TASK	KRIŽALJKA	EKO	DNA	RAZBIBRIGA	BLOKOVI	POPLOČAVANJE
source code	krizaljka.pas krizaljka.c krizaljka.cpp	eko.pas eko.c eko.cpp	dna.pas dna.c dna.cpp	razbibriga.pas razbibriga.c razbibriga.cpp	blokovi.pas blokovi.c blokovi.cpp	poplocavanje.pas poplocavanje.c poplocavanje.cpp
input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	1 second	1 second	1 second	1 second	1 second	4 seconds
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	512 MB
point value	**50**	**80**	**100**	**120**	**140**	**160**
	650					

Since ACTA has entered into force, Slavko has been spending his time offline, solving crosswords. Having solved almost all that he could get his hands on, he wants to make a few crosswords of his own. However, he is too sloppy for such fine work, so he has asked you to help him generate the crosswords.

You are given two words, **A** and **B**. The word **A** must be output **horizontally**, and the word **B** **vertically**, so that the two words cross (i.e. share exactly one letter). The shared letter must be the first letter in **A** that is also contained in **B**, more specifically the first occurrence of that letter in each word.

For example, given the words **A** = "ABBA" and **B** = "CCBB", you need to output 4 lines as shown below:

```
.C..
.C..
ABBA
.B..
```

INPUT

The first and only line of input contains two words, **A** and **B**, not more than 30 characters long, separated by a single space. Both words will contain only uppercase English letters. There will be at least one letter contained in both words.

OUTPUT

Let **N** be the length of word **A**, and **M** the length of word **B**. The output must contain **M** lines, each containing **N** characters. The character grid must contain the two words crossed as described above. All other characters in the grid must be periods (the character '.', without quotes), thus padding all lines to the length of **N** characters.

SAMPLE TESTS

input	**input**	**input**
BANANA PIDZAMA	MAMA TATA	REPUBLIKA HRVATSKA
output	**output**	**output**
.P.... .I.... .D.... .Z.... BANANA .M.... .A....	.T.. MAMA .T.. .A..	H........ REPUBLIKA V........ A........ T........ S........ K........ A........

Lumberjack Mirko needs to chop down **M** metres of wood. It is an easy job for him since he has a nifty new woodcutting machine that can take down forests like wildfire. However, Mirko is only allowed to cut a single row of trees.

Mirko's machine works as follows: Mirko sets a height parameter **H** (in metres), and the machine raises a giant sawblade to that height and cuts off all tree parts higher than **H** (of course, trees not higher than **H** meters remain intact). Mirko then takes the parts that were cut off. For example, if the tree row contains trees with heights of 20, 15, 10, and 17 metres, and Mirko raises his sawblade to 15 metres, the remaining tree heights after cutting will be 15, 15, 10, and 15 metres, respectively, while Mirko will take 5 metres off the first tree and 2 metres off the fourth tree (7 metres of wood in total).

Mirko is **ecologically** minded, so he doesn't want to cut off more wood than necessary. That's why he wants to set his sawblade as high as possible. Help Mirko find the **maximum integer height** of the sawblade that still allows him to cut off **at least M** metres of wood.

INPUT

The first line of input contains two space-separated positive integers, **N** (the number of trees, $1 \leq \mathbf{N} \leq 1\,000\,000$) and **M** (Mirko's required wood amount, $1 \leq \mathbf{M} \leq 2\,000\,000\,000$).

The second line of input contains **N** space-separated positive integers less than 1 000 000 000, the heights of each tree (in metres). The sum of all heights will exceed **M**, thus Mirko will always be able to obtain the required amount of wood.

OUTPUT

The first and only line of output must contain the required height setting.

SAMPLE TESTS

input	**input**
4 7 20 15 10 17	5 20 4 42 40 26 46
output	**output**
15	36

Biologists have discovered a strange DNA molecule, best described as a sequence of **N** characters from the set {A, B}. An unlikely sequence of mutations has resulted in a DNA strand consisting only of A's. Biologists found that very odd, so they began studying the mutations in greater detail.

They discovered two types of mutations. One type results in changing a single character of the sequence (A → B or B → A). The second type changes a whole **prefix** of the sequence, specifically replacing all characters in positions from 1 to **K** (for some **K** between 1 and **N**, inclusive) with the other character (A with B, B with A).

Compute the least possible number of mutations that could convert the starting molecule to its end state (containing only A characters). Mutations can occur in any order.

INPUT

The first line of input contains the positive integer **N** ($1 \leq \mathbf{N} \leq 1\,000\,000$), the length of the molecule.

The second line of input contains a string with **N** characters, with each character being either A or B. This string represents the starting state of the molecule.

OUTPUT

The first and only line of output must contain the required minimum number of mutations.

SAMPLE TESTS

input	**input**	**input**
4 ABBA	5 BBABB	12 AAABBBAAABBB
output	**output**	**output**
2	2	4

Mirko has already solved all crosswords available in daily newspapers, and he is too intimidated by ACTA to download new ones from the Internet. Now he has asked Frane to challenge him with a programming problem. Frane has eagerly responded with a brand new task, and was kind enough not to lock the task under copyright protection, so you are allowed to attempt it too!

Four words with equal lengths can be put together in a square such that two words mark horizontal edges, while the other two mark vertical edges. Horizontal words are read left to right, and vertical ones from top to bottom. Corner letters are shared by the two neighbouring edges. The figure below shows one possible square created from words "HLAD", "NIVA", "HSIN", "DEDA".

H	L	A	D
S			E
I			D
N	I	V	A

Your task is, given a list of equal-length words, to compute the number of different squares that can be put together from a subset of these words. You are **not allowed to repeat a word in the same square**. Two squares are different if they differ in at least one character.

