A. Pasha and Stick

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Pasha has a wooden stick of some positive integer length n. He wants to perform exactly three cuts to get four parts of the stick. Each part must have some positive integer length and the sum of these lengths will obviously be n.

Pasha likes rectangles but hates squares, so he wonders, how many ways are there to split a stick into four parts so that it's possible to form a rectangle using these parts, but is impossible to form a square.

Your task is to help Pasha and count the number of such ways. Two ways to cut the stick are considered distinct if there exists some integer x, such that the number of parts of length x in the first way differ from the number of parts of length x in the second way.

Input

The first line of the input contains a positive integer n ($1 \le n \le 2 \cdot 10^9$) — the length of Pasha's stick.

Output

The output should contain a single integer — the number of ways to split Pasha's stick into four parts of positive integer length so that it's possible to make a rectangle by connecting the ends of these parts, but is impossible to form a square.

Examples

input

```
6
```

output

```
1
```

input

```
20
```

output

```
4
```

Note

There is only one way to divide the stick in the first sample {1, 1, 2, 2}.

Four ways to divide the stick in the second sample are {1, 1, 9, 9}, {2, 2, 8, 8}, {3, 3, 7, 7} and {4, 4, 6, 6}. Note that {5, 5, 5, 5} doesn't work.

B. Vika and Squares

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

Vika has n jars with paints of distinct colors. All the jars are numbered from 1 to n and the i-th jar contains a_i liters of paint of color i.

Vika also has an infinitely long rectangular piece of paper of width 1, consisting of squares of size 1×1. Squares are numbered 1, 2, 3 and so on. Vika decided that she will start painting squares one by one from left to right, starting from the square number 1 and some arbitrary color. If the square was painted in color x, then the next square will be painted in color $x + 1$. In case of $x = n$, next square is painted in color 1. If there is no more paint of the color Vika wants to use now, then she stops.

Square is always painted in only one color, and it takes exactly 1 liter of paint. Your task is to calculate the maximum number of squares that might be painted, if Vika chooses right color to paint the first square.

Input

The first line of the input contains a single integer n ($1 \le n \le 200\,000$) — the number of jars with colors Vika has.

The second line of the input contains a sequence of integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$), where a_i is equal to the number of liters of paint in the i-th jar, i.e. the number of liters of color i that Vika has.

Output

The only line of the output should contain a single integer — the maximum number of squares that Vika can paint if she follows the rules described above.

Examples

input

```
5
2 4 2 3 3
```

output

```
12
```

input

```
3
5 5 5
```

output

```
15
```

input

```
6
10 10 10 1 10 10
```

output

```
11
```

Note

In the first sample the best strategy is to start painting using color 4. Then the squares will be painted in the following colors (from left to right): 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5.

In the second sample Vika can start to paint using any color.

In the third sample Vika should start painting using color number 5.

C. Harmony Analysis

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The semester is already ending, so Danil made an effort and decided to visit a lesson on harmony analysis to know how does the professor look like, at least. Danil was very bored on this lesson until the teacher gave the group a simple task: find 4 vectors in 4-dimensional space, such that every coordinate of every vector is 1 or -1 and any two vectors are orthogonal. Just as a reminder, two vectors in n-dimensional space are considered to be orthogonal if and only if their scalar product is equal to zero, that is:

$$\sum_{i=1}^{n} a_i \cdot b_i = 0$$

.

Danil quickly managed to come up with the solution for this problem and the teacher noticed that the problem can be solved in a more general case for 2^k vectors in 2^k-dimensinoal space. When Danil came home, he quickly came up with the solution for this problem. Can you cope with it?

Input

The only line of the input contains a single integer k ($0 \le k \le 9$).

Output

Print 2^k lines consisting of 2^k characters each. The j-th character of the i-th line must be equal to ' $*$ ' if the j-th coordinate of the i-th vector is equal to -1, and must be equal to ' $+$ ' if it's equal to $+1$. It's guaranteed that the answer always exists.

If there are many correct answers, print any.

