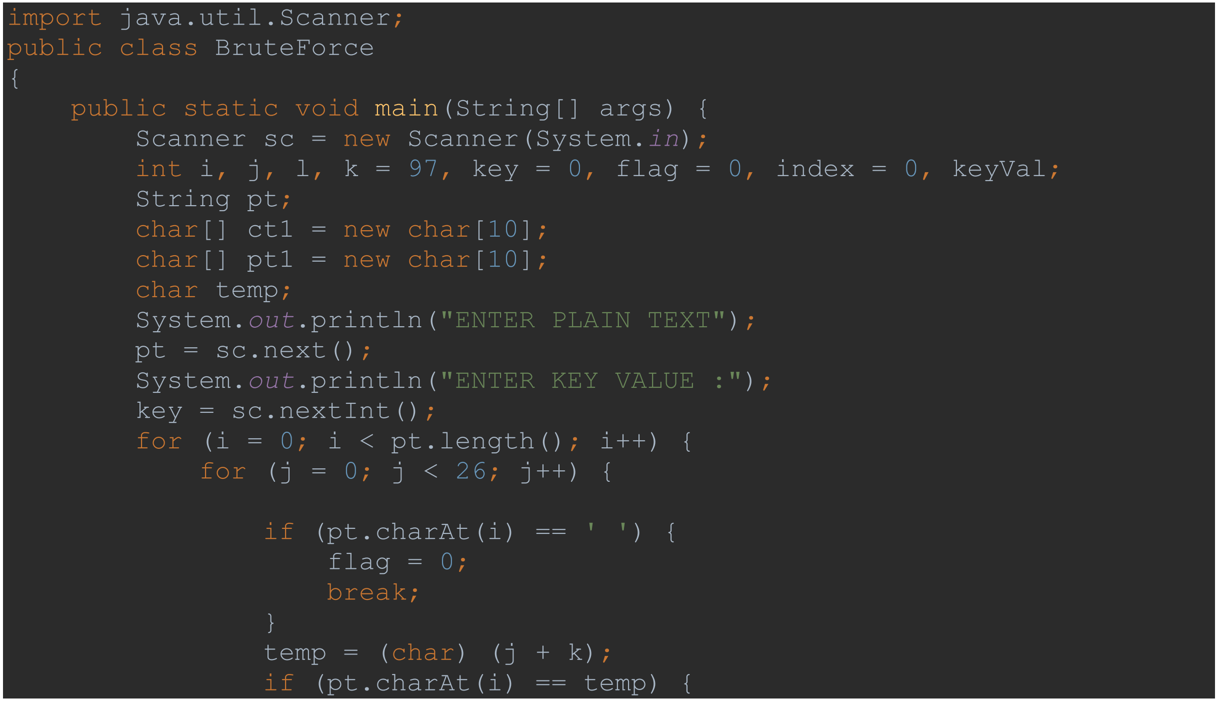
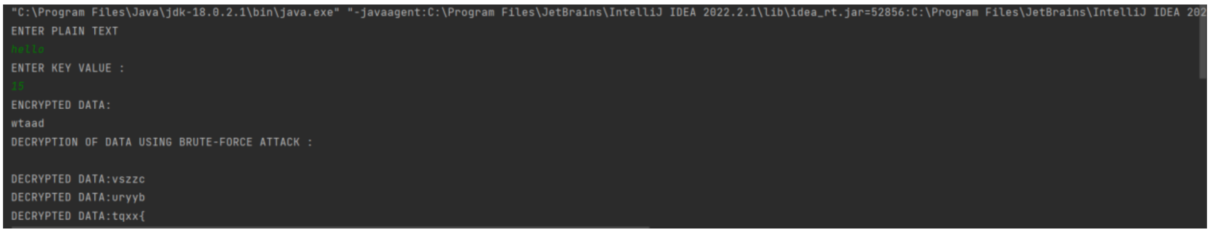
OBJECTIVE (Title 1) Write a program to implement Additive Cipher (Z26) with the following conditions: Plaintext should be in lowercase. Ciphertext should be in uppercase. Brute force attack.

CODE –

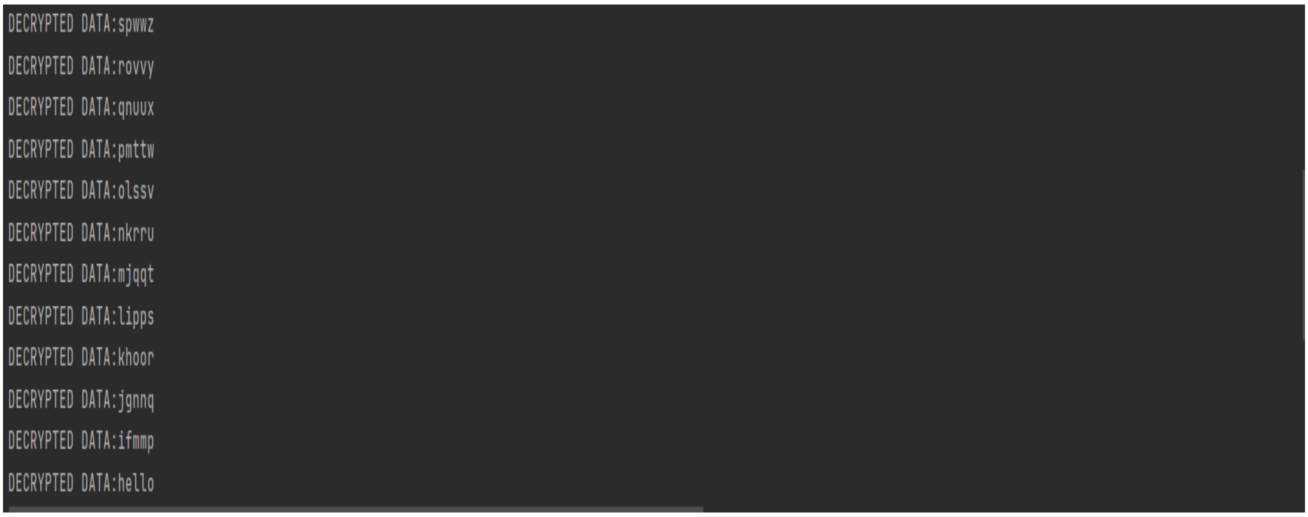


```java
import java.util.Scanner;
public class BruteForce
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int i, j, l, k = 97, key = 0, flag = 0, index = 0, keyVal;
        String pt;
        char[] ct1 = new char[10];
        char[] pt1 = new char[10];
        char temp;
        System.out.println("ENTER PLAIN TEXT");
        pt = sc.next();
        System.out.println("ENTER KEY VALUE :");
        key = sc.nextInt();
        for (i = 0; i < pt.length(); i++) {
            for (j = 0; j < 26; j++) {

                if (pt.charAt(i) == ' ') {
                    flag = 0;
                    break;
                }
                temp = (char) (j + k);
                if (pt.charAt(i) == temp) {
                    flag = 1;
                    index = j;
                    break;
                }
            }
            if (flag == 1) {

                char c = (char) (((index + key) % 26) + 97);
                ct1[i] = c;

            }
        }
        System.out.println("ENCRYPTED DATA:");
        for (i = 0; i < pt.length(); i++) {
            System.out.print(ct1[i]);
        }
        System.out.println("\n" + "DECRYPTION OF DATA USING BRUTE-FORCE
ATTACK :");
        key = 1;
        while (key <= 26) {
            for (i = 0; i < pt.length(); i++) {
                for (j = 0; j < 26; j++) {
                    if (ct1[i] == ' ') {
                        flag = 0;
                        break;
                    }
                    temp = (char) (j + k);
                    if (ct1[i] == temp) {
                        flag = 1;
                        index = j;
                        break;
                    }
                }
                keyVal = index - key;
                if (flag == 1 & keyVal > 0) {
                    pt1[i] = (char) ((keyVal % 26) + 97);

                } else if (flag == 1) {
                    pt1[i] = (char) ((26 + keyVal) + 97);
                }

            }
            System.out.print("\n" + "DECRYPTED DATA:");
            for (i = 0; i < pt.length(); i++) {
                System.out.print(pt1[i]);
            }
            key++;
        }
    }
}
```

Output –



```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.2.1\lib\idea_rt.jar=52856:C:\Program Files\JetBrains\IntelliJ IDEA 202
ENTER PLAIN TEXT
hello
ENTER KEY VALUE :
15
ENCRYPTED DATA:
wtaad
DECRYPTION OF DATA USING BRUTE-FORCE ATTACK :

DECRYPTED DATA:vszzc
DECRYPTED DATA:uryyb
DECRYPTED DATA:tqxx{
```

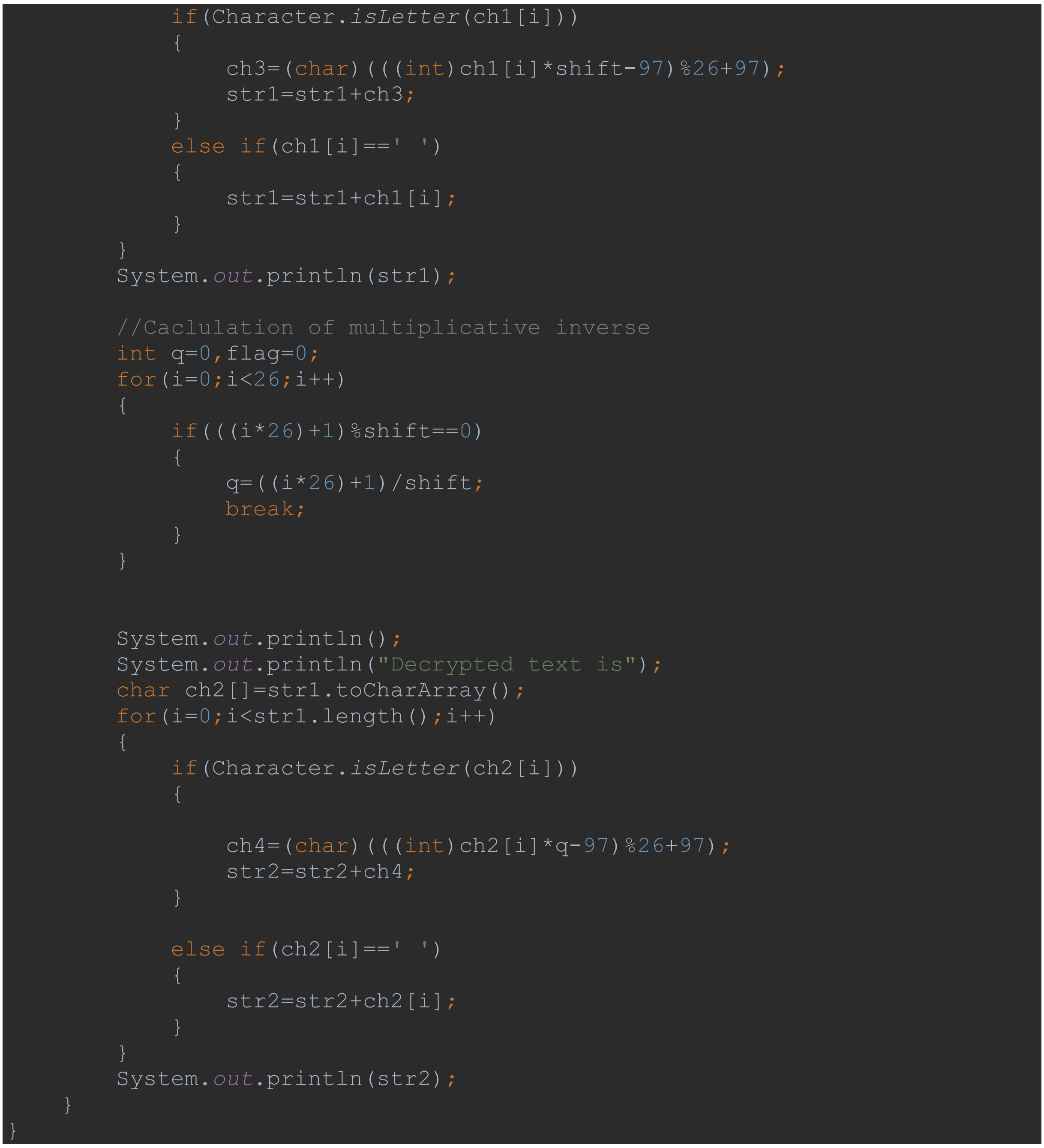


```
DECRYPTED DATA:spwwz

DECRYPTED DATA:rovvy

DECRYPTED DATA:qnuux

DECRYPTED DATA:pmttw

DECRYPTED DATA:olssv

DECRYPTED DATA:nkrru

DECRYPTED DATA:mjqqt

DECRYPTED DATA:lipps

DECRYPTED DATA:khoor

DECRYPTED DATA:jgnnq

DECRYPTED DATA:ifmmp

DECRYPTED DATA:hello
```

OBJECTIVE - Title: 2. Write a program to implement Multiplicative Cipher. Plaintext should be in lowercase. Ciphertext should be uppercase. Brute force attack.

