A - Poisonous Oyster

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

There are four types of oysters, labeled 1, 2, 3, and 4. Exactly one of these types causes stomach trouble if eaten. The other types do not cause stomach trouble when eaten.

Takahashi ate oysters 1 and 2, and Aoki ate oysters 1 and 3. The information on whether each person got sick is given as two strings S_1 and S_2. Specifically, $S_1 = \texttt{sick}$ means Takahashi got sick, and $S_1 = \texttt{fine}$ means Takahashi did not get sick. Likewise, $S_2 = \texttt{sick}$ means Aoki got sick, and $S_2 = \texttt{fine}$ means Aoki did not get sick.

Based on the given information, find which type of oyster causes stomach trouble.

Constraints

- Each of S_1 and S_2 is `sick` or `fine`.

Input

The input is given from Standard Input in the following format:

S_1 S_2

Output

Print the label of the oyster that causes stomach trouble if eaten.

Sample Input 1 Copy

```
sick fine
```

Sample Output 1 Copy

```
2
```

Takahashi (who ate oysters 1 and 2) got sick, and Aoki (who ate oysters 1 and 3) did not get sick, so it can be concluded that oyster 2 causes stomach trouble.

Sample Input 2 Copy

```
fine fine
```

Sample Output 2 Copy

```
4
```

Neither Takahashi (who ate oysters 1 and 2) nor Aoki (who ate oysters 1 and 3) got sick, so it can be concluded that oyster 4 causes stomach trouble.

B - A..B..C

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 200 points

Problem Statement

A string S is given.

Find how many places in S have `A`, `B`, and `C` in this order at even intervals.

Specifically, find the number of triples of integers (i, j, k) that satisfy all of the following conditions. Here, $|S|$ denotes the length of S, and S_x denotes the x-th character of S.

- $1 \leq i < j < k \leq |S|$
- $j - i = k - j$
- $S_i =$ `A`
- $S_j =$ `B`
- $S_k =$ `C`

Constraints

- S is an uppercase English string with length between 3 and 100, inclusive.

Input

The input is given from Standard Input in the following format:

S

Output

Print the answer.

Sample Input 1 Copy

AABCC

Sample Output 1 Copy

```
2
```

There are two triples $(i, j, k) = (1, 3, 5)$ and $(2, 3, 4)$ that satisfy the conditions.

Sample Input 2 Copy

```
ARC
```

Sample Output 2 Copy

```
0
```

Sample Input 3 Copy

```
AABAAABBAEDCCCD
```

Sample Output 3 Copy

```
4
```

C - Make it Simple

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 300 points

Problem Statement

You are given an undirected graph with N vertices and M edges, where the vertices are numbered 1 through N and the edges are numbered 1 through M. Edge i connects vertices u_i and v_i.
To make the graph simple by removing edges, what is the minimum number of edges that must be removed?
Here, a graph is called simple if and only if it does not contain self-loops or multi-edges.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq M \leq 5 \times 10^5$
- $1 \leq u_i \leq N$
- $1 \leq v_i \leq N$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{ll}
N & M \\
u_1 & v_1 \\
u_2 & v_2 \\
\vdots & \\
u_M & v_M
\end{array}
$$

Output

Print the minimum number of edges that must be removed to make the graph simple.

Sample Input 1 Copy

```
3 5
1 2
2 3
3 2
3 1
1 1
```

Sample Output 1 Copy

```
2
```

By removing edges 3 and 5, the graph becomes simple. This is one of the ways to remove the minimum number of edges, so the answer is 2.

Sample Input 2 Copy

```
1 0
```

Sample Output 2 Copy

```
0
```

Sample Input 3 Copy

```
6 10
6 2
4 1
5 1
6 6
5 3
5 1
1 4
6 4
4 2
5 6
```

Sample Output 3 Copy

```
3
```

D - Swap to Gather

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 425 points

Problem Statement

You are given a string S of length N consisting of `0` and `1`. It is guaranteed that S contains at least one `1`.

You may perform the following operation any number of times (possibly zero):

- Choose an integer i ($1 \leq i \leq N-1$) and swap the i-th and $(i+1)$-th characters of S.

Find the minimum number of operations needed so that all `1`s are contiguous.

Here, all `1`s are said to be contiguous if and only if there exist integers l and r ($1 \leq l \leq r \leq N$) such that the i-th character of S is `1` if and only if $l \leq i \leq r$, and `0` otherwise.

Constraints

- $2 \leq N \leq 5 \times 10^5$
- N is an integer.
- S is a length N string of `0` and `1`.
- S contains at least one `1`.

