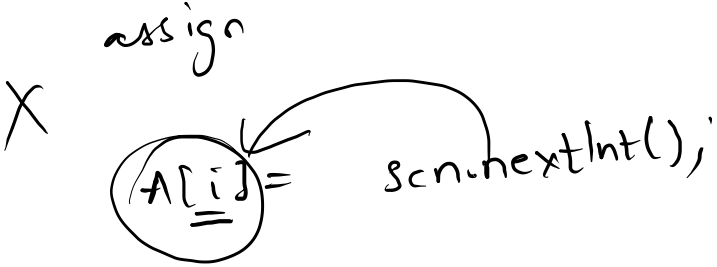
for - each.

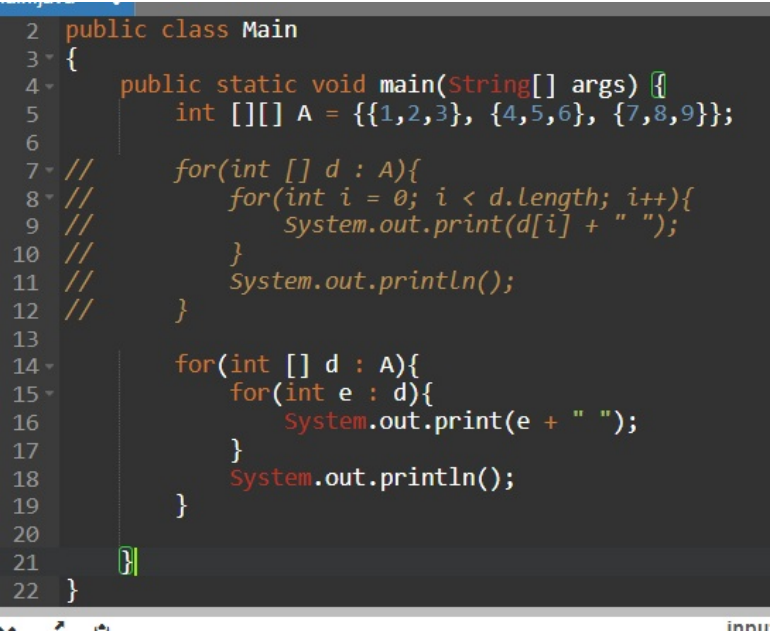
{ 1, 2, 3, 4, 5, 6 }
0 1 2 3 4 5

A[0] A[1] ------ A[5]

Traditional for loop.
↳ till now.

```java
{
    public static void main(String[] args) {
        int [] A = {1,2,3,4,5,6};

        for(int i = 0; i < A.length; i++){
            int ele = A[i];
            System.out.print(ele + " ");
        }

        System.out.println();

        for(int ele : A){
            System.out.print(ele + " ");
        }

        String [] B = {"abc", "def", "ghi"};
        for(String s : B){
            System.out.print(s + " ");
        }
```





```java
public class Main
{
    public static void main(String[] args) {
        int [][] A = {{1,2,3}, {4,5,6}, {7,8,9}};

//      for(int [] d : A){
//          for(int i = 0; i < d.length; i++){
//              System.out.print(d[i] + " ");
//          }
//          System.out.println();
//      }

        for(int [] d : A){
            for(int e : d){
                System.out.print(e + " ");
            }
            System.out.println();
        }

    }
}
```

Modify The Matrix

Problem | Submissions | Leaderboard | Discussions

Once upon a time, there was a company that was developing a system to track the inventory levels of different products in different **warehouses**. They had a boolean matrix **Mat** of size **M X N**, where each cell represented the availability of a product in a specific warehouse. If the value of a cell was **true (or 1)**, it meant that the **product** was available in that **warehouse**.

The company wanted to modify the matrix in such a way that if a cell, **Mat[i][j]**, was **true**, then all cells in the **ith row** and **jth column** of the matrix would also be set to **true**. This would ensure that if a **product** was available in a particular **warehouse**, all the products in that **row** and **column** would also be considered available.