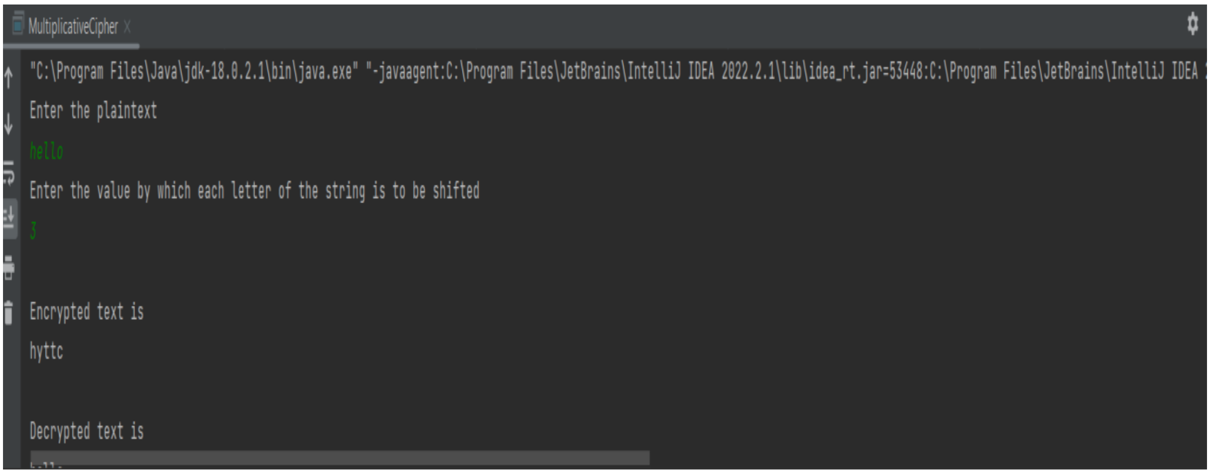
CODE –

```
import java.util.*;

class MultiplicativeCipher
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int shift,i,n;
        String str;
        String str1="";
        String str2="";
        System.out.println("Enter the plaintext");
        str=sc.nextLine();
        str=str.toLowerCase();
        n=str.length();
        char ch1[]=str.toCharArray();
        char ch3;
        char ch4;
        System.out.println("Enter the value by which each letter of the
string is to be shifted");
        shift=sc.nextInt();

        System.out.println();
        System.out.println("Encrypted text is");

        for(i=0;i<n;i++)
        {
```



```
            if(Character.isLetter(ch1[i]))
            {
                ch3=(char)(((int)ch1[i]*shift-97)%26+97);
                str1=str1+ch3;
            }
            else if(ch1[i]==' ')
            {
                str1=str1+ch1[i];
            }
        }
        System.out.println(str1);

        //Caclulation of multiplicative inverse
        int q=0,flag=0;
        for(i=0;i<26;i++)
        {
            if(((i*26)+1)%shift==0)
            {
                q=((i*26)+1)/shift;
                break;
            }
        }


        System.out.println();
        System.out.println("Decrypted text is");
        char ch2[]=str1.toCharArray();
        for(i=0;i<str1.length();i++)
        {
            if(Character.isLetter(ch2[i]))
            {

                ch4=(char)(((int)ch2[i]*q-97)%26+97);
                str2=str2+ch4;
            }

            else if(ch2[i]==' ')
            {
                str2=str2+ch2[i];
            }
        }
        System.out.println(str2);
    }
}
```

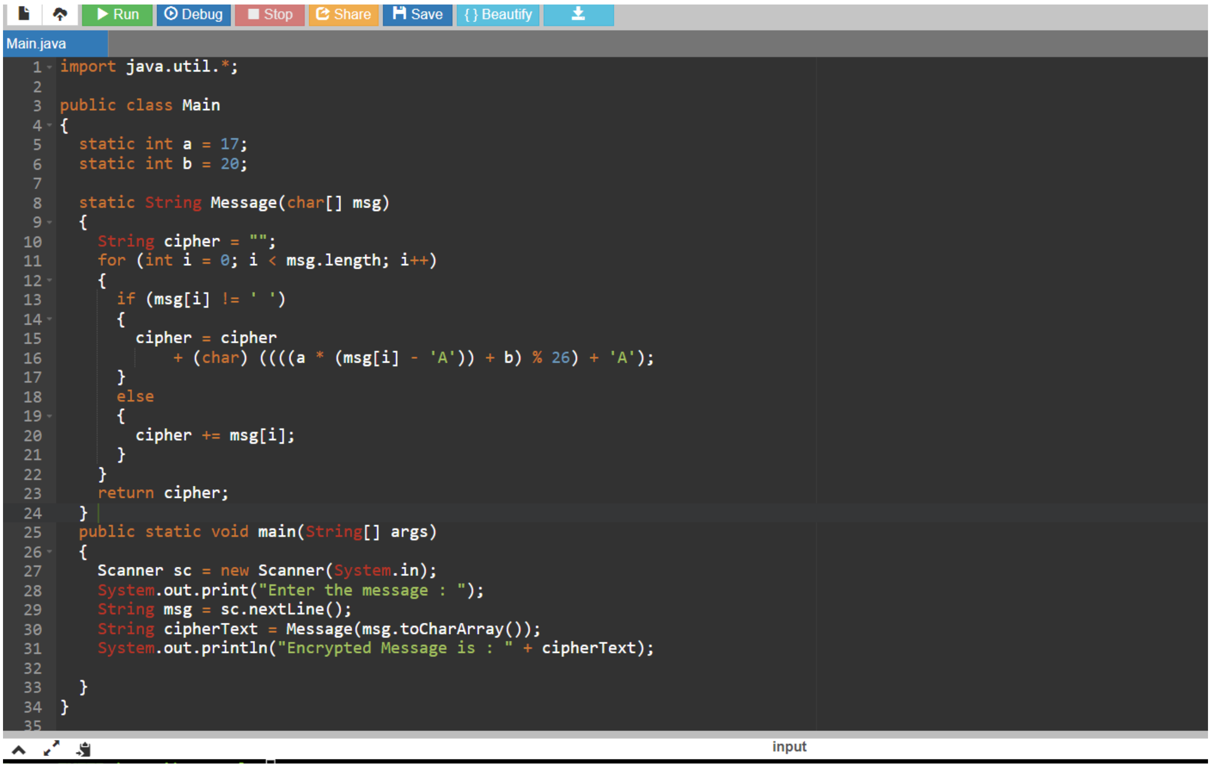
OUTPUT –



```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.2.1\lib\idea_rt.jar=53448:C:\Program Files\JetBrains\IntelliJ IDEA
Enter the plaintext
hello
Enter the value by which each letter of the string is to be shifted
3


Encrypted text is
hyttc


Decrypted text is
```

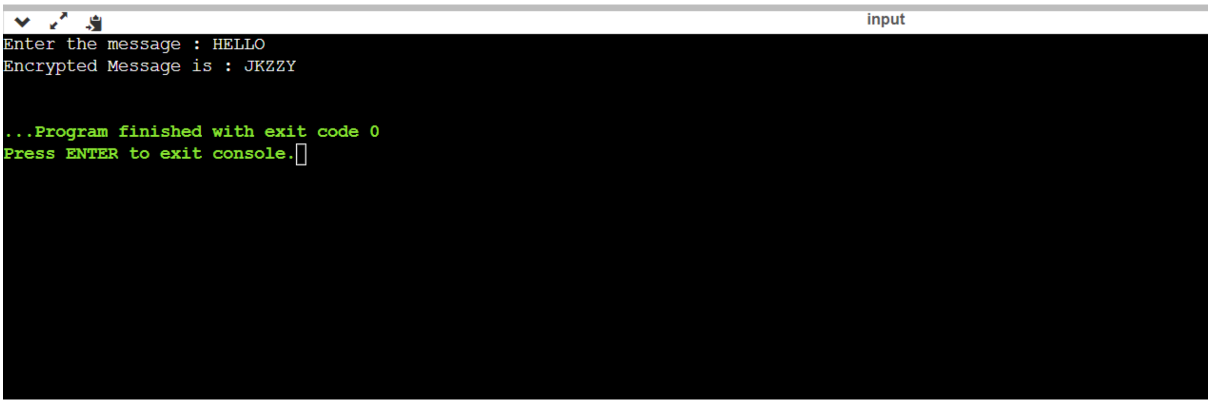
TITLE 3 : Write a program to implement Affine Cipher. Plaintext should be in lowercase. Ciphertext should be uppercase. Brute force attack.

Encryption Program:



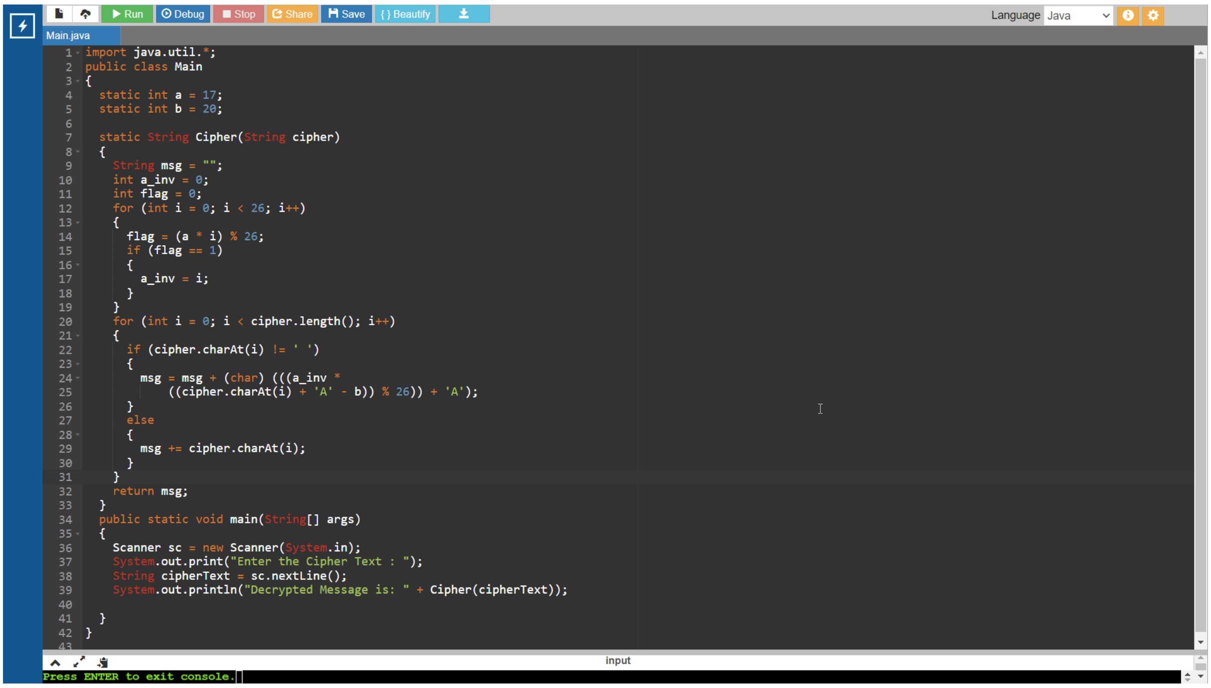
```java
import java.util.*;

public class Main
{
  static int a = 17;
  static int b = 20;

  static String Message(char[] msg)
  {
    String cipher = "";
    for (int i = 0; i < msg.length; i++)
    {
      if (msg[i] != ' ')
      {
        cipher = cipher
            + (char) ((((a * (msg[i] - 'A')) + b) % 26) + 'A');
      }
      else
      {
        cipher += msg[i];
      }
    }
    return cipher;
  }
  public static void main(String[] args)
  {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the message : ");
    String msg = sc.nextLine();
    String cipherText = Message(msg.toCharArray());
    System.out.println("Encrypted Message is : " + cipherText);

  }
}
```

