Prime College

Khushibu ,Nayabazar

AFFILIATED TO TRIBHUWAN UNIVERSITY

Lab Report on
Computer Security and Cyber Law

Submitted By:

Aayush Dongol
BIM(6th semester)
Symbol No: [8206/17]

Submitted To:

BipinTimalsina Sir

TABLE OF CONTENT

Lab 1:

Write a program to implement Vigenere cipher.

Theory:

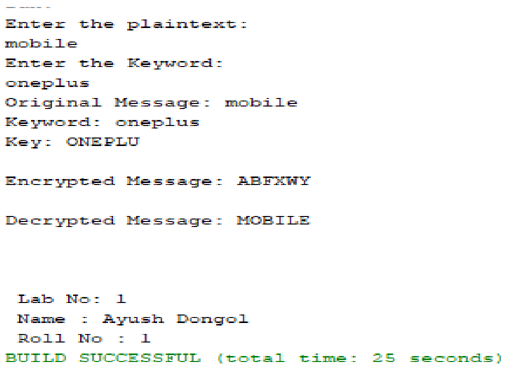
The vigenere cipher is an algorithm that is used to encrypting and decrypting the text. The vigenere cipher is an algorithm of encrypting an alphabetic text that uses a series of interwoven caesar ciphers. It is based on a keyword's letters. It is an example of a polyalphabetic substitution cipher. This algorithm is easy to understand and implement.

Code:

```
 import java.util.*;

 public class Vigenere {

                public static void main(String arg[]) {

                    Scanner s=new Scanner(System.in);

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

                    String plain=s.nextLine();

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

                    String keyword =s.nextLine();

                    encryptDecrypt(plain,keyword);

                 }


                 public static void encryptDecrypt(String plain, String keyword) {

                     String plaintext=plain.toUpperCase();

                     String Keyword=keyword.toUpperCase();

                     char msg[] = plaintext.toCharArray();

                     int msgLen = msg.length;

                     int i,j;


                     char key[] = new char[msgLen];

                     char encryptedMsg[] = new char[msgLen];

                     char decryptedMsg[] = new char[msgLen];

                      for(i = 0, j = 0; i < msgLen; ++i, ++j) {
```

```
         if(j == Keyword.length()) {
           j = 0;
}
         key[i] = Keyword.charAt(j);
      }
      //encryption code
      for(i = 0; i < msgLen; ++i)
        encryptedMsg[i] = (char) (((msg[i] + key[i]) % 26) + 'A');
      //decryption code
      for(i = 0; i < msgLen; ++i)
        decryptedMsg[i] = (char)((((encryptedMsg[i] - key[i]) + 26) % 26) + 'A');
      System.out.println("Original Message: " + plain);
      System.out.println("Keyword: " + keyword);

      System.out.println("Key: " + String.valueOf(key));
      System.out.println();
      System.out.println("Encrypted Message: " + String.valueOf(encryptedMsg));
      System.out.println();
      System.out.println("Decrypted Message: " + String.valueOf(decryptedMsg));
      System.out.println(" ");
      System.out.println(" " );
      System.out.println(" " );
      System.out.println(" Lab No: 1" );
      System.out.println(" Name : Ayush Dongol");
      System.out.println(" Roll No : 1 " );
    }
```

Output:



```
Enter the plaintext:
mobile
Enter the Keyword:
oneplus
Original Message: mobile
Keyword: oneplus
Key: ONEPLU

Encrypted Message: ABFXWY

Decrypted Message: MOBILE



 Lab No: 1
 Name : Ayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 25 seconds)
```

Lab2:

Write a program to implement Rail Fence Cipher.

Theory:

Given a plain-text message and a numeric key, cipher/de-cipher the given text using Rail Fence algorithm. The rail fence cipher (also called a zigzag cipher) is a form of transposition cipher. It derives its name from the way in which it is encoded.