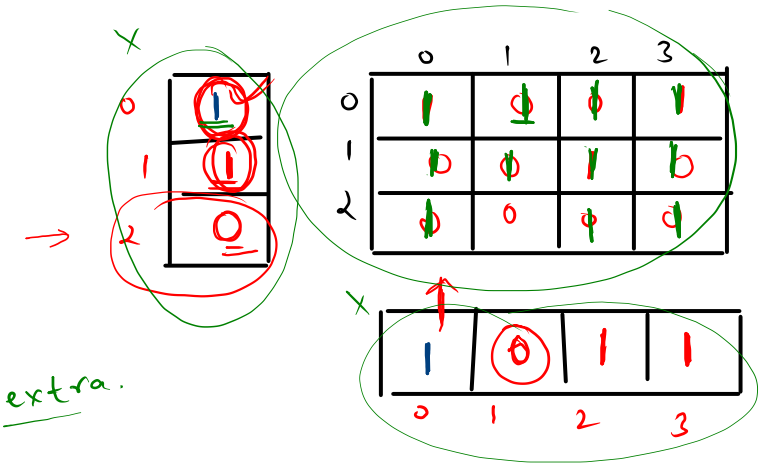
Can you write a program thet modify the matrix such that if a matrix cell **Mat[i][j]** is **1 (or true)** then make all the cells of **ith row** and **jth column** as **1**.

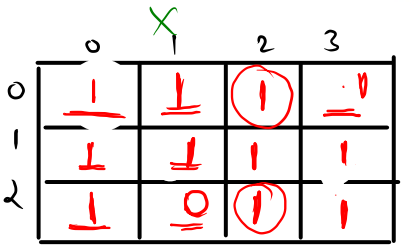
Sample Input 0

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

Sample Output 0

```
1 1 1 1
1 1 1 1
1 0 1 1
```





2 extra.

or →

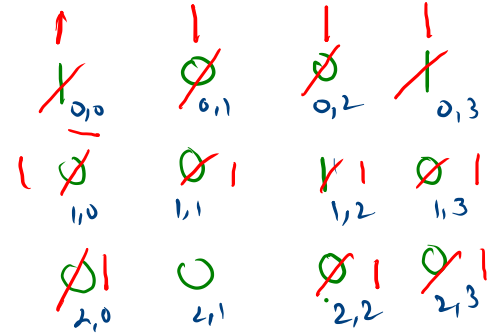
marking / update

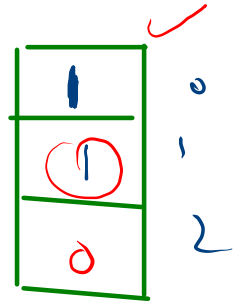
for (i=0 — i<n ; i++)
for (j=0 ; j<m ; j++)
{

}

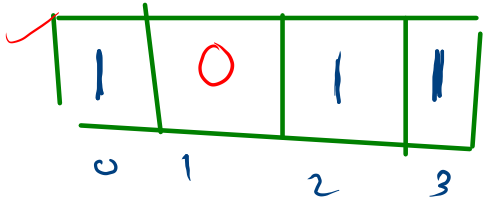
marking

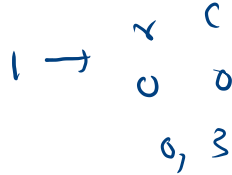
updating.



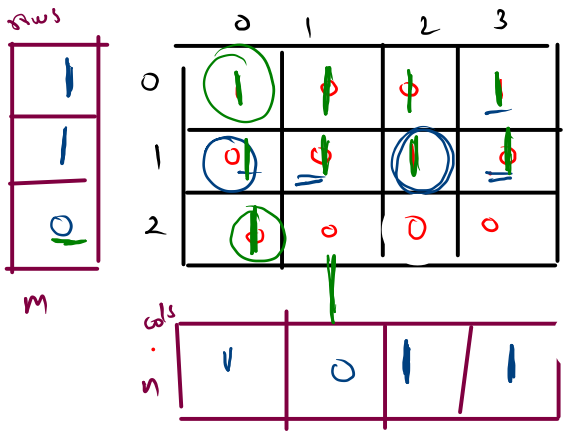


```
3
4
1 0 0 1
0 0 1 0
0 0 0 0
```





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




```java
//solve
int [] rows = new int[m];
int [] cols = new int[n];
//marking
for(int i = 0; i < m; i++){
    for(int j = 0; j < n; j++){
        if(A[i][j] == 1){
            rows[i] = 1;
            cols[j] = 1;
        }
    }
}
//update
for(int i = 0; i < m; i++){
    for(int j = 0; j < n; j++){
        if(rows[i] == 1 || cols[j] == 1){
            A[i][j] = 1;
        }
    }
}
```

