21BAI1844
Shlok Kamath

Lab3

Shlok Kamath

21BAI1844

Cryptography and Network Security Lab (BCSE309P)

Professor: Dr. RAJESH R

Today's task:

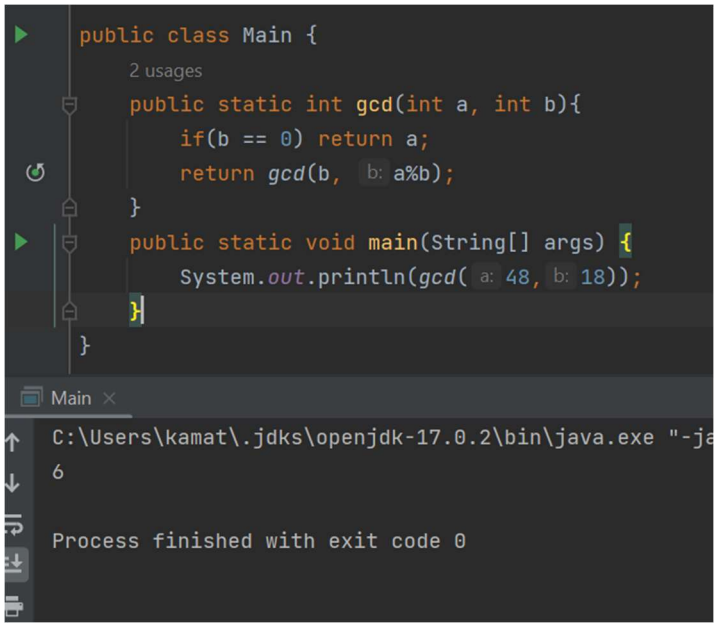
1. EUCLIDEAN ALGORITHM
2. RAILFENCE

EUCLIDEAN ALGORITHM:

Code:

```
public class Main {
    public static int gcd(int a, int b){
        if(b == 0) return a;
        return gcd(b, a%b);
    }
    public static void main(String[] args) {
        System.out.println(gcd(48,18));
    }
}
```

Output:



21BAI1844
Shlok Kamath

Railfence:

```
import java.util.*;
public class Main {
    public static void main(String[] args) {

        //Input
        String s = "JAISHRIRAM";
        int depth = 3;
        int n = s.length();
        System.out.println("Input String: "+s);
        //Encryption
        char[][] arr = new char[n][depth];
        int col=0;
        int row=0;
        int i=0;
        int neg = 0;
        while(i<n){
            arr[col][row] = s.charAt(i);
            col++;
            if(row == depth-1) neg = 1;
            else if(row == 0) neg = 0;
            if(neg == 1) row --;
            else row++;
            i++;
        }
        String res="";
        for(i=0; i<depth; i++){
            for(int j=0; j<n; j++){
                if(arr[j][i] != '\0'){
                    res+=arr[j][i];
                }
            }
        }
        System.out.println("Encoded String: "+res);

        //Decryption
        arr = new char[n][depth];
        i=0;
        col=0;
        row=0;
        while(i<n){
            arr[col][row] = '*';
            col++;
            if(row == depth-1) neg = 1;
            else if(row == 0) neg = 0;
            if(neg == 1) row --;
            else row++;
            i++;
        }
        int k=0;
        for(i=0; i<depth; i++){
            for(int j=0; j<n; j++){
                if(arr[j][i] == '*'){
                    arr[j][i] = res.charAt(k++);
                }
            }
        }
        String ans = "";
        i=0;
```

```
        col=0;
        row=0;
        while(i<n){
            ans += arr[col][row];
            col++;
            if(row == depth-1) neg = 1;
            else if(row == 0) neg = 0;
            if(neg == 1) row --;
            else row++;
            i++;
        }
        System.out.println("Decoded String: "+ans);
    }
}
```

Output: