A - Divisible

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 100 points

Problem Statement

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

Extract all elements of A that are multiples of K, divide them by K, and print the quotients.

Constraints

- $1 \leq N, K \leq 100$
- $1 \leq A_1 < A_2 < \ldots < A_N \leq 100$
- A has at least one multiple of K.
- All given numbers are integers.

Input

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

$$
\begin{array}{l}
N \quad K \\
A_1 \quad A_2 \quad \ldots \quad A_N
\end{array}
$$

Output

Divide all elements of A that are multiples of K and print the quotients in ascending order with spaces in between.

Sample Input 1

```
5 2
2 5 6 7 10
```

Sample Output 1

```
1 3 5
```

The multiples of 2 among the elements in A are $2, 6$, and 10. Divide them by 2 to get $1, 3$, and 5, and print them in ascending order with spaces in between.

Sample Input 2

```
3 1
3 4 7
```

Sample Output 2

```
3 4 7
```

Sample Input 3

```
5 10
50 51 54 60 65
```

Sample Output 3

```
5 6
```

B - Substring

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 200 points

Problem Statement

You are given a string S consisting of lowercase English letters. How many different non-empty substrings does S have?

A substring is a contiguous subsequence. For example, `xxx` is a substring of `yxxxy` but not of `xxyxx`.

Constraints

- S is a string of length between 1 and 100, inclusive, consisting of lowercase English letters.

Input

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

$$S$$

Output

Print the answer.

Sample Input 1

```
yay
```

Sample Output 1

```
5
```

S has the following five different non-empty substrings:

- a
- y
- ay
- ya
- yay

Sample Input 2

```
aababc
```

Sample Output 2

```
17
```

Sample Input 3

```
abracadabra
```

Sample Output 3

```
54
```

C - Ideal Holidays

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 350 points

Problem Statement

In the Kingdom of AtCoder, a week consists of $A+B$ days, with the first through A-th days being holidays and the $(A+1)$-th through $(A+B)$-th being weekdays.

Takahashi has N plans, and the i-th plan is scheduled D_i days later.

He has forgotten what day of the week it is today. Determine if it is possible for all of his N plans to be scheduled on holidays.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq A, B \leq 10^9$
- $1 \leq D_1 < D_2 < \ldots < D_N \leq 10^9$

Input

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

$$
\begin{array}{l}
N \quad A \quad B \\
D_1 \quad D_2 \quad \ldots \quad D_N
\end{array}
$$

Output

Print `Yes` in a single line if it is possible for all of Takahashi's N plans to be scheduled on holidays, and `No` otherwise.

Sample Input 1

```
3 2 5
1 2 9
```

Sample Output 1

```
Yes
```

In this input, a week consists of seven days, with the first through second days being holidays and the third through seventh days being weekdays.

Let us assume today is the seventh day of the week. In this case, one day later would be the first day of the week, two days later would be the second day of the week, and nine days later would also be the second day of the week, making all plans scheduled on holidays. Therefore, it is possible for all of Takahashi's N plans to be scheduled on holidays.

Sample Input 2

```
2 5 10
10 15
```

Sample Output 2

```
No
```

Sample Input 3

```
4 347 347
347 700 705 710
```

Sample Output 3

```
Yes
```

D - Popcount and XOR

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 400 points

Problem Statement

You are given non-negative integers a, b, and C. Determine if there is a pair of non-negative integers (X, Y) that satisfies all of the following five conditions. If such a pair exists, print one.

- $0 \leq X < 2^{60}$
- $0 \leq Y < 2^{60}$
- $\text{popcount}(X) = a$
- $\text{popcount}(Y) = b$
- $X \oplus Y = C$

Here, $\oplus$ denotes the bitwise XOR.

If multiple pairs (X, Y) satisfy the conditions, you may print any of them.

▶ What is popcount?

▶ What is bitwise XOR?

Constraints

- $0 \leq a \leq 60$
- $0 \leq b \leq 60$
- $0 \leq C < 2^{60}$
- All input values are integers.

Input

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

$$a \quad b \quad C$$

Output

If there is a pair of non-negative integers that satisfies the conditions, choose one such pair (X, Y) and print X and Y in this order, with a space in between. If no such pair exists, print `-1`.

Sample Input 1

```
3 4 7
```

Sample Output 1

```
28 27
```

The pair $(X, Y) = (28, 27)$ satisfies the conditions. Here, X and Y in binary are `11100` and `11011`, respectively.