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int m = scn.nextInt();
        int n = scn.nextInt();
        int [][] A = new int[m][n];
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }
        //solve
        int [] rows = new int[m];
        int [] cols = new int[n];
        //marking
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                if(A[i][j] == 1){
                    rows[i] = 1;
                    cols[j] = 1;
                }
            }
        }
        //update
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                if(rows[i] == 1 || cols[j] == 1){
                    A[i][j] = 1;
                }
            }
        }
        //print
        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                System.out.print(A[i][j] + " ");
            }
            System.out.println();
        }



    }
}
```

Shift Matrix Row-Wise

Problem | Submissions | Leaderboard | Discussions

Once upon a time, there was a group of students who were working on a project to design a gaming platform. They had a **2D grid** of N * N size which represented the game board. Each cell of the grid had some data associated with it.

One day, they encountered a problem where they had to shift the elements of the grid row-wise in clock wise direction by a certain number of positions, k. This was necessary to create a more interesting and challenging gaming experience for their users.

The students decided to write a Java program to solve this problem. They came up with an algorithm to **shift** the elements of the grid row-wise by k positions. After implementing the algorithm, they were able to shift the elements of each **row** by **k** positions.

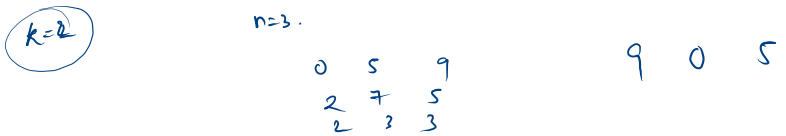
Write a program that shift each row of matrix by **k**.

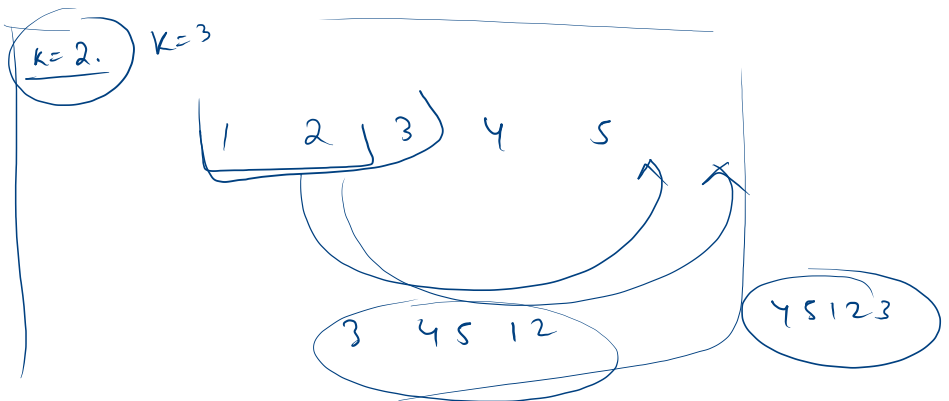
Sample Input 0

