Problem A. Shops

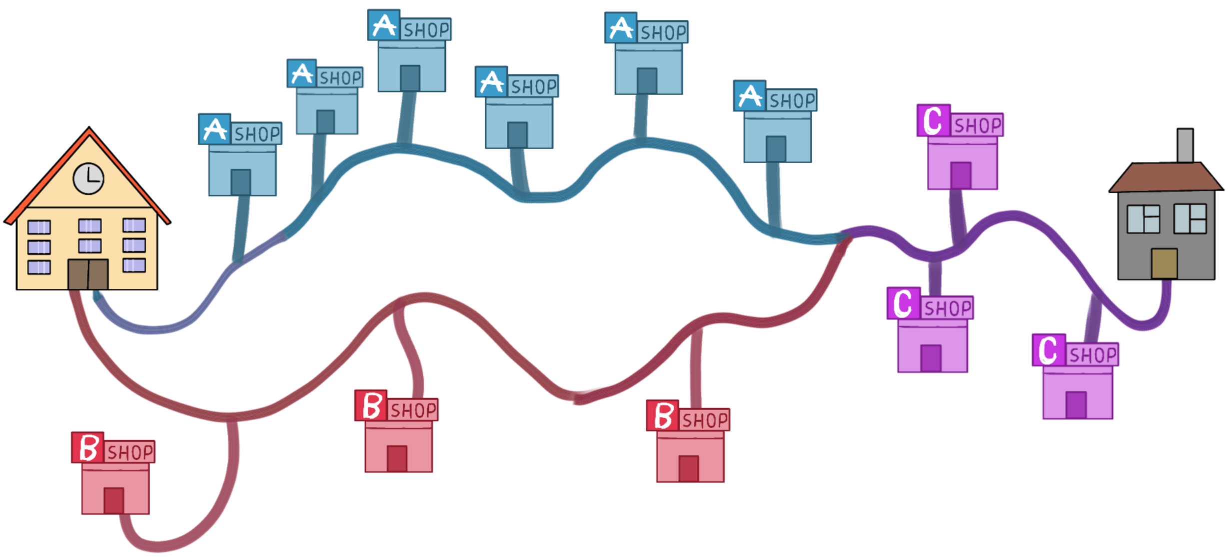
Time limit: 1 second
Memory limit: 256 megabytes

Today is Friday. Anton is finally free from school for two days! Anton can go home in two ways:

- by taking the first path, Anton can visit a shops;
- by taking the second path, Anton can visit b shops.

Anton also knows that there are c shops that are encountered on both paths (in both places). Anton is curious about how many shops are there in his city in total? Let's assume that there are no more shops in the city except for these shops.

Help him answer this question.



Above is one of the possible route options in the second example.

Input

The first line contains three integers a, b, c ($0 \leq c \leq a, b \leq 100$) — the number of shops on the first path, the second path, and on both.

Output

Print a single integer — the total number of shops in the city.

Examples

standard input	standard output
1 2 0	3
9 6 3	12

Problem B. Creating an Array

Time limit:	1 second
Memory limit:	256 megabytes

Sofiia gave Anton an array of **digits**! Although this array was not the first one he had seen, he did not consider it less interesting. After playing with the array, he did not notice how he broke it to a state where he could no longer restore the original.

He was very upset because there were almost countless ways to compose the initial array. However, he remembers an interesting property of the gift: $\sum_{i=1}^{n}\sum_{j=i}^{n} \text{concat}(a_i, a_j)$, which means the sum of concatenations of all pairs of its elements, is **maximum** among all possible arrays made from these digits consisting of the same elements, as the gift.

In other words, we take all pairs of positions i and j such that j is not to the left of i ($i \le j$). And add to the sum $\overline{a_i a_j}$, where $\overline{ab}$ means the number that will result if we write the numbers a and b in order (or $10 \cdot a + b$). This is called the concatenation of a and b.

