

input

```
vwxyz
2
2 2
```

output

```
vwxyz
```

input

```
abcdef
3
1 2 3
```

output

```
fbdcea
```

C. Ilya and Sticks

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In the evening, after the contest Ilya was bored, and he really felt like maximizing. He remembered that he had a set of n sticks and an instrument. Each stick is characterized by its length l_i.

Ilya decided to make a rectangle from the sticks. And due to his whim, he decided to make rectangles in such a way that maximizes their total area. Each stick is used in making at most one rectangle, it is possible that some of sticks remain unused. Bending sticks is not allowed.

Sticks with lengths a_1, a_2, a_3 and a_4 can make a rectangle if the following properties are observed:

- $a_1 \le a_2 \le a_3 \le a_4$
- $a_1 = a_2$
- $a_3 = a_4$

A rectangle can be made of sticks with lengths of, for example, $3\ 3\ 3\ 3$ or $2\ 2\ 4\ 4$. A rectangle cannot be made of, for example, sticks $5\ 5\ 5\ 7$.

Ilya also has an instrument which can reduce the length of the sticks. The sticks are made of a special material, so the length of each stick can be reduced by at most one. For example, a stick with length 5 can either stay at this length or be transformed into a stick of length 4.

You have to answer the question — what maximum total area of the rectangles can Ilya get with a file if makes rectangles from the available sticks?

Input

The first line of the input contains a positive integer n $(1 \le n \le 10^5)$ — the number of the available sticks.

The second line of the input contains n positive integers l_i $(2 \le l_i \le 10^6)$ — the lengths of the sticks.

Output

The first line of the output must contain a single non-negative integer — the maximum total area of the rectangles that Ilya can make from the available sticks.

Sample test(s)

input

```
4
2 4 4 2
```

output

```
8
```

input

```
4
2 2 3 5
```

output

```
0
```

input

```
4
100003 100004 100005 100006
```

output

```
10000800015
```

D. Arthur and Walls

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Finally it is a day when Arthur has enough money for buying an apartment. He found a great option close to the center of the city with a nice price.

Plan of the apartment found by Arthur looks like a rectangle $n \times m$ consisting of squares of size 1×1. Each of those squares contains either a wall (such square is denoted by a symbol "`*`" on the plan) or a free space (such square is denoted on the plan by a symbol "`.`").

Room in an apartment is a maximal connected area consisting of free squares. Squares are considered adjacent if they share a common side.

The old Arthur dream is to live in an apartment where all rooms are rectangles. He asks you to calculate minimum number of walls you need to remove in order to achieve this goal. After removing a wall from a square it becomes a free square. While removing the walls it is possible that some rooms unite into a single one.

Input

The first line of the input contains two integers n, m $(1 \le n, m \le 2000)$ denoting the size of the Arthur apartments.

Following n lines each contain m symbols — the plan of the apartment.

If the cell is denoted by a symbol "`*`" then it contains a wall.

If the cell is denoted by a symbol "`.`" then it this cell is free from walls and also this cell is contained in some of the rooms.

Output

Output n rows each consisting of m symbols that show how the Arthur apartment plan should look like after deleting the minimum number of walls in order to make each room (maximum connected area free from walls) be a rectangle.

If there are several possible answers, output any of them.

Sample test(s)

input

```
5 5
.*.*.
*****
.*.*.
*****
.*.*.
```

output

```
.*.*.
*****
.*.*.
*****
.*.*.
```

input

```
6 7
***.*.*
..*.*.*
*.*.*.*
*.*.*.*
..*...*
*******
```

output

```
***...*
..*...*
..*...*
..*...*
..*...*
*******
```

input

```
4 5
.....
.....
..***
..*..
```

output

```
.....
.....
.....
.....
```

E. Anya and Cubes

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Anya loves to fold and stick. Today she decided to do just that.

Anya has n cubes lying in a line and numbered from 1 to n from left to right, with natural numbers written on them. She also has k stickers with exclamation marks. We know that the number of stickers does not exceed the number of cubes.

Anya can stick an exclamation mark on the cube and get the factorial of the number written on the cube. For example, if a cube reads 5, then after the sticking it reads $5!$, which equals 120.

You need to help Anya count how many ways there are to choose some of the cubes and stick on some of the chosen cubes at most k exclamation marks so that the sum of the numbers written on the chosen cubes after the sticking becomes equal to S. Anya can stick at most one exclamation mark on each cube. Can you do it?

Two ways are considered the same if they have the same set of chosen cubes and the same set of cubes with exclamation marks.

Input

The first line of the input contains three space-separated integers n, k and S ($1 \le n \le 25$, $0 \le k \le n$, $1 \le S \le 10^{16}$) — the number of cubes and the number of stickers that Anya has, and the sum that she needs to get.

The second line contains n positive integers a_i ($1 \le a_i \le 10^9$) — the numbers, written on the cubes. The cubes in the input are described in the order from left to right, starting from the first one.

Multiple cubes can contain the same numbers.

Output

Output the number of ways to choose some number of cubes and stick exclamation marks on some of them so that the sum of the numbers became equal to the given number S.

Sample test(s)

input

```
2 2 30
4 3
```

output

```
1
```

input

```
2 2 7
4 3
```

output

```
1
```

input

```
3 1 1
1 1 1
```

output

```
6
```

Note

In the first sample the only way is to choose both cubes and stick an exclamation mark on each of them.

In the second sample the only way is to choose both cubes but don't stick an exclamation mark on any of them.

In the third sample it is possible to choose any of the cubes in three ways, and also we may choose to stick or not to stick the exclamation mark on it. So, the total number of ways is six.