Encryption Output:



```
Enter the message : HELLO
Encrypted Message is : JKZZY


...Program finished with exit code 0
Press ENTER to exit console.
```

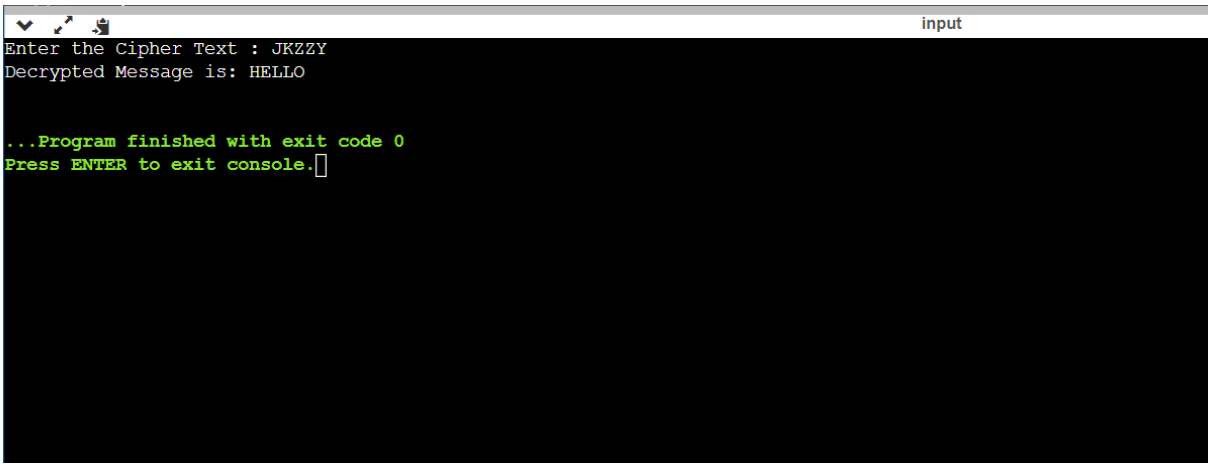
Decryption Program:



```
import java.util.*;
public class Main
{
  static int a = 17;
  static int b = 20;

  static String Cipher(String cipher)
  {
    String msg = "";
    int a_inv = 0;
    int flag = 0;
    for (int i = 0; i < 26; i++)
    {
      flag = (a * i) % 26;
      if (flag == 1)
      {
        a_inv = i;
      }
    }
    for (int i = 0; i < cipher.length(); i++)
    {
      if (cipher.charAt(i) != ' ')
      {
        msg = msg + (char) (((a_inv *
            ((cipher.charAt(i) + 'A' - b)) % 26)) + 'A');
      }
      else
      {
        msg += cipher.charAt(i);
      }
    }
    return msg;
  }
  public static void main(String[] args)
  {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the Cipher Text : ");
    String cipherText = sc.nextLine();
    System.out.println("Decrypted Message is: " + Cipher(cipherText));

  }
}
```

```
Press ENTER to exit console.
```

Decrypted Output:

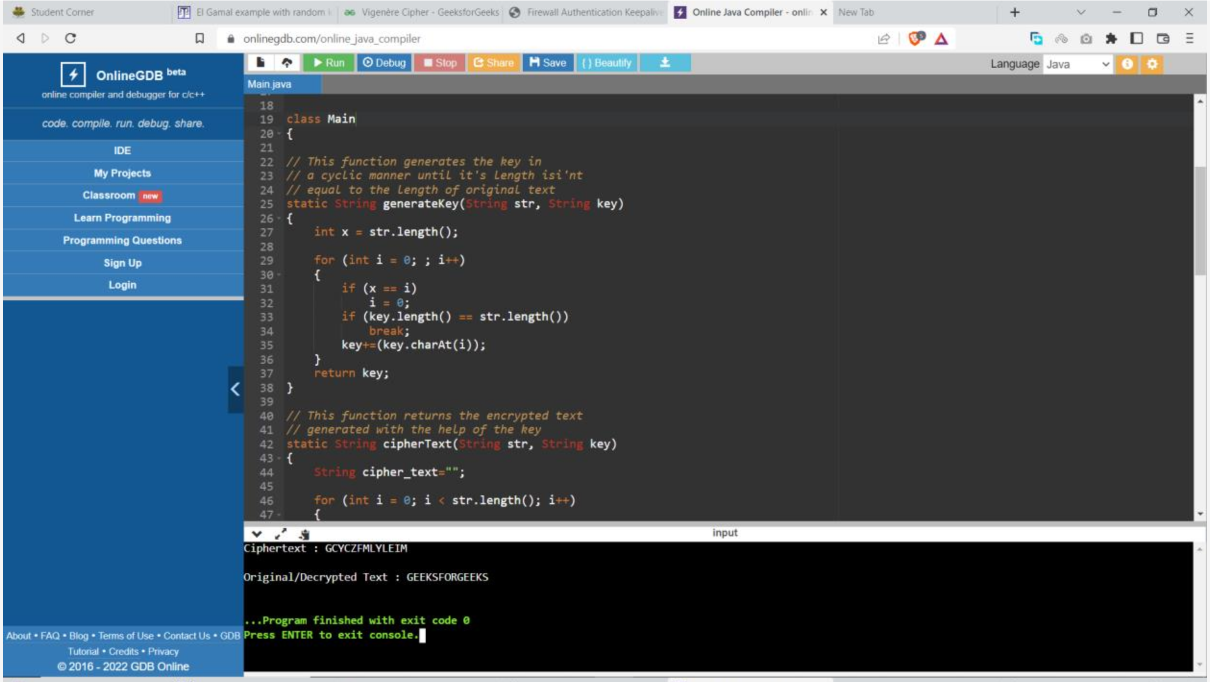


```
Enter the Cipher Text : JKZZY
Decrypted Message is: HELLO


...Program finished with exit code 0
Press ENTER to exit console.
```

Name-Nikhil Srivastava

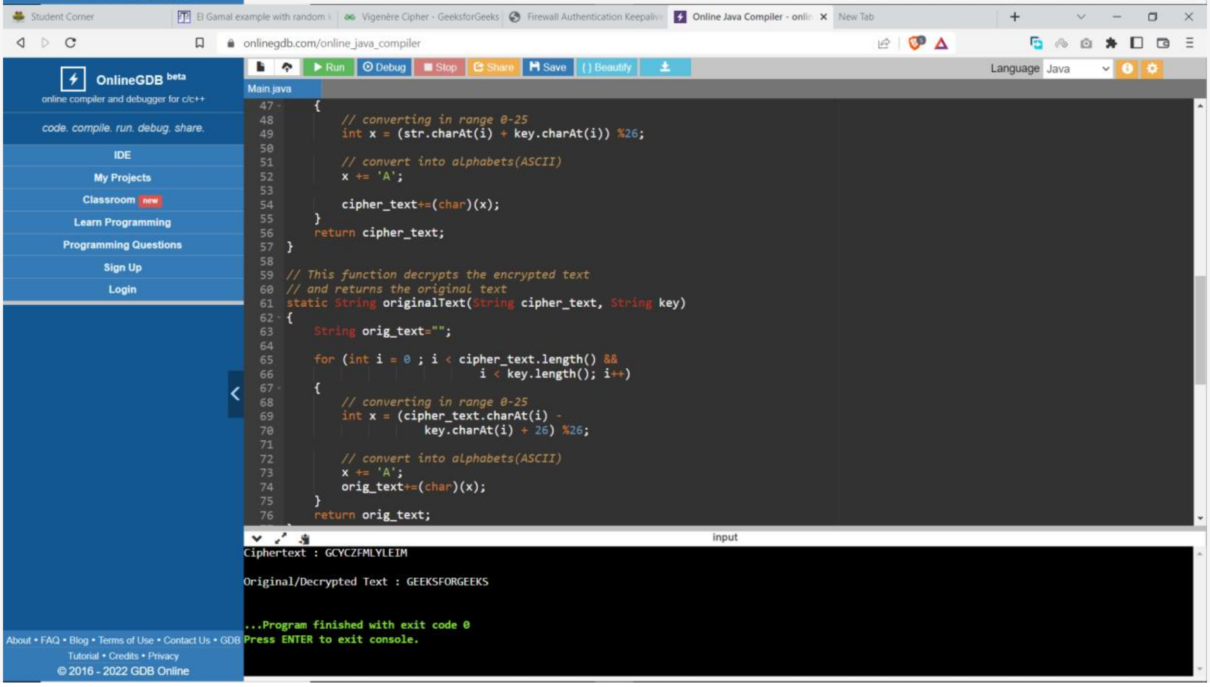
Sec-c

U roll num—191500505


Student Corner
El Gamal example with random
Vigenère Cipher - GeeksforGeeks
Firewall Authentication Keepalive
Online Java Compiler - online
New Tab
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

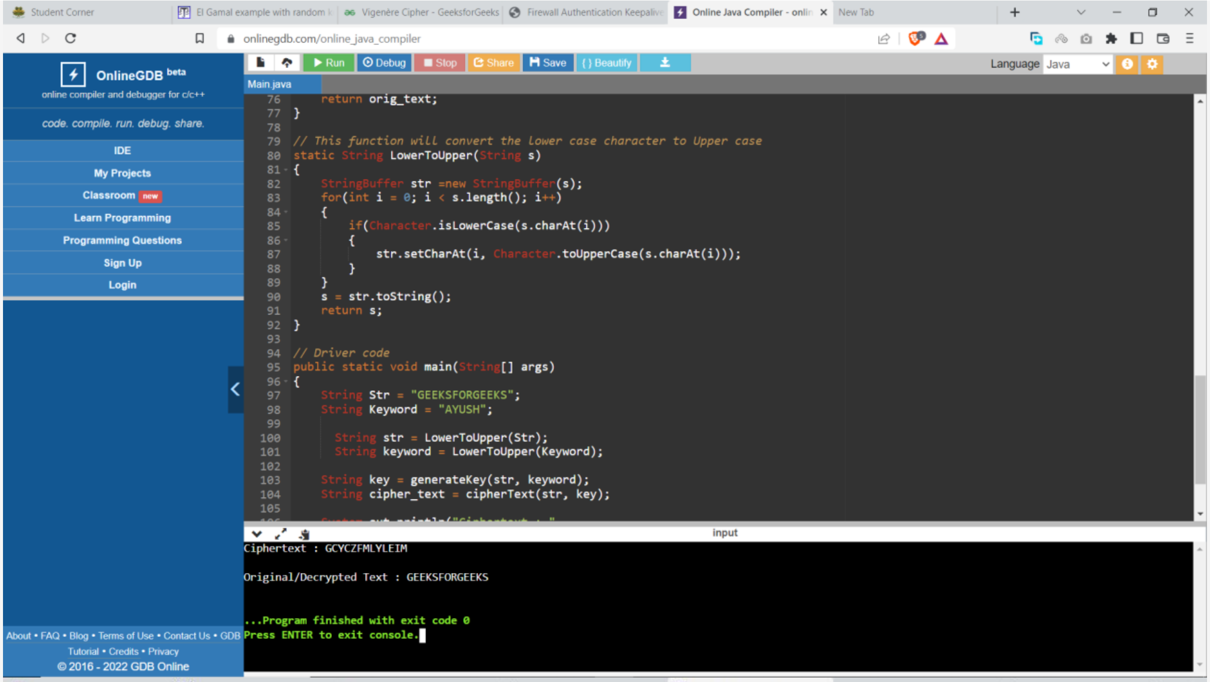
```java

class Main
{

// This function generates the key in
// a cyclic manner until it's length isi'nt
// equal to the length of original text
static String generateKey(String str, String key)
{
    int x = str.length();

    for (int i = 0; ; i++)
    {
        if (x == i)
            i = 0;
        if (key.length() == str.length())
            break;
        key+=(key.charAt(i));
    }
    return key;
}

// This function returns the encrypted text
// generated with the help of the key
static String cipherText(String str, String key)
{
    String cipher_text="";

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