- X in binary is `11100`, so $\mathrm{popcount}(X) = 3$.
- Y in binary is `11011`, so $\mathrm{popcount}(Y) = 4$.
- $X \oplus Y$ in binary is `00111`, so $X \oplus Y = 7$.

If multiple pairs of non-negative integers satisfy the conditions, you may print any of them, so printing `42 45`, for example, would also be accepted.

Sample Input 2

```
34 56 998244353
```

Sample Output 2

```
-1
```

No pair of non-negative integers satisfies the conditions.

Sample Input 3

```
39 47 530423800524412070
```

Sample Output 3

```
540431255696862041 10008854347644927
```

Note that the values to be printed may not fit in 32-bit integers.

E - Set Add Query

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 500 points

Problem Statement

There is an integer sequence $A = (A_1, A_2, \ldots, A_N)$ of length N, where all elements are initially set to 0. Also, there is a set S, which is initially empty.

Perform the following Q queries in order. Find the value of each element in the sequence A after processing all Q queries. The i-th query is in the following format:

- An integer x_i is given. If the integer x_i is contained in S, remove x_i from S. Otherwise, insert x_i to S. Then, for each $j = 1, 2, \ldots, N$, add $|S|$ to A_j if $j \in S$.

Here, $|S|$ denotes the number of elements in the set S. For example, if $S = \{3, 4, 7\}$, then $|S| = 3$.

Constraints

- $1 \leq N, Q \leq 2 \times 10^5$
- $1 \leq x_i \leq N$
- All given numbers are integers.

Input

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

$$
\begin{array}{l}
N \quad Q \\
x_1 \quad x_2 \quad \ldots \quad x_Q
\end{array}
$$

Output

Print the sequence A after processing all queries in the following format:

$$
A_1 \quad A_2 \quad \ldots \quad A_N
$$

Sample Input 1

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

Sample Output 1

```
6 2 2
```

In the first query, 1 is inserted to S, making $S = \{1\}$. Then, $|S| = 1$ is added to A_1. The sequence becomes $A = (1, 0, 0)$.

In the second query, 3 is inserted to S, making $S = \{1, 3\}$. Then, $|S| = 2$ is added to A_1 and A_3. The sequence becomes $A = (3, 0, 2)$.

In the third query, 3 is removed from S, making $S = \{1\}$. Then, $|S| = 1$ is added to A_1. The sequence becomes $A = (4, 0, 2)$.

In the fourth query, 2 is inserted to S, making $S = \{1, 2\}$. Then, $|S| = 2$ is added to A_1 and A_2. The sequence becomes $A = (6, 2, 2)$.

Eventually, the sequence becomes $A = (6, 2, 2)$.

Sample Input 2

```
4 6
1 2 3 2 4 2
```

Sample Output 2

```
15 9 12 7
```

F - Non-overlapping Squares

Time Limit: 3 sec / Memory Limit: 1024 MB

Score: 525 points

Problem Statement

There is an $N \times N$ grid, and the cell at the i-th row from the top and the j-th column from the left $(1 \leq i, j \leq N)$ contains the integer $A_{i,j}$.

You are given an integer M. When choosing three non-overlapping $M \times M$ grids, find the maximum possible sum of the integers written in the chosen grids.

▶ Formal definition of the problem

Constraints

- $2 \leq N \leq 1000$
- $1 \leq M \leq N/2$
- $0 \leq A_{i,j} \leq 10^9$
- All input values are integers.

Input

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

$$
\begin{array}{llll}
N & M & & \\
A_{1,1} & A_{1,2} & \ldots & A_{1,N} \\
A_{2,1} & A_{2,2} & \ldots & A_{2,N} \\
\vdots & \vdots & \ddots & \vdots \\
A_{N,1} & A_{N,2} & \ldots & A_{N,N}
\end{array}
$$

Output

Print the answer.