For example, if Anton had an array $[1, 0, 3]$, then the sum would be equal to $\overline{a_1a_1} + \overline{a_1a_2} + \overline{a_1a_3} + \overline{a_2a_2} + \overline{a_2a_3} + \overline{a_3a_3} = 11 + 10 + 13 + 00 + 03 + 33 = 70$.

Help Anton and print an array that has this property. If there are multiple answers, any of them can be output.

Input

The first line contains 10 integers $c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9$ ($0 \le c_i \le 50$) — where c_i corresponds to the number of digits i in the initial array.

It is guaranteed that the sum of all numbers is greater than zero.

Output

Print an array consisting of $c_0 + c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8 + c_9$ elements, and has the same properties as the array given by Sophia.

Examples

standard input	standard output
0 0 0 0 0 2 0 0 0 0	5 5
1 0 1 1 0 0 0 0 0 0	3 2 0

Note

In the second example, there are such possible arrays:

1. $[0, 2, 3]$, the sum is equal to $\overline{a_1a_1}+\overline{a_1a_2}+\overline{a_1a_3}+\overline{a_2a_2}+\overline{a_2a_3}+\overline{a_3a_3} = 00+02+03+22+23+33 = 83$;
2. $[0, 3, 2]$, the sum is equal to $\overline{a_1a_1}+\overline{a_1a_2}+\overline{a_1a_3}+\overline{a_2a_2}+\overline{a_2a_3}+\overline{a_3a_3} = 00+03+02+33+32+22 = 92$;
3. $[2, 0, 3]$, the sum is equal to $\overline{a_1a_1}+\overline{a_1a_2}+\overline{a_1a_3}+\overline{a_2a_2}+\overline{a_2a_3}+\overline{a_3a_3} = 22+20+23+00+03+33 = 101$;
4. $[2, 3, 0]$, the sum is equal to $\overline{a_1a_1}+\overline{a_1a_2}+\overline{a_1a_3}+\overline{a_2a_2}+\overline{a_2a_3}+\overline{a_3a_3} = 22+23+20+33+30+00 = 128$;
5. $[3, 0, 2]$, the sum is equal to $\overline{a_1a_1}+\overline{a_1a_2}+\overline{a_1a_3}+\overline{a_2a_2}+\overline{a_2a_3}+\overline{a_3a_3} = 33+30+32+00+02+22 = 119$;
6. $[3, 2, 0]$, the sum is equal to $\overline{a_1a_1}+\overline{a_1a_2}+\overline{a_1a_3}+\overline{a_2a_2}+\overline{a_2a_3}+\overline{a_3a_3} = 33+32+30+22+20+00 = 137$.

Problem C. Everyone Loves Permutations

Time limit:	1 second
Memory limit:	256 megabytes

A permutation of length n is an array of length n containing all integers from 1 to n, and all its elements are pairwise distinct.

Having grown up and played with arrays, Anton moved on to studying more interesting arrays — permutations. While writing his thesis, he faced a very difficult task.

He has a permutation p of length n and an integer k. He decided to construct a two-dimensional array a with sizes $(k+1) \times n$.

1. $a_{0j} = j$ for all j $(1 \le j \le n)$;
2. $a_{ij} = a_{(i-1)p_j}$ for all i $(1 \le i \le k)$ and j $(1 \le j \le n)$.

Let $p = [5, 3, 1, 4, 2]$ and $k = 3$, then we have the following array.

a_{ij}	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$i=0$	1	2	3	4	5
$i=1$	5	3	1	4	2
$i=2$	2	1	5	4	3
$i=3$	3	5	2	4	1

For each x $(1 \le x \le n)$, he wants to know the sum of all j such that $a_{ij} = x$, where $1 \le i \le k$. In other words, he wants to find the sum of k numbers — the indices of the number x in each a_i.

Consider the last example. If $x = 1$, the answer will be $3 + 2 + 5 = 10$.

After some deliberation and simple ideas, Anton managed to solve this problem quickly. Now he wants to check if you can solve it too.

Input

The first line of the input contains two integers n, k $(1 \le n \le 10^6, 1 \le k \le 10^9)$ — the length of the permutation and the number of repetitions of operations, respectively.