```
Ciphertext : GCYCZFMLYLEIM

Original/Decrypted Text : GEEKSFORGEEKS


...Program finished with exit code 0
Press ENTER to exit console.
```


Student Corner
El Gamal example with random
Vigenère Cipher - GeeksforGeeks
Firewall Authentication Keepalive
Online Java Compiler - online
New Tab
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

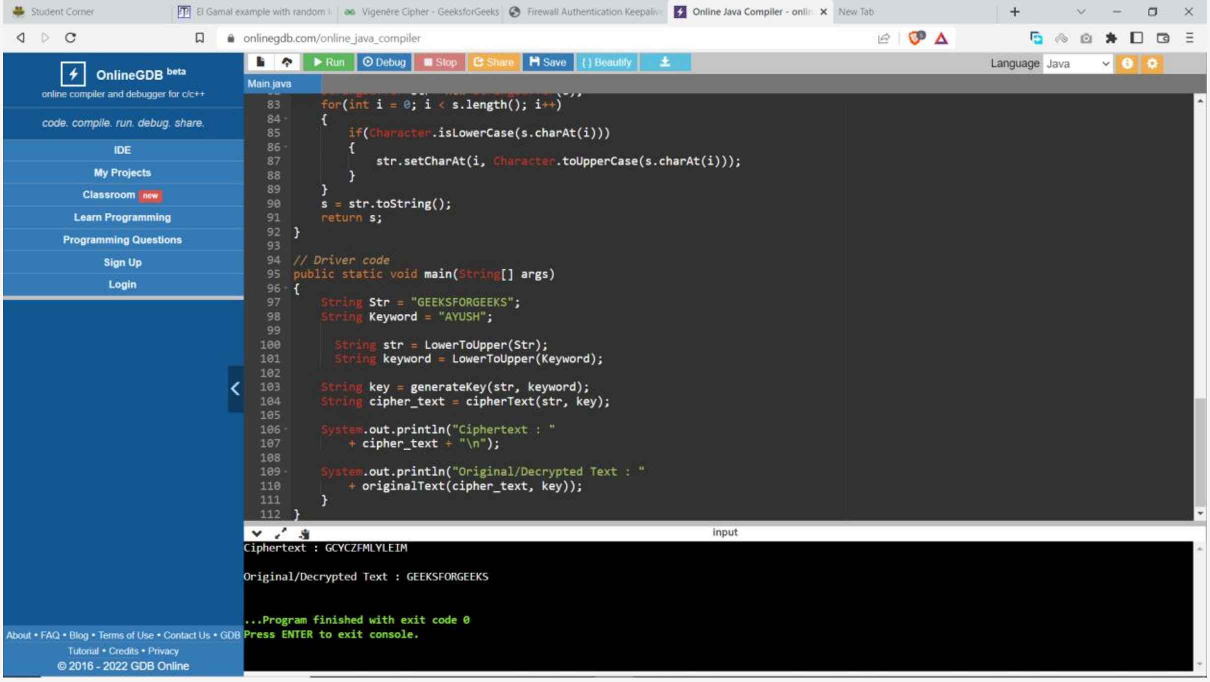
```java
    {
        // converting in range 0-25
        int x = (str.charAt(i) + key.charAt(i)) %26;

        // convert into alphabets(ASCII)
        x += 'A';

        cipher_text+=(char)(x);
    }
    return cipher_text;
}

// This function decrypts the encrypted text
// and returns the original text
static String originalText(String cipher_text, String key)
{
    String orig_text="";

    for (int i = 0 ; i < cipher_text.length() &&
                            i < key.length(); i++)
    {
        // converting in range 0-25
        int x = (cipher_text.charAt(i) -
                    key.charAt(i) + 26) %26;

        // convert into alphabets(ASCII)
        x += 'A';
        orig_text+=(char)(x);
    }
    return orig_text;
```

```
Ciphertext : GCYCZFMLYLEIM

Original/Decrypted Text : GEEKSFORGEEKS


...Program finished with exit code 0
Press ENTER to exit console.
```


Student Corner
El Gamal example with random
Vigenère Cipher - GeeksforGeeks
Firewall Authentication Keepalive
Online Java Compiler - online
New Tab
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

```
    return orig_text;
}

// This function will convert the lower case character to Upper case
static String LowerToUpper(String s)
{
    StringBuffer str =new StringBuffer(s);
    for(int i = 0; i < s.length(); i++)
    {
        if(Character.isLowerCase(s.charAt(i)))
        {
            str.setCharAt(i, Character.toUpperCase(s.charAt(i)));
        }
    }
    s = str.toString();
    return s;
}

// Driver code
public static void main(String[] args)
{
    String Str = "GEEKSFORGEEKS";
    String Keyword = "AYUSH";

      String str = LowerToUpper(Str);
      String keyword = LowerToUpper(Keyword);

    String key = generateKey(str, keyword);
    String cipher_text = cipherText(str, key);

```

```
Ciphertext : GCYCZFMLYLEIM

Original/Decrypted Text : GEEKSFORGEEKS


...Program finished with exit code 0
Press ENTER to exit console.
```

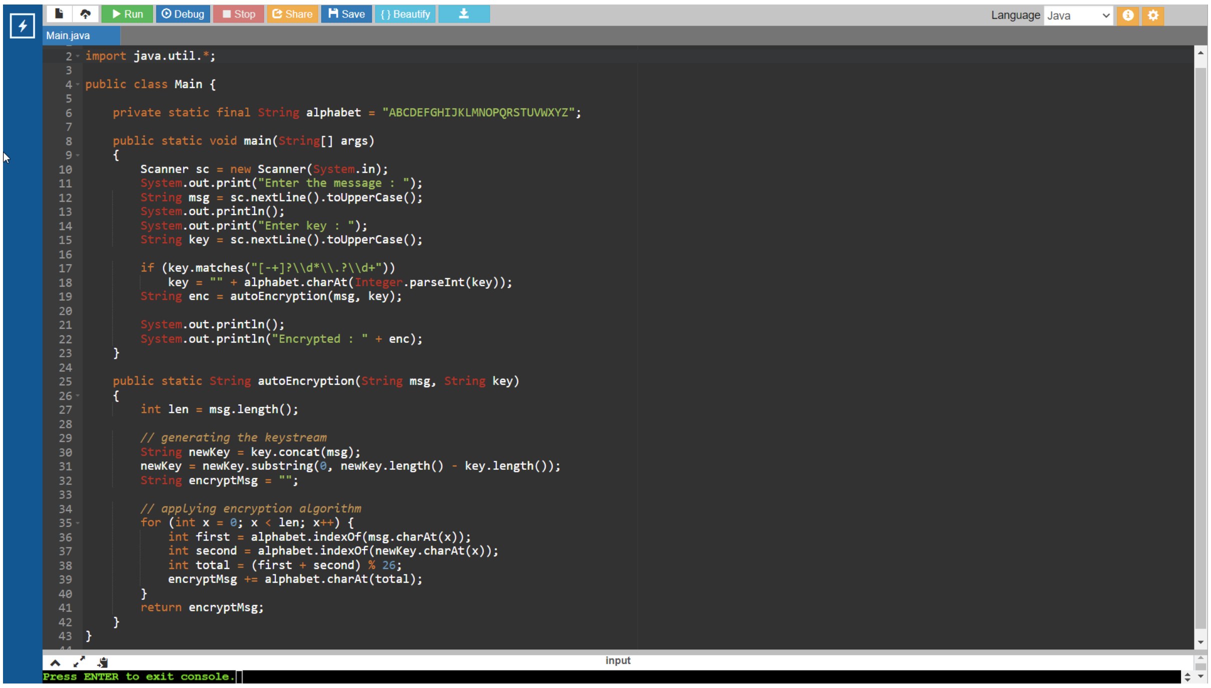

Student Corner
El Gamal example with random
Vigenère Cipher - GeeksforGeeks
Firewall Authentication Keepalive
Online Java Compiler - online
New Tab
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

```
    for(int i = 0; i < s.length(); i++)
    {
        if(Character.isLowerCase(s.charAt(i)))
        {
            str.setCharAt(i, Character.toUpperCase(s.charAt(i)));
        }
    }
    s = str.toString();
    return s;
}

// Driver code
public static void main(String[] args)
{
    String Str = "GEEKSFORGEEKS";
    String Keyword = "AYUSH";

      String str = LowerToUpper(Str);
      String keyword = LowerToUpper(Keyword);

    String key = generateKey(str, keyword);
    String cipher_text = cipherText(str, key);

    System.out.println("Ciphertext : "
        + cipher_text + "\n");

    System.out.println("Original/Decrypted Text : "
        + originalText(cipher_text, key));
    }
}
```

```
Ciphertext : GCYCZFMLYLEIM

Original/Decrypted Text : GEEKSFORGEEKS


...Program finished with exit code 0
Press ENTER to exit console.
```

TITLE 5 : Write a program to implement Autokey Cipher. Plaintext should be in lowercase. Ciphertext should be uppercase. Brute force attack.

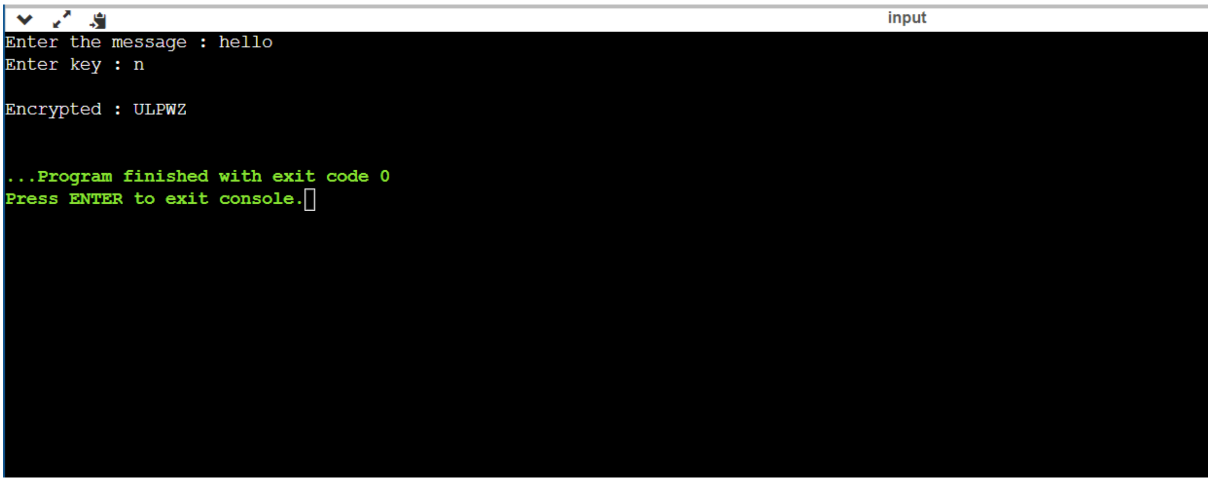
Encryption Program:



```java
import java.util.*;

public class Main {

    private static final String alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the message : ");
        String msg = sc.nextLine().toUpperCase();
        System.out.println();
        System.out.print("Enter key : ");
        String key = sc.nextLine().toUpperCase();

        if (key.matches("[-+]?\\d*\\.?\\d+"))
            key = "" + alphabet.charAt(Integer.parseInt(key));
        String enc = autoEncryption(msg, key);

        System.out.println();
        System.out.println("Encrypted : " + enc);
    }

    public static String autoEncryption(String msg, String key)
    {
        int len = msg.length();

        // generating the keystream
        String newKey = key.concat(msg);
        newKey = newKey.substring(0, newKey.length() - key.length());
        String encryptMsg = "";

        // applying encryption algorithm
        for (int x = 0; x < len; x++) {
            int first = alphabet.indexOf(msg.charAt(x));
            int second = alphabet.indexOf(newKey.charAt(x));
            int total = (first + second) % 26;
            encryptMsg += alphabet.charAt(total);
        }
        return encryptMsg;
    }
}
```

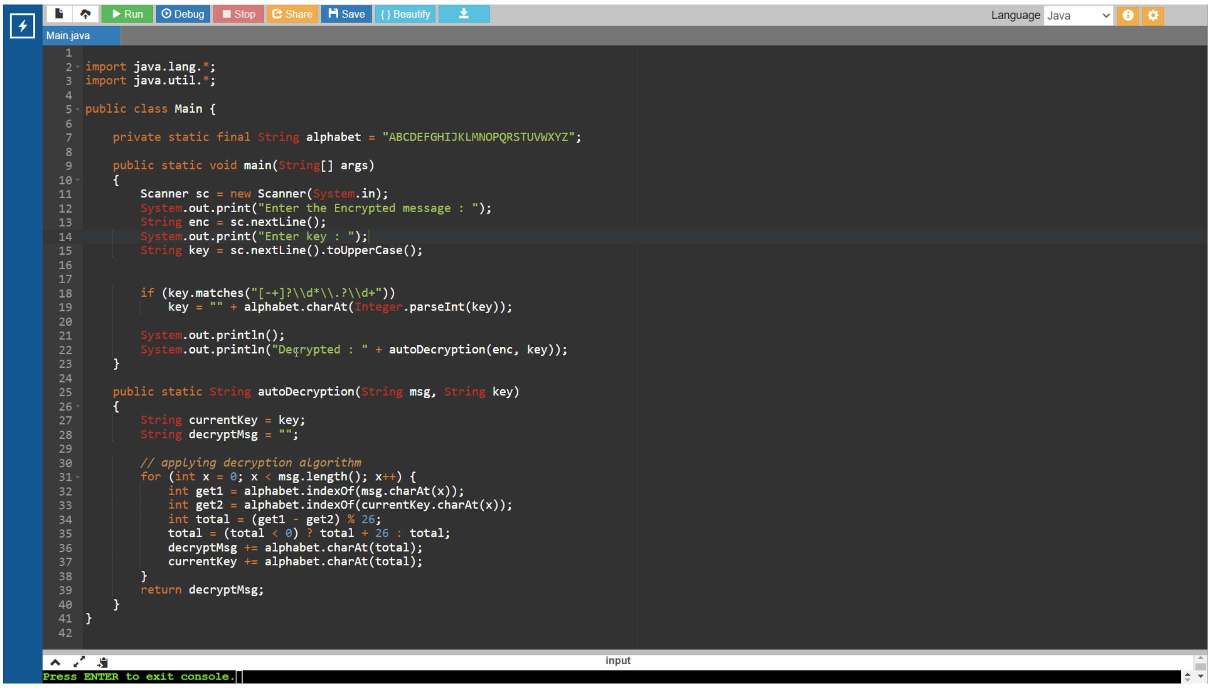
```
Press ENTER to exit console.
```

Encryption Output:



```
Enter the message : hello
Enter key : n

Encrypted : ULPWZ


...Program finished with exit code 0
Press ENTER to exit console.
```

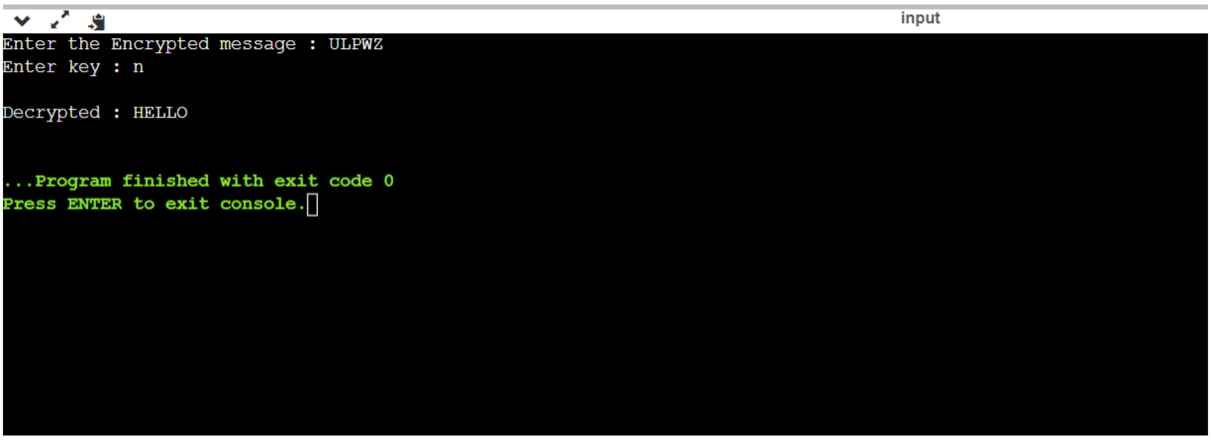
Decryption Program:



```java

import java.lang.*;
import java.util.*;

public class Main {

    private static final String alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Encrypted message : ");
        String enc = sc.nextLine();
        System.out.print("Enter key : ");
        String key = sc.nextLine().toUpperCase();


        if (key.matches("[-+]?\\d*\\.?\\d+"))
            key = "" + alphabet.charAt(Integer.parseInt(key));

        System.out.println();
        System.out.println("Decrypted : " + autoDecryption(enc, key));
    }

    public static String autoDecryption(String msg, String key)
    {
        String currentKey = key;
        String decryptMsg = "";

        // applying decryption algorithm
        for (int x = 0; x < msg.length(); x++) {
            int get1 = alphabet.indexOf(msg.charAt(x));
            int get2 = alphabet.indexOf(currentKey.charAt(x));
            int total = (get1 - get2) % 26;
            total = (total < 0) ? total + 26 : total;
            decryptMsg += alphabet.charAt(total);
            currentKey += alphabet.charAt(total);
        }
        return decryptMsg;
    }
}

```

```
Press ENTER to exit console.
```

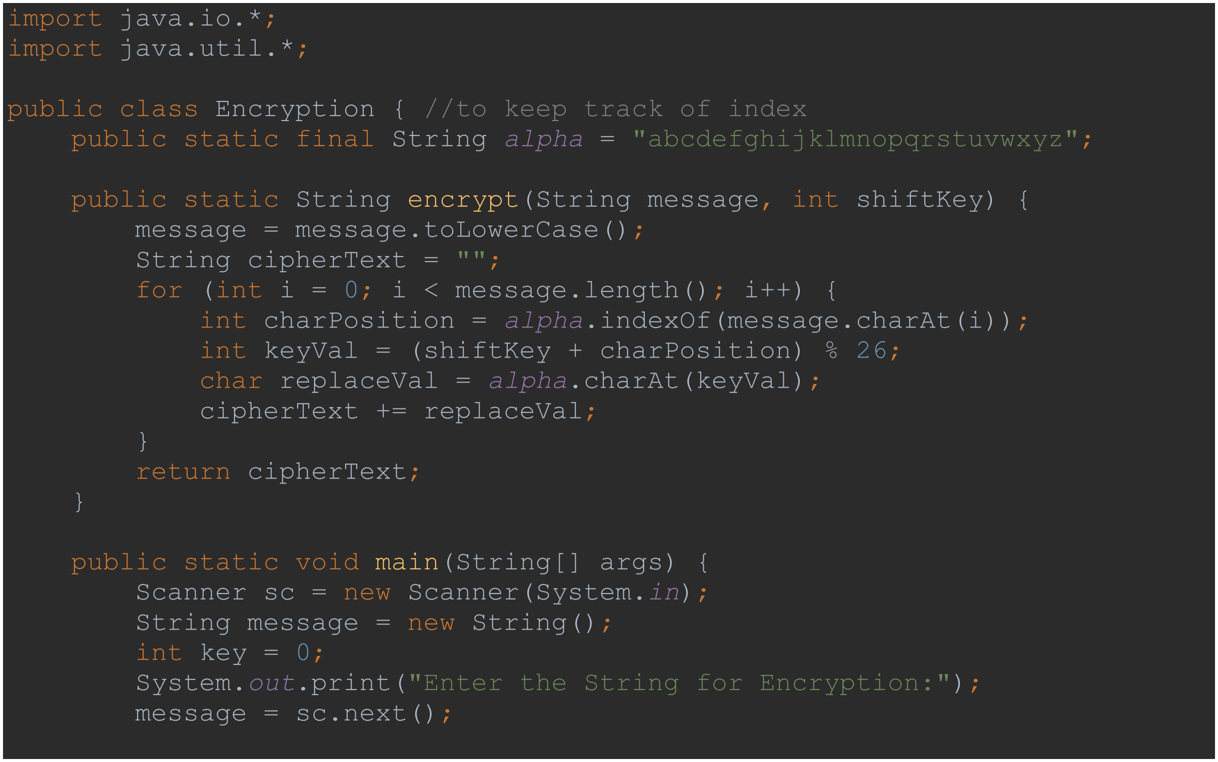
Decrypted Output:



```
Enter the Encrypted message : ULPWZ
Enter key : n

Decrypted : HELLO


...Program finished with exit code 0
Press ENTER to exit console.
```

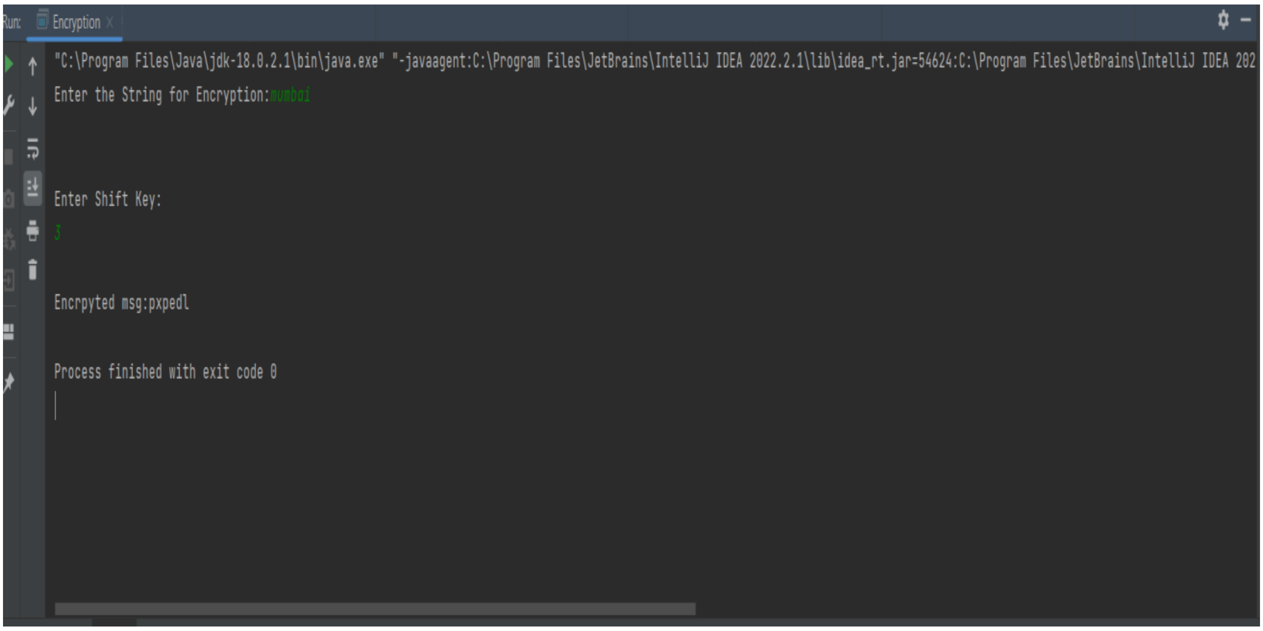
OBJECTIVE (Title 6) - Write a program to implement Playfair Cipher to encrypt & decrypt the given message where the key matrix can be formed by using a given keyword.

CODE: - Encryption using shift key



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

public class Encryption { //to keep track of index
    public static final String alpha = "abcdefghijklmnopqrstuvwxyz";

    public static String encrypt(String message, int shiftKey) {
        message = message.toLowerCase();
        String cipherText = "";
        for (int i = 0; i < message.length(); i++) {
            int charPosition = alpha.indexOf(message.charAt(i));
            int keyVal = (shiftKey + charPosition) % 26;
            char replaceVal = alpha.charAt(keyVal);
            cipherText += replaceVal;
        }
        return cipherText;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String message = new String();
        int key = 0;
        System.out.print("Enter the String for Encryption:");
        message = sc.next();
        System.out.println("\n\nEnter Shift Key:");
        key = sc.nextInt();
        System.out.println("\nEncrpyted msg:" + encrypt(message, key));
    } //main method ends
} //Main Class End
```