Input

The input is given from Standard Input in the following format:

N
S

Output

Print the answer.

Sample Input 1

Copy

```
7
0101001
```

Sample Output 1 Copy

```
3
```

For example, the following three operations make all `1` s contiguous:

- Choose $i = 2$ and swap the 2nd and 3rd characters. Then, $S =$ `0011001` .
- Choose $i = 6$ and swap the 6th and 7th characters. Then, $S =$ `0011010` .
- Choose $i = 5$ and swap the 5th and 6th characters. Then, $S =$ `0011100` .

It is impossible to do this in two or fewer swaps, so the answer is 3.

Sample Input 2 Copy

```
3
100
```

Sample Output 2 Copy

```
0
```

All `1` s are already contiguous, so no swaps are needed.

Sample Input 3 Copy

```
10
0101001001
```

Sample Output 3 Copy

```
7
```

E - GCD of Subset

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 475 points

Problem Statement

You are given a sequence $A = (A_1, A_2, \ldots, A_N)$ of length N and a positive integer K (at most N).
For each $i = 1, 2, \ldots, N$, solve the following problem:

- When you choose K elements from A that include A_i, find the maximum possible GCD (greatest common divisor) of those chosen elements.

Constraints

- $1 \leq K \leq N \leq 1.2 \times 10^6$
- $1 \leq A_i \leq 10^6$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{llll}
N & K & & \\
A_1 & A_2 & \ldots & A_N
\end{array}
$$

Output

Print N lines. The j-th line should contain the answer for $i = j$.

Sample Input 1 Copy

```
5 2
3 4 6 7 12
```

Sample Output 1 Copy

```
3
4
6
1
6
```

For $i = 1$, choosing A_1 and A_3 yields $\gcd(\{3, 6\}) = 3$, which is the maximum.
For $i = 2$, choosing A_2 and A_5 yields $\gcd(\{4, 12\}) = 4$, which is the maximum.
For $i = 3$, choosing A_3 and A_5 yields $\gcd(\{6, 12\}) = 6$, which is the maximum.
For $i = 4$, choosing A_4 and A_2 yields $\gcd(\{7, 4\}) = 1$, which is the maximum.
For $i = 5$, choosing A_5 and A_3 yields $\gcd(\{12, 6\}) = 6$, which is the maximum.

Sample Input 2 Copy

```
3 3
6 10 15
```

Sample Output 2 Copy

```
1
1
1
```

Sample Input 3 Copy

```
10 3
414003 854320 485570 52740 833292 625990 909680 885153 435420 221663
```

Sample Output 3 Copy

```
59
590
590
879
879
590
20
879
590
59
```

F - Prefix LIS Query

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 500 points

Problem Statement

You are given a sequence $A = (A_1, A_2, \ldots, A_N)$ of length N.

Answer Q queries. The i-th query ($1 \leq i \leq Q$) is as follows:

- You are given integers R_i and X_i. Consider a subsequence (not necessarily contiguous) of $(A_1, A_2, \ldots, A_{R_i})$ that is strictly increasing and consists only of elements at most X_i. Find the maximum possible length of such a subsequence. It is guaranteed that $X_i \geq \min\{A_1, A_2, \ldots, A_{R_i}\}$.

Constraints

- $1 \leq N, Q \leq 2 \times 10^5$
- $1 \leq A_i \leq 10^9$
- $1 \leq R_i \leq N$
- $\min\{A_1, A_2, \ldots, A_{R_i}\} \leq X_i \leq 10^9$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{llll}
N & Q & & \\
A_1 & A_2 & \ldots & A_N \\
R_1 & X_1 & & \\
R_2 & X_2 & & \\
\vdots & & & \\
R_Q & X_Q & &
\end{array}
$$

Output

Print Q lines. The i-th line should contain the answer to the i-th query.

Sample Input 1 Copy

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

Sample Output 1 Copy

```
2
1
2
```

- 1st query: For the sequence $(2, 4)$, the longest strictly increasing subsequence with all elements at most 5 has length 2.
  Specifically, $(2, 4)$ qualifies.
- 2nd query: For the sequence $(2, 4, 1, 3, 3)$, the longest strictly increasing subsequence with all elements at most 2 has length 1.
  Specifically, (2) and (1) qualify.
- 3rd query: For the sequence $(2, 4, 1, 3, 3)$, the longest strictly increasing subsequence with all elements at most 3 has length 2.
  Specifically, $(2, 3)$ and $(1, 3)$ qualify.