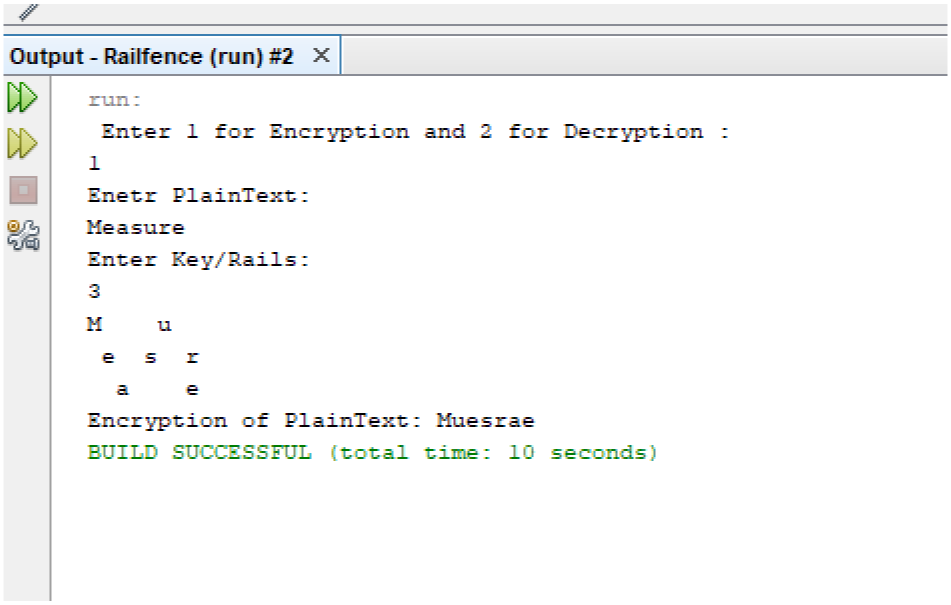
Code:

```
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;
```

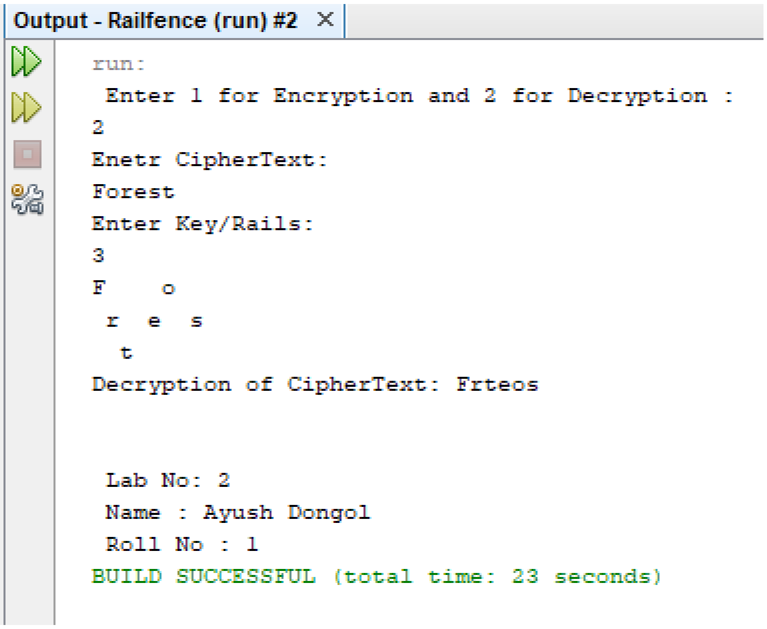
```
   int j=0;
   int row=key;
   int col=text.length();
   char[][]a= new char[row][col];
   for(int i=0; i< col;i++) {
      if(j == 0 || j == key - 1)
         check= !check;
         a[j][i]='*';
         if(check)
            j++;
         else
            j--;
   }
   int index=0;
   check =false;
   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 : Ayush Dongol");
      System.out.println(" Roll No : 1 " );
     }
    }
 }: 1 " );
    }
}
```

Output:



```
run:
 Enter 1 for Encryption and 2 for Decryption :
1
Enetr PlainText:
Measure
Enter Key/Rails:
3
M    u
 e  s  r
  a    e
Encryption of PlainText: Muesrae
BUILD SUCCESSFUL (total time: 10 seconds)
```



```
run:
 Enter 1 for Encryption and 2 for Decryption :
2
Enetr CipherText:
Forest
Enter Key/Rails:
3
F    o
 r  e  s
  t
Decryption of CipherText: Frteos


 Lab No: 2
 Name : Ayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 23 seconds)
```

Lab 3:

Write a program to find the GCD of the given two numbers using Euclidean Algorithm.

Theory:

The algorithm is based on the below facts.