Output: -

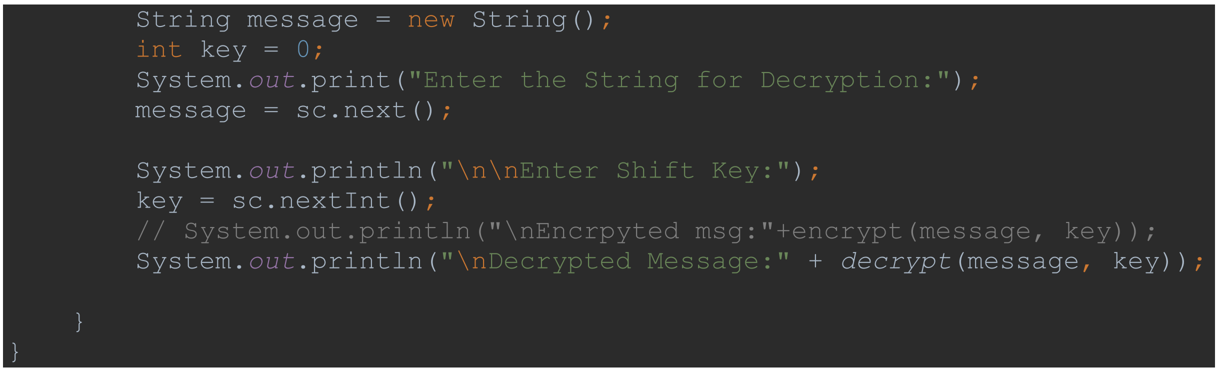


```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.2.1\lib\idea_rt.jar=54624:C:\Program Files\JetBrains\IntelliJ IDEA 202
Enter the String for Encryption:mumbai


Enter Shift Key:
3


Encrpyted msg:pxpedl


Process finished with exit code 0
```

Decryption using shift key: -

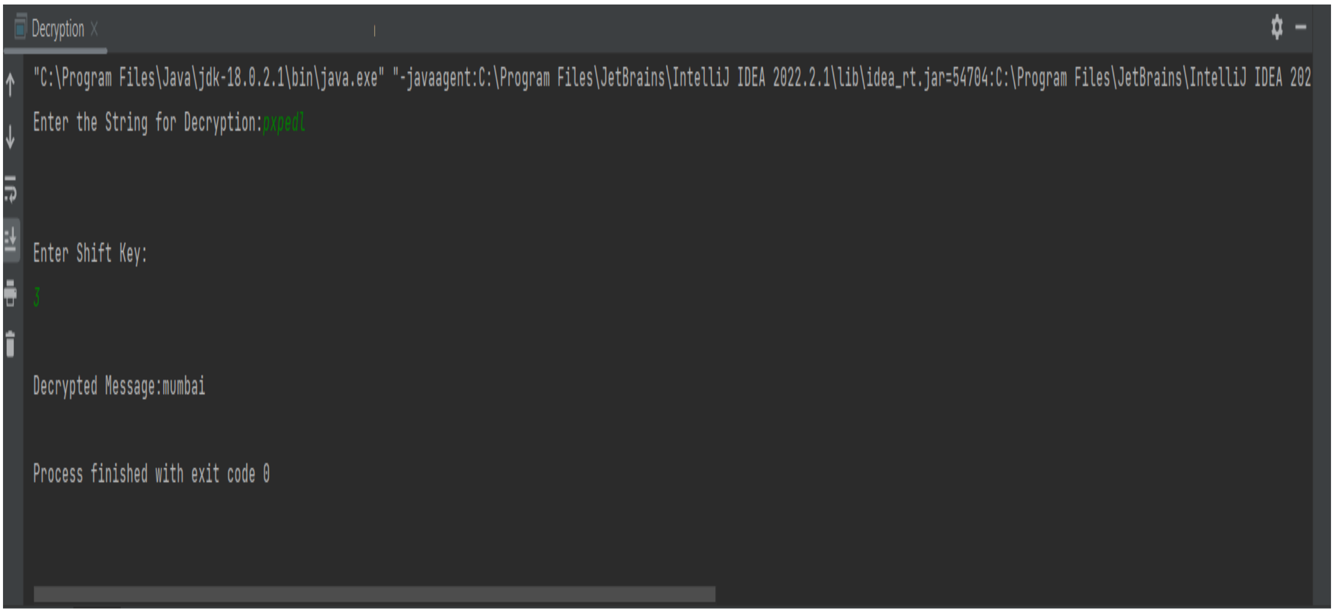
CODE –

```
import java.io.*;
import java.util.*;
public class Decryption { //to keep track of index
    public static final String ALPHABET = "abcdefghijklmnopqrstuvwxyz";

    public static String decrypt(String cipherText, int shiftKey) {
        cipherText = cipherText.toLowerCase();
        String message = "";
        for (int i = 0; i < cipherText.length(); i++) {
            int charPosition = ALPHABET.indexOf(cipherText.charAt(i));
            int keyVal = (charPosition - shiftKey) % 26;
            if (keyVal < 0) {
                keyVal = ALPHABET.length() + keyVal;
            }
            char replaceVal = ALPHABET.charAt(keyVal);
            message += replaceVal;
        }
        return message;
    }

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



```
        String message = new String();
        int key = 0;
        System.out.print("Enter the String for Decryption:");
        message = sc.next();

        System.out.println("\n\nEnter Shift Key:");
        key = sc.nextInt();
        // System.out.println("\nEncrpyted msg:"+encrypt(message, key));
        System.out.println("\nDecrypted Message:" + decrypt(message, key));

    }
}
```

Output –



```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.2.1\lib\idea_rt.jar=54704:C:\Program Files\JetBrains\IntelliJ IDEA 202
Enter the String for Decryption:pxpedl



Enter Shift Key:
3


Decrypted Message:mumbai


Process finished with exit code 0
```

Name-Nikhil Srivastava

Sec-c

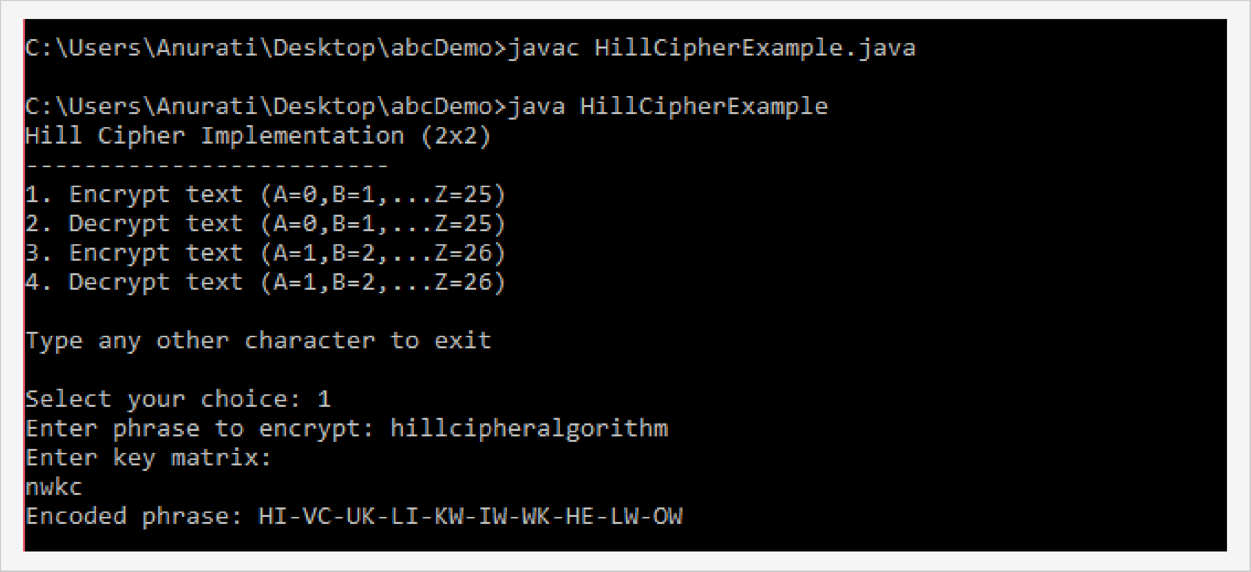
U roll num—191500505