Sample Input 2 Copy

```
10 8
2 5 6 5 2 1 7 9 7 2
7 8
5 2
2 3
2 6
7 3
8 9
9 6
8 7
```

Sample Output 2

Copy

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

G - Unevenness

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 675 points

Problem Statement

There is a grid with N rows and N columns. Let (i, j) denote the cell at the i-th row and j-th column. (i, j) contains an integer $A_{i,j}$.
You are also given two coprime positive integers P and Q.

You may perform the following operation any number of times (possibly zero), as long as the total cost does not exceed $\frac{P}{Q}$:

- Choose a positive real number x. Choose one cell in the grid and either increase or decrease the value in that cell by x. This operation incurs a cost of x.

After performing all operations, let $B_{i,j}$ be the value in cell (i, j). Define the **non-uniformity** U as the sum of absolute differences of adjacent cells. Formally,

$$U = \sum_{i=1}^{N} \sum_{j=1}^{N-1} |B_{i,j} - B_{i,j+1}| + \sum_{i=1}^{N-1} \sum_{j=1}^{N} |B_{i,j} - B_{i+1,j}|.$$

Find the minimum possible value of U after performing the operations in an optimal way, and print that value. Also, print one valid final configuration $B_{i,j}$ that achieves this minimum U.

Constraints

- $2 \leq N \leq 10$
- $1 \leq P \leq 10^{12}$
- $1 \leq Q \leq 10^{12}$
- $\gcd(P, Q) = 1$
- $0 \leq A_{i,j} \leq 10$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$\begin{array}{llll} N & P & Q & \\ A_{1,1} & A_{1,2} & \dots & A_{1,N} \\ A_{2,1} & A_{2,2} & \dots & A_{2,N} \\ \vdots & & & \\ A_{N,1} & A_{N,2} & \dots & A_{N,N} \end{array}$$

Output

Print U and $B_{i,j}$ in the following format:

$$
\begin{aligned}
&U \\
&B_{1,1} \ B_{1,2} \ \dots \ B_{1,N} \\
&B_{2,1} \ B_{2,2} \ \dots \ B_{2,N} \\
&\vdots \\
&B_{N,1} \ B_{N,2} \ \dots \ B_{N,N}
\end{aligned}
$$

Your output is considered correct if it meets all of the following conditions. (Note that the tolerance for U is very strict.)

- The absolute or relative error between your output U and its true minimum value is at most $2^{-51} (\approx 4.44 \times 10^{-16})$.
- The absolute or relative error between $\sum_{i=1}^{N} \sum_{j=1}^{N-1} |B_{i,j} - B_{i,j+1}| + \sum_{i=1}^{N-1} \sum_{j=1}^{N} |B_{i,j} - B_{i+1,j}|$ and U is at most 10^{-10}.
- $\sum_{i=1}^{N} \sum_{j=1}^{N} |A_{i,j} - B_{i,j}|$ is at most $\frac{P}{Q} + \max\left(1, \frac{P}{Q}\right) \times 10^{-10}$.

If there are multiple solutions, you may print any one of them.

Sample Input 1

Copy

```
3 3 1
3 6 1
2 4 2
5 7 9
```

Sample Output 1

Copy

```
24.000000000000000000
3.000000000000000000 4.000000000000000000 1.000000000000000000
3.000000000000000000 4.000000000000000000 2.000000000000000000
5.000000000000000000 7.000000000000000000 9.000000000000000000
```

By performing the following operations, we obtain $U = 24$, which is the minimum possible value. The total cost is $2 + 1 = 3$.

- Let $x = 2$. Decrease the value in cell $(1, 2)$ by 2.
- Let $x = 1$. Increase the value in cell $(2, 1)$ by 1.

Sample Input 2 Copy

```
5 3 1
1 1 1 1 1
1 3 3 3 1
1 3 2 3 1
1 3 3 3 1
1 1 1 1 1
```

Sample Output 2 Copy

```
20.5714285714285714281
1.000000000000000000 1.000000000000000000 1.000000000000000000 1.000000000000000000 1.000000000000000000
1.000000000000000000 2.714285714285714285 2.714285714285714285 2.714285714285714285 1.000000000000000000
1.000000000000000000 2.714285714285714285 2.714285714285714285 2.714285714285714285 1.000000000000000000
1.000000000000000000 2.714285714285714285 2.714285714285714285 2.714285714285714285 1.000000000000000000
1.000000000000000000 1.000000000000000000 1.000000000000000000 1.000000000000000000 1.000000000000000000
```

Sample Input 3 Copy

```
2 393 1
0 0
0 0
```

Sample Output 3 Copy

```
0.000000000000000000
0.000000000000000000 0.000000000000000000
0.000000000000000000 0.000000000000000000
```