INPUT

The first line of input contains the positive integer **N** ($4 \leq \mathbf{N} \leq 100\,000$), the number of words in the list.

Each of the next **N** lines contains a single word consisting only of uppercase English letters.

Each word will contain at most 10 characters. All words will be distinct and have equal length.

OUTPUT

The first and only line of output must contain the required number of distinct squares. **Note:** Test data will ensure that the solution fits in a 64-bit integer data type (int64 in Pascal, long long in C/C++).

SAMPLE TESTS

input

```
4
NIVA
HLAD
HSIN
DEDA
```

output

```
2
```

input

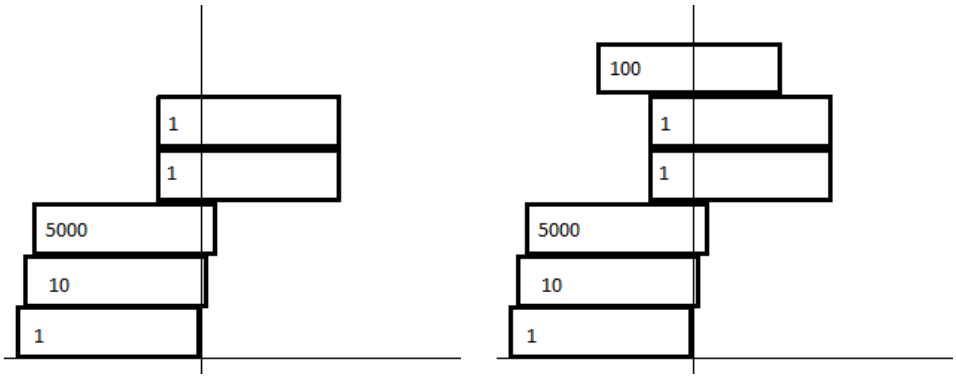
```
6
BAKA
BARA
BALC
CALC
ARHC
BLIC
```

output

```
8
```

N rectangles with given masses ($\mathbf{m}_i$) and equal lengths (**2**) and heights (**h**) are arranged in a Cartesian plane such that:

- rectangle edges are parallel to the coordinate axes;
- the y-coordinates of lower horizontal edges are distinct and assume the following values: 0, **h**, 2**h**, 3**h**, …, (**N** - 1)**h**;
- the lowest rectangle's lower left corner has coordinates (-2, 0), while the lower right corner coincides with the origin.



The **X-centre of a rectangle** is the x-coordinate of the midpoint of its lower edge.

The **X-barycentre** of one or more rectangles is the weighted average of their X-centres. It is computed as

$$Xbarycentre = \frac{\sum_i m_i \cdot Xcentre(i)}{\sum_i m_i}$$

In other words, the mass of each rectangle is multiplied by its X-centre and the sum of these products is then divided by the total mass of the rectangles.

An arrangement is **stable** if, for each rectangle A:

- the X-barycentre of rectangles above A has distance of at most 1 from the X-centre of A (i.e. is contained in the x-interval that covers A).

Intuitively, stability of an arrangement can be understood as the precondition for the arrangement to **not fall apart**. The arrangement in the figure on the left is unstable since the X-barycentre of the top two rectangles falls outside the rectangle underneath (the distance of the X-barycentre to the X-centre of the underlying rectangle is greater than 1). The arrangement in the figure on the right is stable.

Given the masses of all rectangles, find the **largest** ("rightmost") **possible x-coordinate** of **any** rectangle corner in a stable arrangement. You are not allowed to change the order of rectangles (they are given from the lowest to the highest one).

INPUT

The first line of input contains the positive integer **N** ($2 \leq \mathbf{N} \leq 300\,000$), the number of rectangles.

Each of the next **N** lines contains a single positive integer less than 10 000, the mass of a rectangle. The masses are given in order **from the lowest to the highest** rectangle.

OUTPUT

The first and only line of output must contain the required rightmost x-coordinate. The given result must be within 0.000001 of the official solution.

SCORING

In test cases worth 30% of points, the rectangles will be ordered from the heaviest to the lightest one.

SAMPLE TESTS

input	**input**	**input**
2 1 1	3 1 1 1	3 1 1 9
output	**output**	**output**
1.000000	1.500000	1.900000

Mirko's ASCII street is made of **N** lowercase letters of the English alphabet. The city government occasionally replaces the tiles in the street. However, the letter tiles are in high demand, so the government has only **M** different tile patterns available.

The **i**th tile pattern consists of $\mathbf{L_i}$ letters. A tile cannot be rotated or broken into pieces, and it can only be placed such that the **tile letters coincide with the contiguous letter subsequence in the street**. Tiles can overlap and we can use multiple tiles of the same pattern.

A street cell is **untileable** if it cannot be covered by any tile. Compute the number of untileable cells.

INPUT

The first line of input contains the positive integer **N** ($1 \leq \mathbf{N} \leq 300\,000$), the length of the street.

The second line of input contains **N** lowercase English letters, the letter sequence in the street.

The third line of input contains the positive integer **M** ($1 \leq \mathbf{M} \leq 5000$), the number of tile patterns.

Each of the next **M** lines contains a description of a tile pattern with length $\mathbf{L_i}$ ($1 \leq \mathbf{L_i} \leq 5000$). The tile pattern descriptions consist of lowercase English letters.

OUTPUT

The first and only line of output must contain the required number of untileable cells.

SAMPLE TESTS

input	**input**	**input**
6 abcbab 2 cb cbab	4 abab 2 bac baba	6 abcabc 2 abca cab
output	**output**	**output**
2	4	1