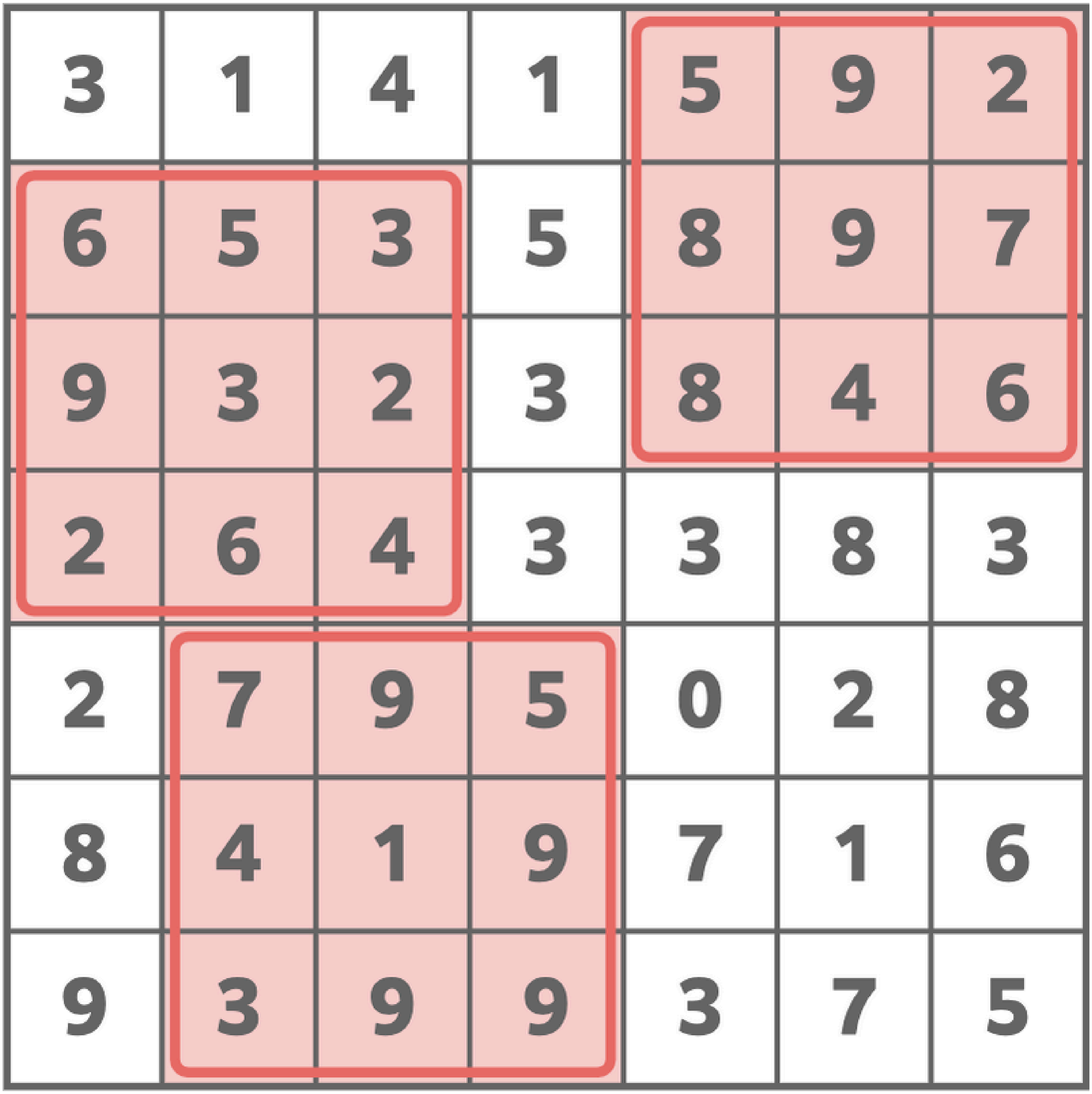
Sample Input 1

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

Sample Output 1

```
154
```

From the given grid, if we choose three 3×3 grids as shown in the figure below (this corresponds to setting $(i_1, j_1, i_2, j_2, i_3, j_3) = (1, 5, 2, 1, 5, 2)$), the sum of the numbers written in the chosen grids will be 154.



There is no way to make the sum 155 or greater while satisfying the conditions in the problem statement, so print 154.

Sample Input 2

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

Sample Output 2

```
27
```

The following choice is optimal.