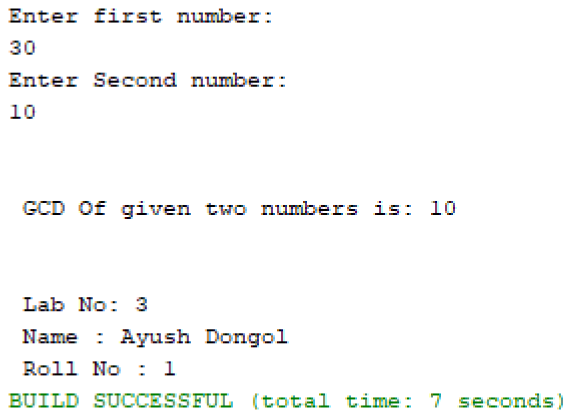
- If we subtract a smaller number from a larger (we reduce a larger number), GCD doesn’t change. So if we keep subtracting repeatedly the larger of two, we end up with GCD.
- Now instead of subtraction, if we divide the smaller number, the algorithm stops when we find remainder 0.

Code:

```
import java.util.*;

class gcd {

      public static void main(String[] args) {

        int gcd=0;

        Scanner scan= new Scanner(System.in);

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

        int a =scan.nextInt();

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

        int b=scan.nextInt();

        for( int i=1; i<=a &&  i<=b;i++) {

           if(a%i==0 && b%i==0){

               gcd=i;

           }

        }

        System.out.println(" ");

        System.out.println(" " );

        System.out.println(" GCD Of given two numbers is: "+gcd);

        System.out.println(" ");

        System.out.println(" " );

        System.out.println(" Lab No: 3" );

        System.out.println(" Name : Ayush Dongol");

        System.out.println(" Roll No : 1 " );
```

```
    }
  }
```

Output:



```
Enter first number:
30
Enter Second number:
10


 GCD Of given two numbers is: 10


 Lab No: 3
 Name : Ayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 7 seconds)
```

Lab 4:

Write a program to find the Addictive Inverse of the given number in given modulo.

Theory:

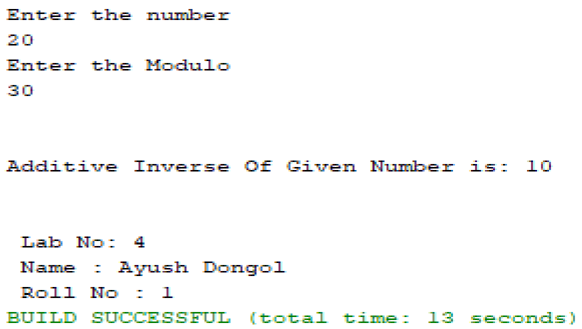
An additive inverse of a number is defined as the value, which on adding with the original number results in zero value. It is the value we add to a number to yield zero. Suppose, a is the original number, then its additive inverse will be minus of a i.e.,-a, such that;

a+(-a) = a – a = 0

Code:

```
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 : Ayush Dongol");

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

    }

  }
```

Output:



```
Enter the number
20
Enter the Modulo
30


Additive Inverse Of Given Number is: 10


 Lab No: 4
 Name : Ayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 13 seconds)
```

Lab 5:

Write a program to find the Multiplicative Inverse of a given number in the given modulo using Extended Euclidean Algorithm

Theory:

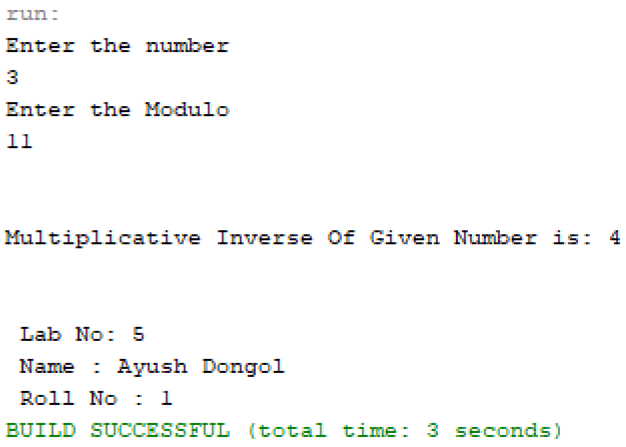
The multiplicative inverse of a number say, N is represented by 1/N or N^{-1}. It is also called reciprocal**,** derived from a Latin word 'reciprocus'. The meaning of inverse is something which is opposite. The reciprocal of a number obtained is such that when it is multiplied with the original number the value equals to identity 1. In other words, it is a method of dividing a number by its own to generate identity 1, such as N/N = 1.