```
3
0 5 9
2 7 5
2 3 3
2
```

Sample Output 0

```
9 0 5
5 2 7
3 2 3
```



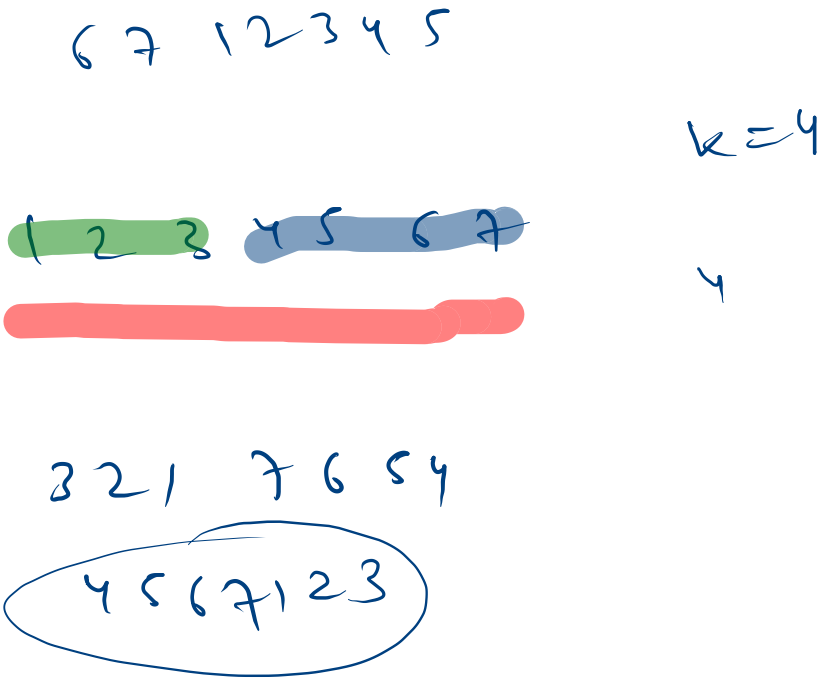


Sample Input 0

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

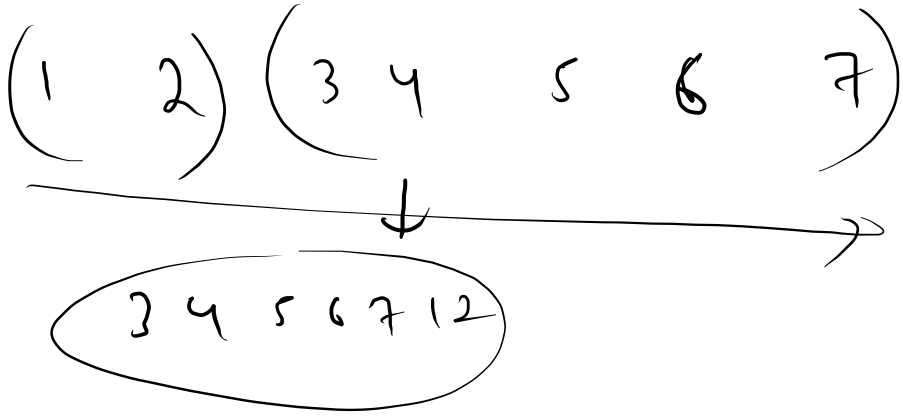
Sample Output 0

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



Rotate Left

K = 2



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void reverse(int [] d, int i, int j){
        while(i < j){
            int tmp = d[i];
            d[i] = d[j];
            d[j] = tmp;
            i++;
            j--;
        }
    }


    public static void rotate(int [] d, int k){
        //roatate left
        int n = d.length;
        k = k % n;
        reverse(d, 0, k-1);
        reverse(d, k , n-1);
        reverse(d, 0, n-1);
    }
```

```java
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [][] A = new int[n][n];
        for(int i = 0; i < n; i++){
            for(int j = 0; j < n; j++){
                A[i][j] = scn.nextInt();
            }
        }
        int k = scn.nextInt();

        //logic
        for(int [] d : A){
            rotate(d, k);
        }

        //print
       for(int i = 0; i < n; i++){
            for(int j = 0; j < n; j++){
                System.out.print(A[i][j] + " ");
            }
           System.out.println();
        }
    }
}
```

Print Characters

Amy is a high school student who is passionate about **coding**. One day, her computer science teacher gives the class an assignment to **print all the characters of a given string in separate lines.**

Amy immediately gets to work and writes a simple program. However, she feels that her solution is too basic and wants to find a more efficient way to solve the problem.

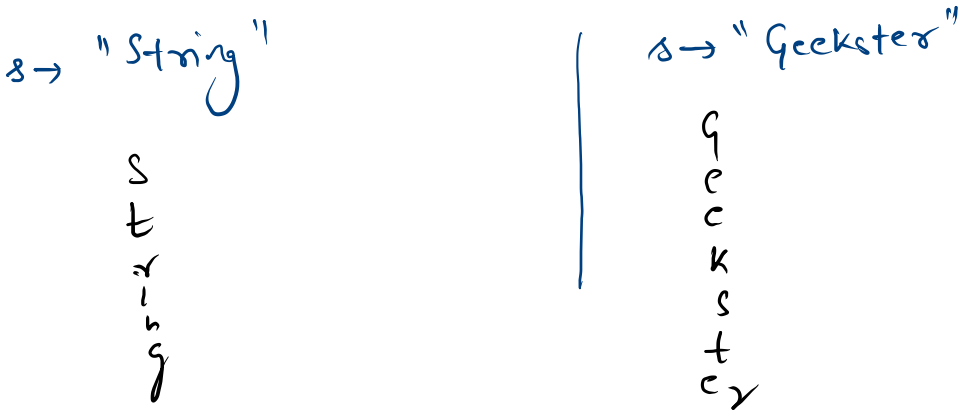
can you help Amy by writing a program thst print all the characters of a given string in separate lines.