Sample Input 4 Copy

```
5 36 5
4 8 7 5 4
0 6 8 3 5
3 7 1 4 5
4 7 1 5 6
2 0 2 4 6
```

Sample Output 4 Copy

```
71.400000000000000014
4.000000000000000000 8.000000000000000000 7.000000000000000000 5.000000000000000000 4.000000000000000000
2.228571428571428571 6.000000000000000000 7.000000000000000000 4.000000000000000000 5.000000000000000000
3.000000000000000000 7.000000000000000000 1.742857142857142857 4.000000000000000000 5.000000000000000000
4.000000000000000000 7.000000000000000000 1.742857142857142857 5.000000000000000000 6.000000000000000000
2.000000000000000000 1.485714285714285714 2.000000000000000000 4.000000000000000000 6.000000000000000000
```

Sample Input 5 Copy

```
5 160 7
6 3 2 7 9
0 1 5 5 7
7 8 4 7 5
4 0 8 5 6
3 6 1 9 0
```

Sample Output 5 Copy

```
65.428571428571428571
6.000000000000000000 3.000000000000000000 3.000000000000000000 7.000000000000000000 9.000000000000000000
1.000000000000000000 1.000000000000000000 5.000000000000000000 5.000000000000000000 7.000000000000000000
7.000000000000000000 7.000000000000000000 5.000000000000000000 5.000000000000000000 5.000000000000000000
4.000000000000000000 4.000000000000000000 5.000000000000000000 5.000000000000000000 5.023809523809523809
3.000000000000000000 5.023809523809523809 5.000000000000000000 5.095238095238095238 0.000000000000000000
```

Sample Input 6

Copy

```
10 193926872645 2752096782
5 0 8 0 0 2 6 5 4 5
5 5 5 9 7 0 3 3 6 5
0 0 0 2 7 2 8 0 5 9
4 8 2 5 8 2 4 9 2 0
8 7 3 2 8 4 7 9 8 4
4 1 0 4 9 3 7 5 8 7
1 6 2 6 5 3 5 4 7 9
7 3 7 6 3 9 3 2 2 5
8 9 3 6 3 0 8 6 4 0
0 0 9 7 6 2 1 9 7 6
```

Sample Output 6

Copy

```
346.6045935084415210714
5.000000000000000000 5.000000000000000000 5.0943878534377365339 0.000000000000000000 0.000000000000000000 2.000000000000000000 5.0314626178125788449 5.000000000000000000 5.000000000000000000 5.000000000000000000
5.000000000000000000 5.000000000000000000 5.000000000000000000 7.000000000000000000 7.000000000000000000 2.000000000000000000 3.000000000000000000 3.000000000000000000 5.000000000000000000 5.000000000000000000
0.000000000000000000 0.000000000000000000 0.000000000000000000 2.000000000000000000 7.000000000000000000 2.000000000000000000 4.000000000000000000 3.000000000000000000 5.000000000000000000 5.125850471250315379 3
4.000000000000000000 7.000000000000000000 2.000000000000000000 5.000000000000000000 8.000000000000000000 2.000000000000000000 4.000000000000000000 8.0314626178125788445 2.000000000000000000 2.000000000000000000
7.0314626178125788445 7.000000000000000000 3.000000000000000000 3.000000000000000000 8.000000000000000000 4.000000000000000000 7.000000000000000000 8.0314626178125788445 8.000000000000000000 4.000000000000000000
4.000000000000000000 2.000000000000000000 2.000000000000000000 4.000000000000000000 8.000000000000000000 3.000000000000000000 7.000000000000000000 5.000000000000000000 8.000000000000000000 7.000000000000000000
4.000000000000000000 6.000000000000000000 2.000000000000000000 6.000000000000000000 5.000000000000000000 3.000000000000000000 5.000000000000000000 4.000000000000000000 7.000000000000000000 7.062925235625157689 4
7.000000000000000000 6.000000000000000000 6.000000000000000000 6.000000000000000000 3.000000000000000000 3.000000000000000000 3.000000000000000000 3.000000000000000000 3.000000000000000000 5.000000000000000000
8.000000000000000000 8.000000000000000000 6.000000000000000000 6.000000000000000000 3.000000000000000000 2.000000000000000000 6.000000000000000000 6.000000000000000000 4.000000000000000000 4.000000000000000000
0.000000000000000000 0.000000000000000000 7.062925235625157689 4 7.000000000000000000 6.000000000000000000 2.000000000000000000 2.000000000000000000 7.062925235625157689 4 7.000000000000000000 6.000000000000000000
```