The second line contains the permutation p $(1 \le p_i \le n)$.

Output

Print n integers, where the i-th number is the answer for $x = i$.

Scoring

1. (8 points): $k = 1$;
2. (17 points): $p_i = i$;
3. (26 points): $n \le 2000, k \le 2000$;
4. (28 points): $n \le 2000$, for any i and j, there exists a k such that $p[p[p \dots p[i] \dots]] = j$, where the nesting is taken k times;
5. (9 points): for any i and j, there exists a k such that $p[p[p \dots p[i] \dots]] = j$, where the nesting is taken k times;
6. (12 points): without additional restrictions.

Examples

standard input	standard output
3 2 2 1 3	3 3 6
5 3 5 3 1 4 2	10 9 8 12 6

Problem D. Mole

Time limit: 1 second
Memory limit: 256 megabytes

After a difficult research, Anton decided to relax at his country house. He has a beautiful garden with many different flowers. But, oh no, upon arrival he saw a significant number of holes in the ground. It's a mole!

Now, armed with a shovel, Anton will be waiting for the mole. The mole can emerge from any hole. Anton wants to choose a position so that in the worst case, he runs to the mole in the **minimum** amount of time.

The garden can be represented as a matrix $n \times m$, where n is the number of rows, and m is the number of columns. Rows are numbered from top to bottom, from 1 to n. Columns are numbered from left to right, from 1 to m. Thus, the cell with index $(1; 1)$ is located in the upper left corner.

Each cell of the garden $a_{i,j}$ describes the state of this cell:

- $a_{i,j}$ = "." — this cell does not contain flowers or holes;
- $a_{i,j}$ = "F" — this cell contains flowers;
- $a_{i,j}$ = "H" — this cell contains a hole.

Anton also knows that the number of holes does not exceed 100.

As a person who has put a lot of time into these flowers, your heart cannot bear trampling on them. Therefore, you need to find a path in such a way that it does not pass through them.

At any moment in time, Anton can move from position (x, y) to any of the following positions: $(x-1, y)$, $(x+1, y)$, $(x, y-1)$, $(x, y+1)$, provided that the new position does not contain flowers and is inside the garden.

Find all positions $(x; y)$ from which Anton will run to the moles in the worst case in the minimum amount of time.

Input

The first line contains two integers n, m ($1 \le n \cdot m \le 2 \cdot 10^5$) — the length and width of the garden.

The next n lines contain m characters each — the description of the garden.

It is guaranteed that from each cell that does not contain flowers, it is possible to reach any other cell that does not contain flowers by moving through cells that do not contain flowers.

It is guaranteed that there is at least one hole, and the number of holes in the garden does not exceed 100.

Output

In the first line, print a single integer x ($1 \le x \le n \cdot m$) — the number of optimal positions.

In each of the next x lines, print the optimal positions $(x; y)$ for waiting for the mole ($1 \le x \le n$, $1 \le y \le m$).

Positions can be printed in any order.

Scoring

Let k be the number of holes in the garden.

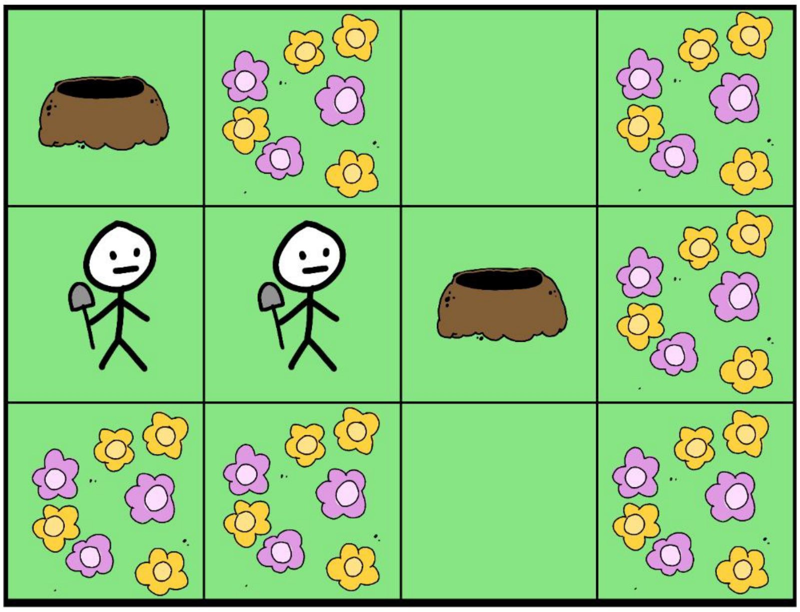
1. (6 points): $n = 1, m = 2$;
2. (9 points): $n = 1$;