```java
import java.util.ArrayList;
import java.util.Scanner;
public class HillCipherExample {
    //method to accept key matrix
    private static int[][] getKeyMatrix() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter key matrix:");
        String key = sc.nextLine();
        //int len = key.length();
        double sq = Math.sqrt(key.length());
        if (sq != (long) sq) {
            System.out.println("Cannot Form a square matrix");
        }
        int len = (int) sq;
        int[][] keyMatrix = new int[len][len];
        int k = 0;
        for (int i = 0; i < len; i++)
        {
            for (int j = 0; j < len; j++)
            {
                keyMatrix[i][j] = ((int) key.charAt(k)) - 97;
                k++;
            }
        }
        return keyMatrix;
    }
    // Below method checks whether the key matrix is valid (det=0)
    private static void isValidMatrix(int[][] keyMatrix) {
        int det = keyMatrix[0][0] * keyMatrix[1][1] - keyMatrix[0][1] * keyMatrix[1][0];
        // If det=0, throw exception and terminate
        if(det == 0) {
            throw new java.lang.Error("Det equals to zero, invalid key matrix!");
        }
    }
    // This method checks if the reverse key matrix is valid (matrix mod26 = (1,0,0,1)
        private static void isValidReverseMatrix(int[][] keyMatrix, int[][] reverseMatrix) {
        int[][] product = new int[2][2];
        // Find the product matrix of key matrix times reverse key matrix
        product[0][0] = (keyMatrix[0][0]*reverseMatrix[0][0] + keyMatrix[0][1] * reverseMatrix[1][0]) % 26;
        product[0][1] = (keyMatrix[0][0]*reverseMatrix[0][1] + keyMatrix[0][1] * reverseMatrix[1][1]) % 26;
        product[1][0] = (keyMatrix[1][0]*reverseMatrix[0][0] + keyMatrix[1][1] * reverseMatrix[1][0]) % 26;
        product[1][1] = (keyMatrix[1][0]*reverseMatrix[0][1] + keyMatrix[1][1] * reverseMatrix[1][1]) % 26;
        // Check if a=1 and b=0 and c=0 and d=1
        // If not, throw exception and terminate
        if(product[0][0] != 1 || product[0][1] != 0 || product[1][0] != 0 || product[1][1] != 1) {
            throw new java.lang.Error("Invalid reverse matrix found!");
        }
    }
    // This method calculates the reverse key matrix
    private static int[][] reverseMatrix(int[][] keyMatrix) {
        int detmod26 = (keyMatrix[0][0] * keyMatrix[1][1] - keyMatrix[0][1] * keyMatrix[1][0]) % 26; // Calc det
        int factor;
        int[][] reverseMatrix = new int[2][2];
        // Find the factor for which is true that
        // factor*det = 1 mod 26
        for(factor=1; factor < 26; factor++)
        {
            if((detmod26 * factor) % 26 == 1)
            {
                break;
            }
        }
        // Calculate the reverse key matrix elements using the factor found
        reverseMatrix[0][0] = keyMatrix[1][1]         * factor % 26;
        reverseMatrix[0][1] = (26 - keyMatrix[0][1])   * factor % 26;
        reverseMatrix[1][0] = (26 - keyMatrix[1][0])   * factor % 26;
        reverseMatrix[1][1] = keyMatrix[0][0]         * factor % 26;
        return reverseMatrix;
    }
    // This method echoes the result of encrypt/decrypt
    private static void echoResult(String label, int adder, ArrayList<Integer> phrase) {
        int i;
        System.out.print(label);
        // Loop for each pair
        for(i=0; i < phrase.size(); i += 2) {
            System.out.print(Character.toChars(phrase.get(i) + (64 + adder)));
            System.out.print(Character.toChars(phrase.get(i+1) + (64 + adder)));
            if(i+2 <phrase.size()) {
                System.out.print("-");
            }
        }
        System.out.println();
    }
    // This method makes the actual encryption
    public static void encrypt(String phrase, boolean alphaZero)
    {
        int i;
        int adder = alphaZero ? 1 : 0; // For calclulations depending on the alphabet
```

```java
        int[][] keyMatrix;
        ArrayList<Integer> phraseToNum = new ArrayList<>();
        ArrayList<Integer> phraseEncoded = new ArrayList<>();
        // Delete all non-english characters, and convert phrase to upper case
        phrase = phrase.replaceAll("[^a-zA-Z]","").toUpperCase();

        // If phrase length is not an even number, add "Q" to make it even
        if(phrase.length() % 2 == 1) {
            phrase += "Q";
        }
        // Get the 2x2 key matrix from sc
                   keyMatrix = getKeyMatrix();
                   // Check if the matrix is valid (det != 0)
                   isValidMatrix(keyMatrix);
                   // Convert characters to numbers according to their
                   // place in ASCII table minus 64 positions (A=65 in ASCII table)
                   // If we use A=0 alphabet, subtract one more (adder)
                   for(i=0; i < phrase.length(); i++) {
                       phraseToNum.add(phrase.charAt(i) - (64 + adder));
                   }
                   // Find the product per pair of the phrase with the key matrix modulo 26
                   // If we use A=1 alphabet and result is 0, replace it with 26 (Z)
                   for(i=0; i < phraseToNum.size(); i += 2) {
                       int x = (keyMatrix[0][0] * phraseToNum.get(i) + keyMatrix[0][1] * phraseToNum.get(i+1)) % 26;
                       int y = (keyMatrix[1][0] * phraseToNum.get(i) + keyMatrix[1][1] * phraseToNum.get(i+1)) % 26;
                       phraseEncoded.add(alphaZero ? x : (x == 0 ? 26 : x ));
                       phraseEncoded.add(alphaZero ? y : (y == 0 ? 26 : y ));
                   }
                   // Print the result
                   echoResult("Encoded phrase: ", adder, phraseEncoded);
               }
               // This method makes the actual decryption
               public static void decrypt(String phrase, boolean alphaZero)
               {
```

```
        int i, adder = alphaZero ? 1 : 0;
        int[][] keyMatrix, revKeyMatrix;
        ArrayList<Integer> phraseToNum = new ArrayList<>();
        ArrayList<Integer> phraseDecoded = new ArrayList<>();
        // Delete all non-english characters, and convert phrase to upper case
        phrase = phrase.replaceAll("[^a-zA-Z]","").toUpperCase();

        // Get the 2x2 key matrix from sc
        keyMatrix = getKeyMatrix();
        // Check if the matrix is valid (det != 0)
        isValidMatrix(keyMatrix);
        // Convert numbers to characters according to their
        // place in ASCII table minus 64 positions (A=65 in ASCII table)
        // If we use A=0 alphabet, subtract one more (adder)
        for(i=0; i < phrase.length(); i++) {
            phraseToNum.add(phrase.charAt(i) - (64 + adder));
        }
        // Find the reverse key matrix
        revKeyMatrix = reverseMatrix(keyMatrix);
        // Check if the reverse key matrix is valid (product = 1,0,0,1)
        isValidReverseMatrix(keyMatrix, revKeyMatrix);
        // Find the product per pair of the phrase with the reverse key matrix modulo 26
        for(i=0; i < phraseToNum.size(); i += 2) {
            phraseDecoded.add((revKeyMatrix[0][0] * phraseToNum.get(i) + revKeyMatrix[0][1] * phraseToNum.get(i+1)) % 26);
            phraseDecoded.add((revKeyMatrix[1][0] * phraseToNum.get(i) + revKeyMatrix[1][1] * phraseToNum.get(i+1)) % 26);
        }
        // Print the result
        echoResult("Decoded phrase: ", adder, phraseDecoded);
    }
    //main method
    public static void main(String[] args) {
        String opt, phrase;
        byte[] p;
```

```
                Scanner sc = new Scanner(System.in);
                System.out.println("Hill Cipher Implementation (2x2)");
                System.out.println("-------------------------");
                System.out.println("1. Encrypt text (A=0,B=1,...Z=25)");
                System.out.println("2. Decrypt text (A=0,B=1,...Z=25)");
                System.out.println("3. Encrypt text (A=1,B=2,...Z=26)");
                System.out.println("4. Decrypt text (A=1,B=2,...Z=26)");
                System.out.println();
                System.out.println("Type any other character to exit");
                System.out.println();
                System.out.print("Select your choice: ");
                opt = sc.nextLine();
                switch (opt)
                {
                    case "1":
                        System.out.print("Enter phrase to encrypt: ");
                        phrase = sc.nextLine();
                        encrypt(phrase, true);
                        break;
                    case "2":
                        System.out.print("Enter phrase to decrypt: ");
                        phrase = sc.nextLine();
                        decrypt(phrase, true);
                        break;
                    case "3":
                        System.out.print("Enter phrase to encrypt: ");
                        phrase = sc.nextLine();
                        encrypt(phrase, false);
                        break;
                    case "4":
                        System.out.print("Enter phrase to decrypt: ");
                        phrase = sc.nextLine();
                        decrypt(phrase, false);
                        break;
                }
            }
        }
```



```
C:\Users\Anurati\Desktop\abcDemo>javac HillCipherExample.java

C:\Users\Anurati\Desktop\abcDemo>java HillCipherExample
Hill Cipher Implementation (2x2)
-------------------------
1. Encrypt text (A=0,B=1,...Z=25)
2. Decrypt text (A=0,B=1,...Z=25)
3. Encrypt text (A=1,B=2,...Z=26)
4. Decrypt text (A=1,B=2,...Z=26)

Type any other character to exit

Select your choice: 1
Enter phrase to encrypt: hillcipheralgorithm
Enter key matrix:
nwkc
Encoded phrase: HI-VC-UK-LI-KW-IW-WK-HE-LW-OW
```

Name-Nikhil Srivastava

Sec-c

U roll num—191500505

```
package com.sanfoundry.setandstring;

public class TranspositionCipher
{
    public static String selectedKey;
    public static char   sortedKey[];
    public static int    sortedKeyPos[];

    // default constructor define the default key
     public TranspositionCipher()
     {
         selectedKey = "megabuck";
         sortedKeyPos = new int[selectedKey.length()];
         sortedKey = selectedKey.toCharArray();
     }

     // Parameterized constructor define the custom key
     public TranspositionCipher(String myKey)
     {
         selectedKey = myKey;
         sortedKeyPos = new int[selectedKey.length()];
         sortedKey = selectedKey.toCharArray();
     }

     // To reorder data do the sorting on selected key
     public static void doProcessOnKey()
     {
         // Find position of each character in selected key and arrange it on
         // alphabetical order
         int min, i, j;
         char orginalKey[] = selectedKey.toCharArray();
         char temp;
         // First Sort the array of selected key
         for (i = 0; i < selectedKey.length(); i++)
         {
             min = i;
```

```
            for (j = i; j < selectedKey.length(); j++)
            {
                if (sortedKey[min] > sortedKey[j])
                {
                    min = j;
                }
            }
            if (min != i)
            {
                temp = sortedKey[i];
                sortedKey[i] = sortedKey[min];
                sortedKey[min] = temp;
            }
        }
        // Fill the position of array according to alphabetical order
        for (i = 0; i < selectedKey.length(); i++)
        {
            for (j = 0; j < selectedKey.length(); j++)
            {
                if (orginalKey[i] == sortedKey[j])
                    sortedKeyPos[i] = j;
            }
        }
    }

    // to encrypt the targeted string
    public static String doEncryption(String plainText)
    {
        int min, i, j;
        char orginalKey[] = selectedKey.toCharArray();
        char temp;
        doProcessOnKey();
        // Generate encrypted message by doing encryption using Transpotion
        // Cipher
        int row = plainText.length() / selectedKey.length();
        int extrabit = plainText.length() % selectedKey.length();
        int exrow = (extrabit == 0) ? 0 : 1;
        int rowtemp = -1, coltemp = -1;
        int totallen = (row + exrow) * selectedKey.length();
        char pmat[][] = new char[(row + exrow)][(selectedKey.length())];
        char encry[] = new char[totallen];
        int tempcnt = -1;
        row = 0;
        for (i = 0; i < totallen; i++)
        {
            coltemp++;
            if (i < plainText.length())
            {
                if (coltemp == (selectedKey.length()))
                {
                    row++;
                    coltemp = 0;
                }
                pmat[row][coltemp] = plainText.charAt(i);
            }
            else
            { // do the padding ...
                pmat[row][coltemp] = '*';
            }
        }
        int len = -1, k;
        for (i = 0; i < selectedKey.length(); i++)
        {
                    for (k = 0; k < selectedKey.length(); k++)
                    {
                        if (i == sortedKeyPos[k])
                        {
                            break;
                        }
                    }
                    for (j = 0; j <= row; j++)
                    {
                        len++;
                        encry[len] = pmat[j][k];
                    }
                }
                String p1 = new String(encry);
                return (new String(p1));
            }

            // to decrypt the targeted string
            public static String doDecryption(String s)
            {
                int min, i, j, k;
                char key[] = selectedKey.toCharArray();
                char encry[] = s.toCharArray();
                char temp;
                doProcessOnKey();
                // Now generating plain message
                int row = s.length() / selectedKey.length();
                char pmat[][] = new char[row][(selectedKey.length())];
                int tempcnt = -1;
                for (i = 0; i < selectedKey.length(); i++)
                {
                    for (k = 0; k < selectedKey.length(); k++)
                    {
                        if (i == sortedKeyPos[k])
                        {
                            break;
                        }
                    }
                    for (j = 0; j < row; j++)
                    {
                        tempcnt++;
                        pmat[j][k] = encry[tempcnt];
                    }
                }
                // store matrix character in to a single string
                char p1[] = new char[row * selectedKey.length()];
                k = 0;
                for (i = 0; i < row; i++)
                {
                    for (j = 0; j < selectedKey.length(); j++)
                    {
                        if (pmat[i][j] != '*')
                        {
                            p1[k++] = pmat[i][j];
                        }
                    }
                }
                p1[k++] = '\0';
                return (new String(p1));
            }