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

Sample Input 3

```
16 4
74 16 58 32 97 52 43 51 40 58 13 24 65 11 63 29
98 75 40 77 15 50 83 85 35 46 38 37 56 38 63 55
95 42 10 70 53 40 25 10 70 32 33 19 52 79 74 58
33 91 53 11 65 63 78 77 81 46 81 63 11 82 55 62
39 95 92 69 77 89 14 84 53 78 71 81 66 39 96 29
74 26 60 55 89 35 32 64 17 26 74 92 84 33 59 82
23 69 10 95 94 14 58 58 97 95 62 58 72 55 71 43
93 77 27 87 74 72 91 37 53 80 51 71 37 35 97 46
81 88 26 79 78 30 53 68 83 28 59 28 74 55 20 86
93 13 25 19 53 53 17 24 69 14 67 81 10 19 69 90
88 83 62 92 22 31 27 34 67 48 42 32 68 14 96 87
44 69 25 48 68 42 53 82 44 42 96 31 13 56 68 83
63 87 24 75 16 70 63 99 95 10 63 26 56 12 77 49
94 83 69 95 48 41 40 97 45 61 26 38 83 91 44 31
43 69 54 64 20 60 17 15 62 25 58 50 59 63 88 70
72 95 21 28 41 14 77 22 64 78 33 55 67 51 78 40
```

Sample Output 3

```
3295
```

The following choice is optimal.

74	16	58	32	97	52	43	51	40	58	13	24	65	11	63	29
98	75	40	77	15	50	83	85	35	46	38	37	56	38	63	55
95	42	10	70	53	40	25	10	70	32	33	19	52	79	74	58
33	91	53	11	65	63	78	77	81	46	81	63	11	82	55	62
39	95	92	69	77	89	14	84	53	78	71	81	66	39	96	29
74	26	60	55	89	35	32	64	17	26	74	92	84	33	59	82
23	69	10	95	94	14	58	58	97	95	62	58	72	55	71	43
93	77	27	87	74	72	91	37	53	80	51	71	37	35	97	46
81	88	26	79	78	30	53	68	83	28	59	28	74	55	20	86
93	13	25	19	53	53	17	24	69	14	67	81	10	19	69	90
88	83	62	92	22	31	27	34	67	48	42	32	68	14	96	87
44	69	25	48	68	42	53	82	44	42	96	31	13	56	68	87
63	87	24	75	16	70	63	99	95	10	63	26	56	12	77	49
94	83	69	95	48	41	40	97	45	61	26	38	83	91	44	31
43	69	54	64	20	60	17	15	62	25	58	50	59	63	88	70
72	95	21	28	41	14	77	22	64	78	33	55	67	51	78	40

G - Grid Coloring 2

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 600 points

Problem Statement

There is an $N \times N$ grid where each cell contains an integer between 0 and 5, inclusive. Let (i, j) denote the cell at the i-th row from the top and the j-th column from the left $(1 \le i, j \le N)$. The integer written in cell (i, j) is $A_{i,j}$.

You can perform the following operation any number of times, possibly zero:

- Choose a cell (i, j) with 0 written in it and an integer x between 1 and 5, inclusive. Change the number written in the chosen cell to x.

After the operations, let $B_{i,j}$ be the integer written in cell (i, j). The **cost** of the grid is defined as the sum of the squares of the differences between the integers written in adjacent cells. In other words, the cost is represented by the following formula:

$$\sum_{i=1}^{N}\sum_{j=1}^{N-1}(B_{i,j} - B_{i,j+1})^2 + \sum_{i=1}^{N-1}\sum_{j=1}^{N}(B_{i,j} - B_{i+1,j})^2$$

Among all possible states of the grid after the operations, find the one with the minimum cost.

If multiple grid states have the minimum cost, you may print any of them.

Constraints

- $1 \le N \le 20$
- $0 \le A_{i,j} \le 5 \ (1 \le i \le N, 1 \le j \le N)$
- All input values are integers.

Input

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

$$\begin{array}{llll} N \\ A_{1,1} & A_{1,2} & \ldots & A_{1,N} \\ A_{2,1} & A_{2,2} & \ldots & A_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N,1} & A_{N,2} & \ldots & A_{N,N} \end{array}$$

Output

Print N lines. The i-th line $(1 \leq i \leq N)$ should contain $B_{i,1}, B_{i,2}, \ldots, B_{i,N}$ in this order, with spaces in between, after performing operations to minimize the cost.

