

PPL Assignment no 1

name - Archit Gaikwad

SE Comp A 54

1)Juggling algorithm is one of the efficient algorithms used for array rotation. Now, let us see

about juggling algorithm and a program to rotate an array using the same algorithm.

Juggling Algorithm :

In this method, divide the array into M sets, where $M = \text{GCD}(n, k)$, and then rotate the elements

in each set.

From the number of elements (n) of the array and number of rotations (k) to be made to the

array, the $\text{GCD}(n, k)$ number of blocks are made.

Then in each block, shifting will take place to the corresponding elements in the block.

After all the elements in all the blocks are shifted, the array will be rotated for the given number

of times.

For Example: If we want to rotate the below array by 2 positions.

1 2 3 4 5 6

$M = \text{GCD}(6, 2) = 2;$

Initial Array : 1 2 3 4 5 6

First Set Moves : 5 2 1 4 3 6

Second Set Moves : 5 6 1 2 3 4

Solution Code-

```
import java.util.Arrays;

import java.util.Scanner;

public class JugglingAlgorithm {

    private static int gcd(int a, int b) {

        while (b != 0) {

            int temp = b;

            b = a % b;

            a = temp;

        }

        return a;

    }

    public static int[] rotateRight(int[] arr, int k) {

        int n = arr.length;

        if (n == 0) return arr;

        // Handle negative rotations and normalize k

        k = k % n;

        if (k < 0) k += n;

        if (k == 0) return Arrays.copyOf(arr, n);

        int m = gcd(n, k);
```

```

int[] rotatedArray = Arrays.copyOf(arr, n);

for (int i = 0; i < m; i++) {
    int temp = rotatedArray[i];
    int j = i;

    while (true) {
        int nextJ = (j - k + n) % n; // Calculate next index with positive modulo

        if (nextJ == i) break;

        rotatedArray[j] = rotatedArray[nextJ];
        j = nextJ;
    }
    rotatedArray[j] = temp;
}

return rotatedArray;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter array elements: ");

```

```

int[] arr = Arrays.stream(scanner.nextLine().split(" "))
    .mapToInt(Integer::parseInt)
    .toArray();

System.out.print("Enter the number of rotations: ");

int k = scanner.nextInt();

int[] result = rotateRight(arr, k);

System.out.println("Rotated array: " + Arrays.toString(result));
}
}

```

Output-

Enter array elements: 1 2 3 4 5 6

Enter the number of rotations: 2

Rotated array: [5, 6, 1, 2, 3, 4]

2)Program to find the first non-repeating character in a string is discussed here. Given a string,

the task is to find the first non-repeating character in the string.

Input and Output format:

The first line of the input consists of a string.

Sample Input 1:

teeterson

Sample Output 1:

r

Sample Input 2:

charactercharacter

Sample Output 2:

All characters are repetitive

Algorithm to find the first non-repeating character in a string

Input the string from the user.

Start traversing the string using two loops.

Use the first loop to scan the characters of the string one by one.

Use the second loop to find if the current character is occurring in the latter part of the string or

not.

If it is not occurring, print that character.

Else, continue traversing.

Solution Code-

```
import java.util.Scanner;
```

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

```

public class FirstNonRepeatingChar {

    public static char check(String s) {

        for (int i = 0; i < s.length(); i++) {

            int count = 0;

            for (int j = 0; j < s.length(); j++) {

                if (s.charAt(j) == s.charAt(i)) {

                    count++;

                }

            }

            if (count == 1) {

                return s.charAt(i);

            }

        }

        System.out.println("All the characters are repetitive");

        return '0';

    }

}

```

```

public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the string: ");

    String message = scanner.nextLine();

    System.out.println(check(message));

}

```

```
        scanner.close();  
    }  
}
```

output -

Enter the string : teeterson

r

Enter the string : charactercharacter

All the characters are repetitive

0