Code:

```
import java.util.*;

public class multiplicativeinverse {

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

    number = number % modulo;

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

      if ((number * x) % modulo == 1){

        System.out.println("Multiplicative 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: 5" );

        System.out.println(" Name : Ayush Dongol");

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

    }

 }
```

Output:



```
run:
Enter the number
3
Enter the Modulo
11


Multiplicative Inverse Of Given Number is: 4


 Lab No: 5
 Name : Ayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 3 seconds)
```

Lab 6:

Write a program to check whether the given number two number are coprime or not.

Theory:

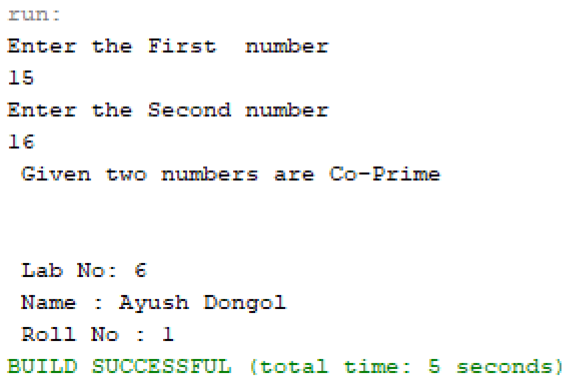
A Co-prime number is a set of numbers or integers which have only 1 as their common factor i.e. their highest common factor (HCF) will be 1. Co-prime numbers are also known as relatively prime or mutually prime numbers.

Code:

```
import java.util.*;

public class coprime {

    static int gcd(int a, int b) {


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

            return 0;a

        }

            // 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 : Ayush Dongol");
    System.out.println(" Roll No : 1 " );
  }
   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
15
Enter the Second number
16
 Given two numbers are Co-Prime


 Lab No: 6
 Name : Ayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 5 seconds)
```

Lab 7:

Write a program to find Totient value of the given number.

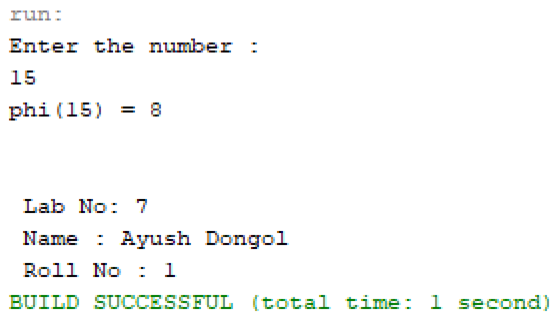
Theory:

The totient function $\phi(n)$, also called Euler's totient function, is defined as the number of positive integers $\le n$ that are relatively prime to (i.e., do not contain any factor in common with) n, where 1 is counted as being relatively prime to all numbers. Since a number less than or equal to and relatively prime to a given number is called a totative, the totient function $\phi(n)$ can be simply defined as the number of totatives of n.

Code:

```
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 : Ayush Dongol");

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

  }

 }
```

Output:



```
run:
Enter the number :
15
phi(15) = 8


 Lab No: 7
 Name : Ayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 1 second)
```

Lab 8:

Write a program to implement RSA algorithm.

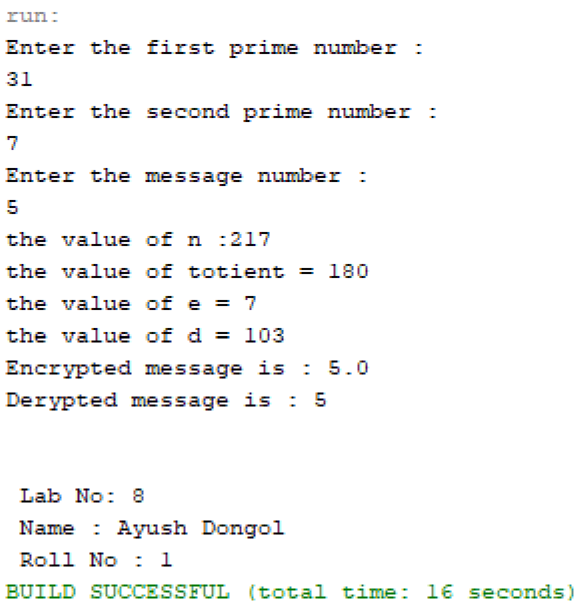
Theory:

RSA (Rivest–Shamir–Adleman) is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. Asymmetric means that there are two different keys. This is also called public key cryptography, because one of the keys can be given to anyone.