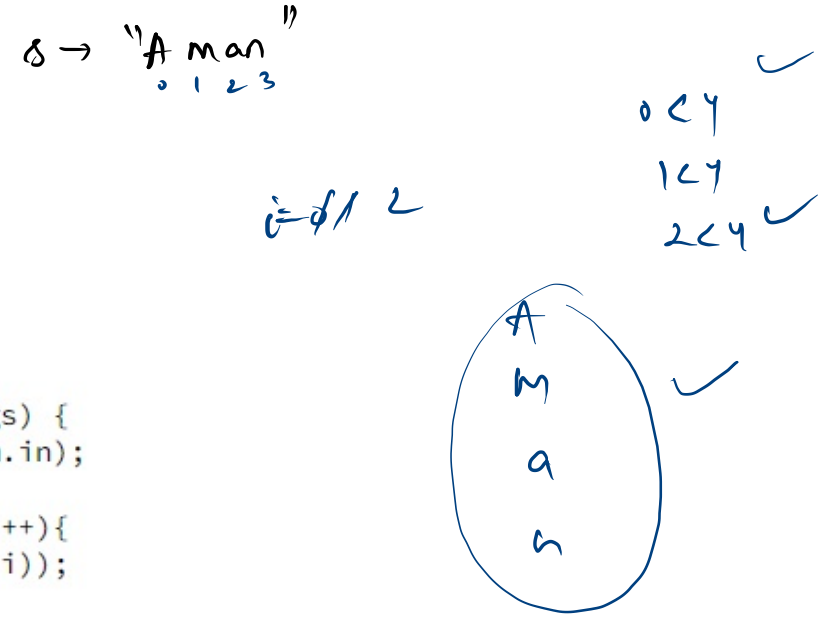
Sample Input 0

```
String
```

Sample Output 0

```
S
t
r
i
n
g
```






```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String s = scn.next();
        for(int i = 0; i < s.length(); i++){
            System.out.println(s.charAt(i));
        }
    }
}
```

ideally

- first → dry run → Aman's code.
- 2nd → write by yourself
- 3rd → try HW
- 4th → visit A if possible try.

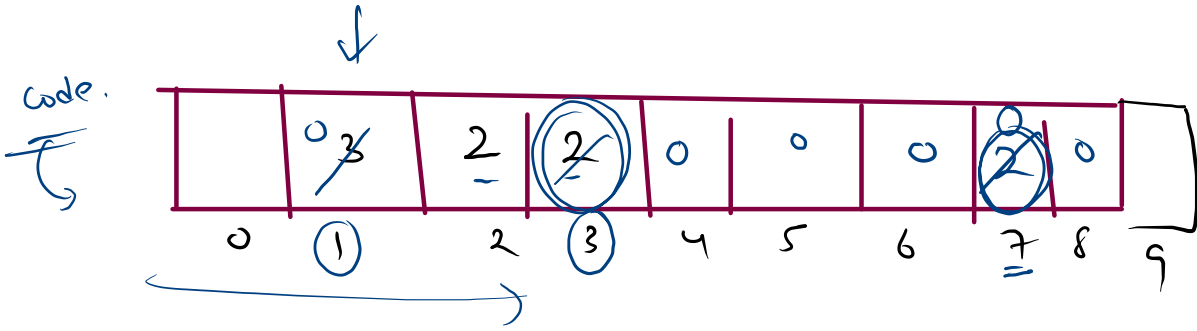
K - most freq.

0-9 → 10

Sample Input 0

```
7
1 1 1 2 2 3 3
2
```

K=3.



Sample Output 0

```
1 3
```

for (k times)

{ → max

1

}

12345

↓

k=2

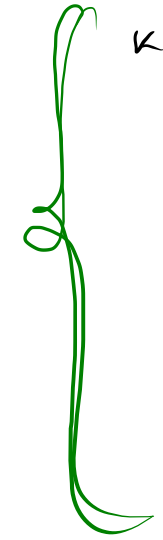
34512

1000) 12345 (12

12000

345

12345 % 1000



k=2

A = n / 10^{k+1} = 12

B = n % 10^{k+1} = 345

$B \times 10^{k} + A$

345 00

+12

34512

?

1 2 3 4 5

k = -2

4 5 1 2 3

k = 2

1 2 3 4 5 6 7

6 7 1 2 3 4 5

1 %

A → $n / 10^k$ = 12345

B → $n \% 10^k$ = 67

res → $B \times 10^{len-k} + A$