Examples

input

```
2
```

output

```
++**
+*+*
++++
+**+
```

Note

Consider all scalar products in example:

- Vectors 1 and 2: $(+1)\cdot(+1)+(+1)\cdot(-1)+(-1)\cdot(+1)+(-1)\cdot(-1)=0$
- Vectors 1 and 3: $(+1)\cdot(+1)+(+1)\cdot(+1)+(-1)\cdot(+1)+(-1)\cdot(+1)=0$
- Vectors 1 and 4: $(+1)\cdot(+1)+(+1)\cdot(-1)+(-1)\cdot(-1)+(-1)\cdot(+1)=0$
- Vectors 2 and 3: $(+1)\cdot(+1)+(-1)\cdot(+1)+(+1)\cdot(+1)+(-1)\cdot(+1)=0$
- Vectors 2 and 4: $(+1)\cdot(+1)+(-1)\cdot(-1)+(+1)\cdot(-1)+(-1)\cdot(+1)=0$
- Vectors 3 and 4: $(+1)\cdot(+1)+(+1)\cdot(-1)+(+1)\cdot(-1)+(+1)\cdot(+1)=0$

D. Vika and Segments

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

Vika has an infinite sheet of squared paper. Initially all squares are white. She introduced a two-dimensional coordinate system on this sheet and drew n black horizontal and vertical segments parallel to the coordinate axes. All segments have width equal to 1 square, that means every segment occupy some set of neighbouring squares situated in one row or one column.

Your task is to calculate the number of painted cells. If a cell was painted more than once, it should be calculated exactly once.

Input

The first line of the input contains a single integer n ($1 \le n \le 100\,000$) — the number of segments drawn by Vika.

Each of the next n lines contains four integers x_1, y_1, x_2 and y_2 ($-10^9 \le x_1, y_1, x_2, y_2 \le 10^9$) — the coordinates of the endpoints of the segments drawn by Vika. It is guaranteed that all the segments are parallel to coordinate axes. Segments may touch, overlap and even completely coincide.

Output

Print the number of cells painted by Vika. If a cell was painted more than once, it should be calculated exactly once in the answer.

Examples

input

```
3
0 1 2 1
1 4 1 2
0 3 2 3
```

output

```
8
```

input

```
4
-2 -1 2 -1
2 1 -2 1
-1 -2 -1 2
1 2 1 -2
```

output

```
16
```

Note

In the first sample Vika will paint squares $(0, 1)$, $(1, 1)$, $(2, 1)$, $(1, 2)$, $(1, 3)$, $(1, 4)$, $(0, 3)$ and $(2, 3)$.

E. Alphabet Permutations

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given a string s of length n, consisting of first k lowercase English letters.

We define a c*-repeat* of some string q as a string, consisting of c copies of the string q. For example, string "acbacbacbacb" is a 4-repeat of the string "acb".

Let's say that string a contains string b as a subsequence, if string b can be obtained from a by erasing some symbols.

Let p be a string that represents some permutation of the first k lowercase English letters. We define function $d(p)$ as the smallest integer such that a $d(p)$-repeat of the string p contains string s as a subsequence.

There are m operations of one of two types that can be applied to string s:

1. Replace all characters at positions from l_i to r_i by a character c_i.
2. For the given p, that is a permutation of first k lowercase English letters, find the value of function $d(p)$.

All operations are performed sequentially, in the order they appear in the input. Your task is to determine the values of function $d(p)$ for all operations of the second type.

Input

The first line contains three positive integers n, m and k ($1 \le n \le 200\,000, 1 \le m \le 20000, 1 \le k \le 10$) — the length of the string s, the number of operations and the size of the alphabet respectively. The second line contains the string s itself.

Each of the following lines m contains a description of some operation:

1. Operation of the first type starts with 1 followed by a triple l_i, r_i and c_i, that denotes replacement of all characters at positions from l_i to r_i by character c_i ($1 \le l_i \le r_i \le n$, c_i is one of the first k lowercase English letters).
2. Operation of the second type starts with 2 followed by a permutation of the first k lowercase English letters.

Output

For each query of the second type the value of function $d(p)$.

Examples

input

```
7 4 3
abacaba
1 3 5 b
2 abc
1 4 4 c
2 cba
```

output

```
6
5
```

Note

After the first operation the string s will be abbbbba.

In the second operation the answer is 6-repeat of abc: ABcaBcaBcaBcaBcAbc.

After the third operation the string s will be abbcbba.

In the fourth operation the answer is 5-repeat of cba: cbAcBacBaCBacBA.

Uppercase letters means the occurrences of symbols from the string s.