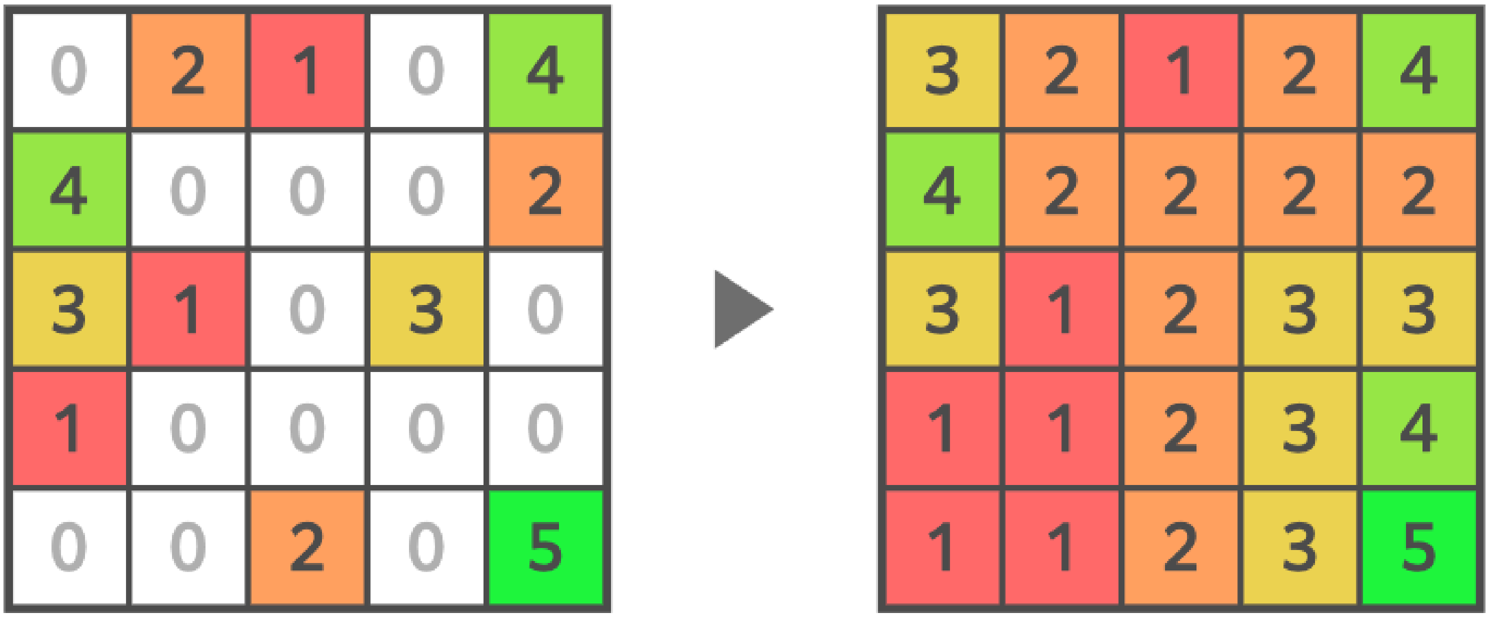
Sample Input 1

```
5
0 2 1 0 4
4 0 0 0 2
3 1 0 3 0
1 0 0 0 0
0 0 2 0 5
```

Sample Output 1

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

The given grid is as follows:



After performing operations to achieve the state shown on the right of the figure, the cost will be $2^2 \times 6 + 1^2 \times 18 + 0^2 \times 16 = 42$.

The cost cannot be 41 or less, so printing the corresponding $B_{i,j}$ for this state would be accepted.

Sample Input 2

```
3
0 0 0
0 0 0
0 0 0
```

Sample Output 2

```
0 0 0
0 0 0
0 0 0
```

The cost is already 0 from the beginning, so not performing any operations minimizes the cost.

If multiple grid states have the minimum cost, you may print any of them, so the following output would also be accepted:

```
2 2 2
2 2 2
2 2 2
```

Sample Input 3

```
10
1 0 0 3 0 0 0 0 0 0
1 0 0 4 0 1 0 5 0 0
0 0 0 0 0 0 2 0 3 0
0 0 2 0 0 0 4 0 0 3
0 3 4 3 3 0 3 0 0 5
4 1 3 4 4 0 2 1 0 0
2 0 1 0 5 2 0 1 1 5
0 0 0 5 0 0 3 2 4 0
4 5 0 0 3 2 0 3 5 0
4 0 0 5 0 0 0 3 0 5
```

Sample Output 3

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