3. (15 points): $k = 1, n \cdot m \leq 5 \cdot 10^3$;
4. (22 points): $n \cdot m \leq 5 \cdot 10^3$;
5. (17 points): $k = 1$;
6. (31 points): without additional restrictions.

Examples

standard input	standard output
3 4 HF.F ..HF FF.F	2 2 1 2 2
4 9 ......FFH .F..FHFF. HF....... .FHF..FFF	2 1 6 3 4

Note



Above is the first example and the optimal waiting positions are marked.

Problem E. GCD, Sum, Multiply. What?...

Time limit:	3 seconds
Memory limit:	256 megabytes

The author has used up all the creative skills on previous problems, so Anton won't be tortured in this statement. He will just give you an interesting problem.

You are given an array a consisting of n integers. You are also given q queries $[l; r]$. For each query, find the maximum value of $\text{sum}[tl; tr] \times \gcd[tl; tr]$ over all pairs $(tl; tr)$, where

- $l \le tl \le tr \le r$;
- $\text{sum}[tl; tr]$ — the sum of all numbers in the segment $[tl; tr]$;
- $\gcd[tl; tr]$ — the greatest common divisor of all numbers in the segment $[tl; tr]$.

The greatest common divisor of two numbers a and b is the largest positive integer x that divides both a and b.

The greatest common divisor of a set of numbers is the largest positive integer x that divides all elements of the set.

Input

The first line contains two integers n, q ($1 \le n, q \le 2 \cdot 10^5$) — the number of elements in the array and the number of queries, respectively.

The second line contains n integers a_i ($1 \le a_i \le 6 \cdot 10^6$) — the description of the array.

Each of the next q lines contains two integers l, r ($1 \le l \le r \le n$) — the description of the queries.

Output

Print q integers — the answers to the queries.

Scoring

1. (4 points): $n \le 3$;
2. (8 points): $n, q \le 10^3$;
3. (5 points): $n \le 10^3$;
4. (17 points): $n, q \le 10^5$;
5. (14 points): $n \le 10^5$;
6. (5 points): $a_i \le 20$;
7. (7 points): $a_i \le 10^3$;
8. (16 points): $l = 1$;
9. (24 points): no additional constraints.

Examples

standard input	standard output
3 2 3 3 2 1 3 2 3	18 9
8 6 2 4 8 8 8 2 4 16 1 8 2 5 3 4 2 4 7 7 3 6	256 192 128 128 16 192

Note

In the first example, there are following segments:

- $[1;1]$ — $\mathrm{sum}[1;1] \times \mathrm{gcd}[1;1] = 3 \times 3 = 9$;
- $[1;2]$ — $\mathrm{sum}[1;2] \times \mathrm{gcd}[1;2] = 6 \times 3 = 18$;
- $[1;3]$ — $\mathrm{sum}[1;3] \times \mathrm{gcd}[1;3] = 8 \times 1 = 8$;
- $[2;2]$ — $\mathrm{sum}[2;2] \times \mathrm{gcd}[2;2] = 3 \times 3 = 9$;
- $[2;3]$ — $\mathrm{sum}[2;3] \times \mathrm{gcd}[2;3] = 5 \times 1 = 5$;
- $[3;3]$ — $\mathrm{sum}[3;3] \times \mathrm{gcd}[3;3] = 2 \times 2 = 4$.