1. The problem statement given

HATFD1010

Rotate a String Left by K Positions

Write a program to rotate a given string s left by k positions without using any built-in string functions.

For example, rotating "abcdef" by 2 would give "cdefab".

Instructions: Do not use built-in rotation or substring functions. Implement the logic manually.

2. The program

```
import java.util.*;

class Main {

    public static String rotateLeft(String s, int k) {
        int n = s.length();
        if (n == 0) return s;
        k = myMod(k, n);
        StringBuilder rotated = new StringBuilder(n);
        for (int i = 0; i < n; i++) {
            rotated.append(s.charAt(myMod(i + k, n)));
        }
        return rotated.toString();
    }



public static int myMod(int a, int b) {
```

```
        if (b == 0) {
            throw new ArithmeticException("Modulo by zero is not allowed.");
        }
        while (a < 0) {
            a += b;
        }
        return a % b;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str1 = sc.nextLine();
        int rotate = sc.nextInt();
        sc.close();
        try{
            String rotated = rotateLeft(str1, rotate);
            System.out.println(rotated);
        } catch (ArithmeticException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }
}
```

3. **Sample inputs** along with their respective outputs

Case 1

Input 1: str1-abcdefg

Rotate- 3

Output 1:

defgab

Explanation: The string "abcdefg" is rotated left by 3 positions.

Case 2

Input 1: str1-a

Rotate- 3

Output 1:

a

Explanation: The string "a" is rotated left by 3 positions. Because the string length is 1, a rotation of 3 (or any number greater than 0) is equivalent to a rotation of 0, leaving the string unchanged.

Case 3

Input 3: str1-xyz

Rotate- 3

Output 3:

yzx

Explanation: The string "xyz" is rotated left by 1 position.