Code:

```
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 : Ayush Dongol");
    System.out.println(" Roll No : 1 " );
}
static int gcd(int e, int z)
{
    if (e == 0) {
        return z;
        }
        else {
            return gcd(z % e, e);
        }
    }
}
```

Output:



```
run:
Enter the first prime number :
31
Enter the second prime number :
7
Enter the message number :
5
the value of n :217
the value of totient = 180
the value of e = 7
the value of d = 103
Encrypted message is : 5.0
Derypted message is : 5


 Lab No: 8
 Name : Ayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 16 seconds)
```

Lab 9:

Write a program that acts as a malicious code.

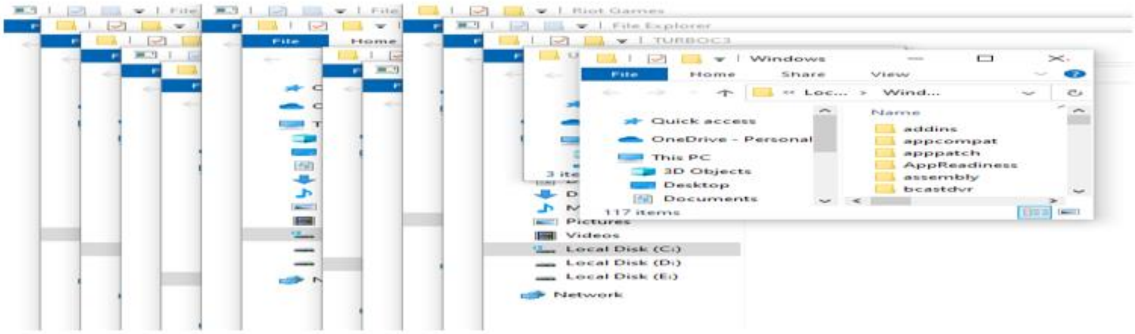
Theory:

Malicious code is unwanted files or programs that can cause harm to a computer or compromise data stored on a computer. Various classifications of malicious code include viruses, worms, and Trojan horses.

Code:

```
package computersecuritylabh;

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 : Ayush Dongol");
        System.out.println(" Roll No : 1 " );
        }
       }
      }
     }
   }
   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:

An Encryption Key is secret value, which is used as an input given by user to the Encryption algorithm along with the plain Text and plain text is converted to Cipher-Text by alphabet shift (move of letters further in the alphabet) with help of given key. Decryption is the Process of converting back the Cipher-Text to original plaintext with an inverse alphabet shift with the help of same given key by that user.

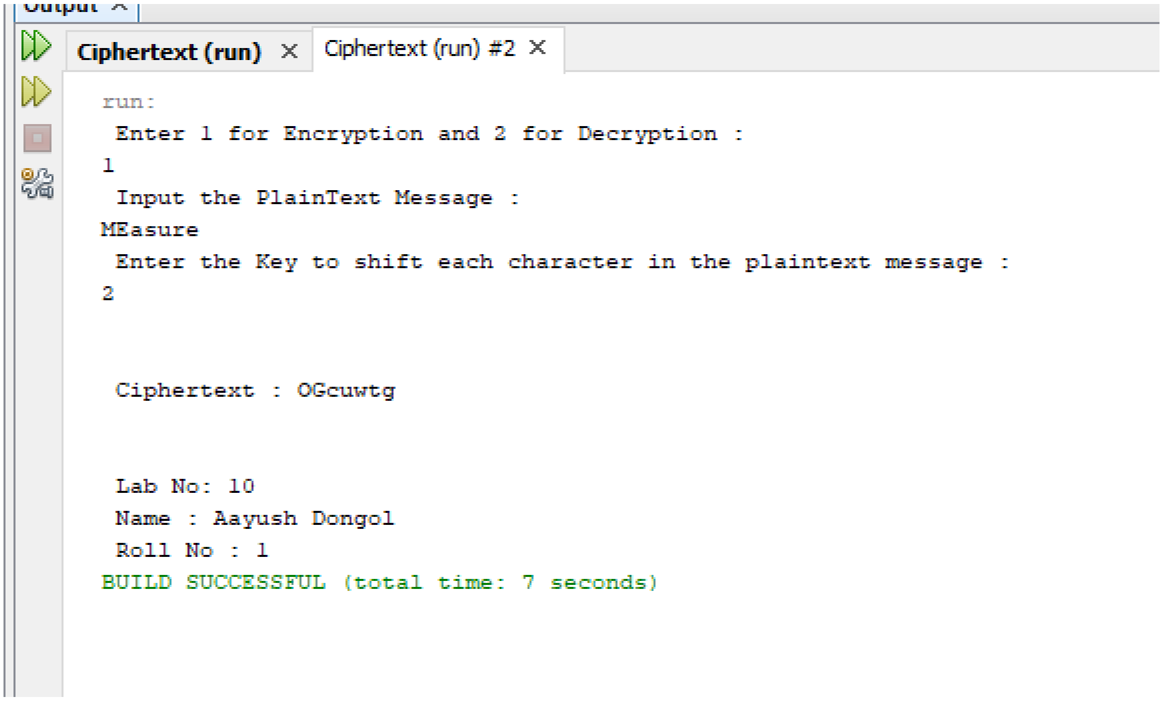
Code:

```
import java.util.Scanner;

public class Ciphertext  {

  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;

            }
```

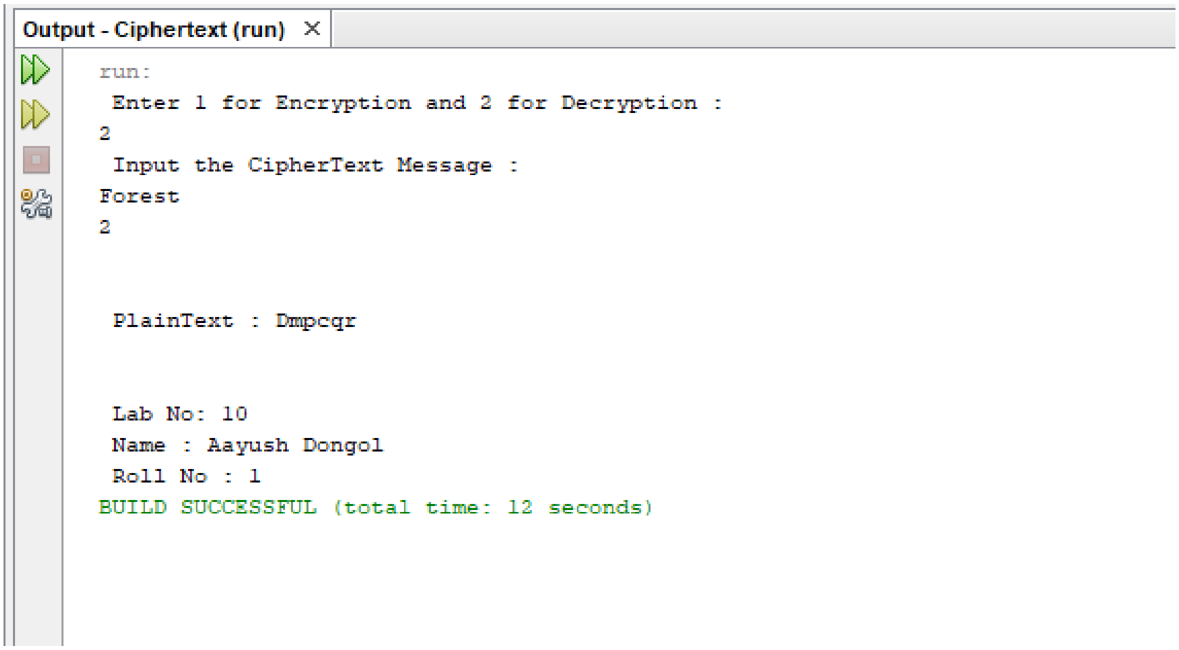
```
    else if(alphabet >= 'A' && alphabet <= 'Z' )  {
      alphabet = (char) (alphabet + shift);
      if(alphabet > 'Z' )  {
         alphabet = (char) (alphabet+'A'-'Z'-1);
       }
       ciphertext = ciphertext + alphabet;
     }
     else {
      ciphertext = ciphertext + alphabet;
    }
  }
  System.out.println(" ");
  System.out.println(" " );
  System.out.println(" Ciphertext : " + ciphertext);
  System.out.println(" ");
  System.out.println(" " );
  System.out.println(" Lab No: 1" );
  System.out.println(" Name : Ayush Agarwal");
  System.out.println(" Roll No : 5 " );
}
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 : Aayush Dongol");
         System.out.println(" Roll No : 1" );
       }
   }
}
```

Output:



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


 Ciphertext : OGcuwtg


 Lab No: 10
 Name : Aayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 7 seconds)
```



```
run:
 Enter 1 for Encryption and 2 for Decryption :
2
 Input the CipherText Message :
Forest
2


 PlainText : Dmpcqr


 Lab No: 10
 Name : Aayush Dongol
 Roll No : 1
BUILD SUCCESSFUL (total time: 12 seconds)
```