            @SuppressWarnings("static-access")
            public static void main(String[] args)
            {
                TranspositionCipher tc = new TranspositionCipher();
                System.out.println("Encrypted Message is: "
                        + tc.doEncryption("Sanfoundry"));
                System.out.println("Decrypted Message is: "
                        + tc.doDecryption(tc.doEncryption("Sanfoundry")));
            }
        }
```

Output:

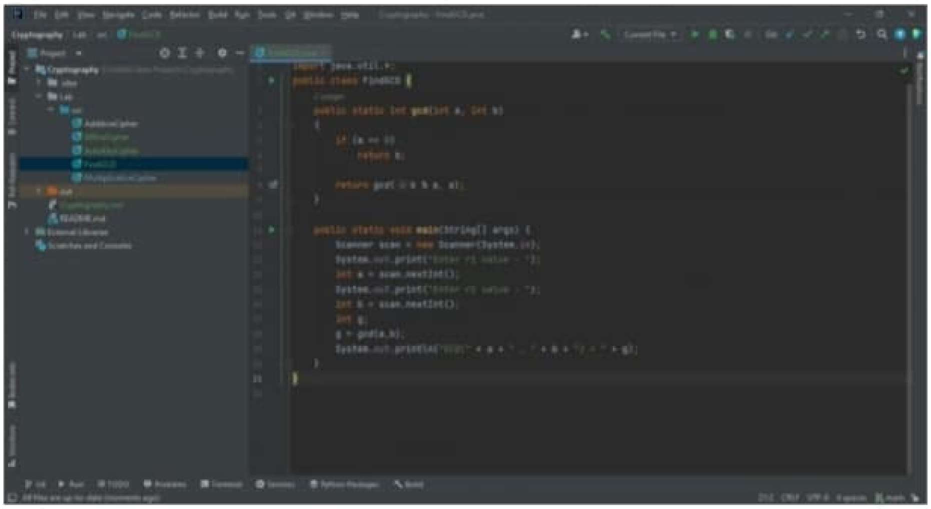
```
$ javac TranspositionCipher.java
$ java TranspositionCipher

Encrypted Message is: f*o*n*ayn*d*Sru*
Decrypted Message is: Sanfoundry
```

Name – Nikhil Srivastava

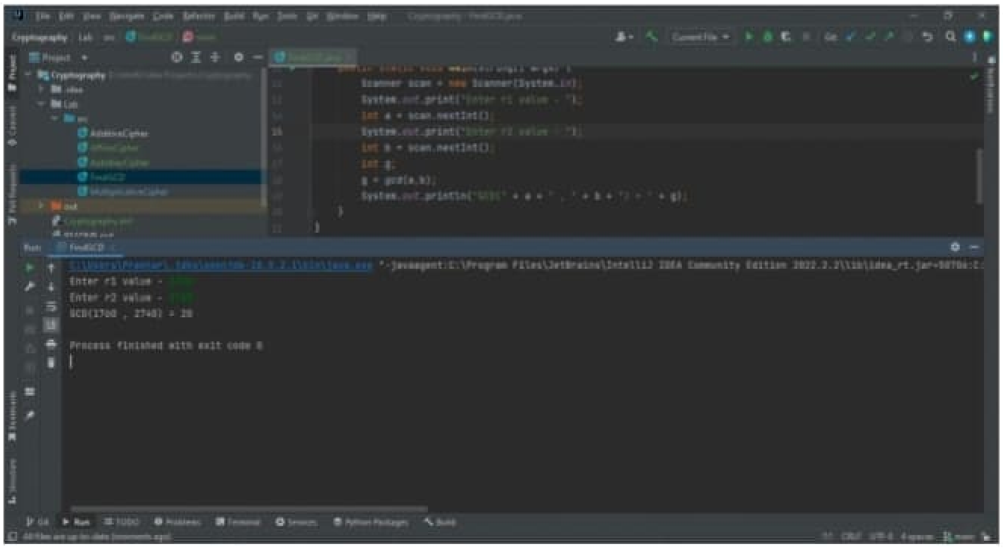
Section – C (2)

Roll No. – 29

Q - W.A.P. to implement Euclidean Algorithm to find the GCD of given numbers.



```
import java.util.*;
public class FindGCD {
    public static int gcd(int a, int b)
    {
        if (a == 0)
            return b;

        return gcd(b % a, a);
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter r1 value - ");
        int a = scan.nextInt();
        System.out.print("Enter r2 value - ");
        int b = scan.nextInt();
        int g;
        g = gcd(a,b);
        System.out.println("GCD(" + a + " , " + b + ") = " + g);
    }
}
```



```
[illegible]\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.2\lib\idea_rt.jar=50784:C:
Enter r1 value - [illegible]
Enter r2 value - [illegible]
GCD(1760 , 2740) = 20

Process finished with exit code 0
```

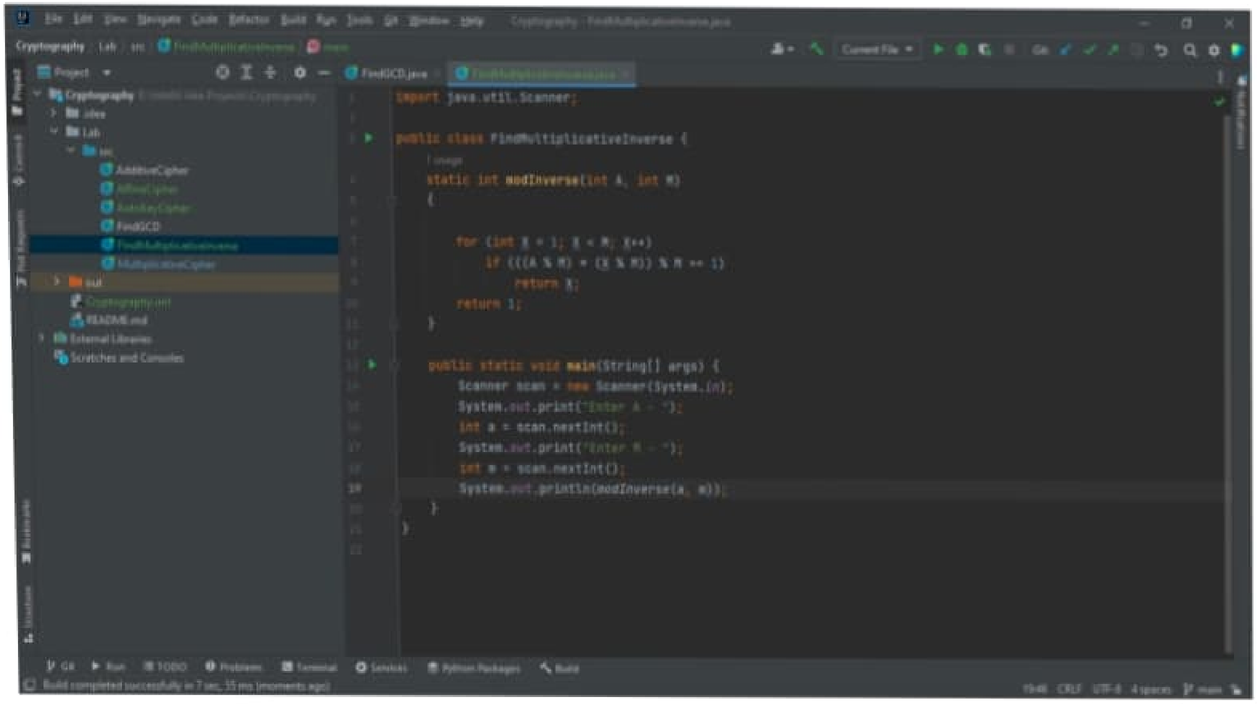
Name - Nikhil Srivastava

Section - C (2)

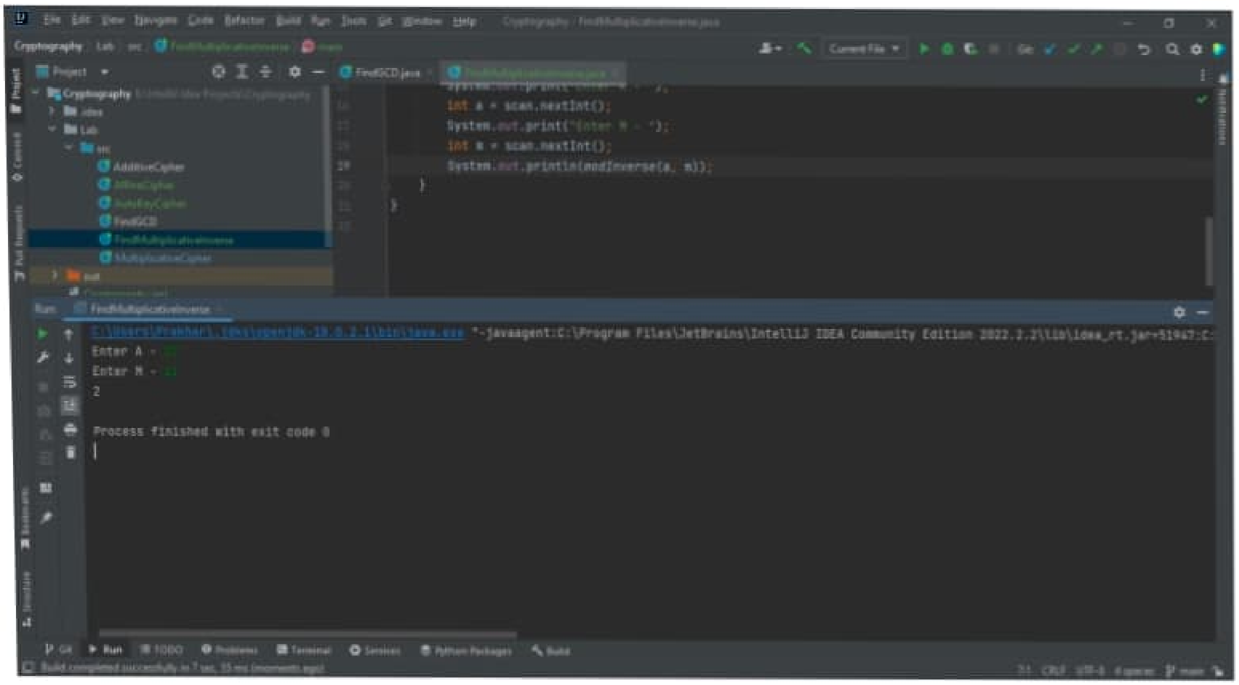
Roll No. - 29

Q - Write a program to find out the Multiplicative inverse of a given number by using Extended Euclidean algorithm.

```
import java.util.Scanner;

public class FindMultiplicativeInverse {

static int modInverse(int A, int M)

{

for (int X = 1; X < M; X++)

if (((A % M) * (X % M)) % M == 1)

return X;

return 1;

}

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

System.out.print("Enter A - ");

int a = scan.nextInt();

System.out.print("Enter M - ");

int m = scan.nextInt();

System.out.println(modInverse(a, m) }}
```



```java
import java.util.Scanner;

public class FindMultiplicativeInverse {
    static int modInverse(int A, int M)
    {

        for (int X = 1; X < M; X++)
            if (((A % M) * (X % M)) % M == 1)
                return X;
        return 1;
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter A - ");
        int a = scan.nextInt();
        System.out.print("Enter M - ");
        int m = scan.nextInt();
        System.out.println(modInverse(a, m));
    }
}
```



```java
        int a = scan.nextInt();
        System.out.print("Enter M - ");
        int m = scan.nextInt();
        System.out.println(modInverse(a, m));
    }
}
```

```
C:\Users\Prakhar\.jdks\openjdk-19.0.2.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.2.2\lib\idea_rt.jar=51947:C:
Enter A - [illegible]
Enter M - [illegible]
2

Process finished with exit code 0
```

Name-Nikhil Srivastava

Sec-c

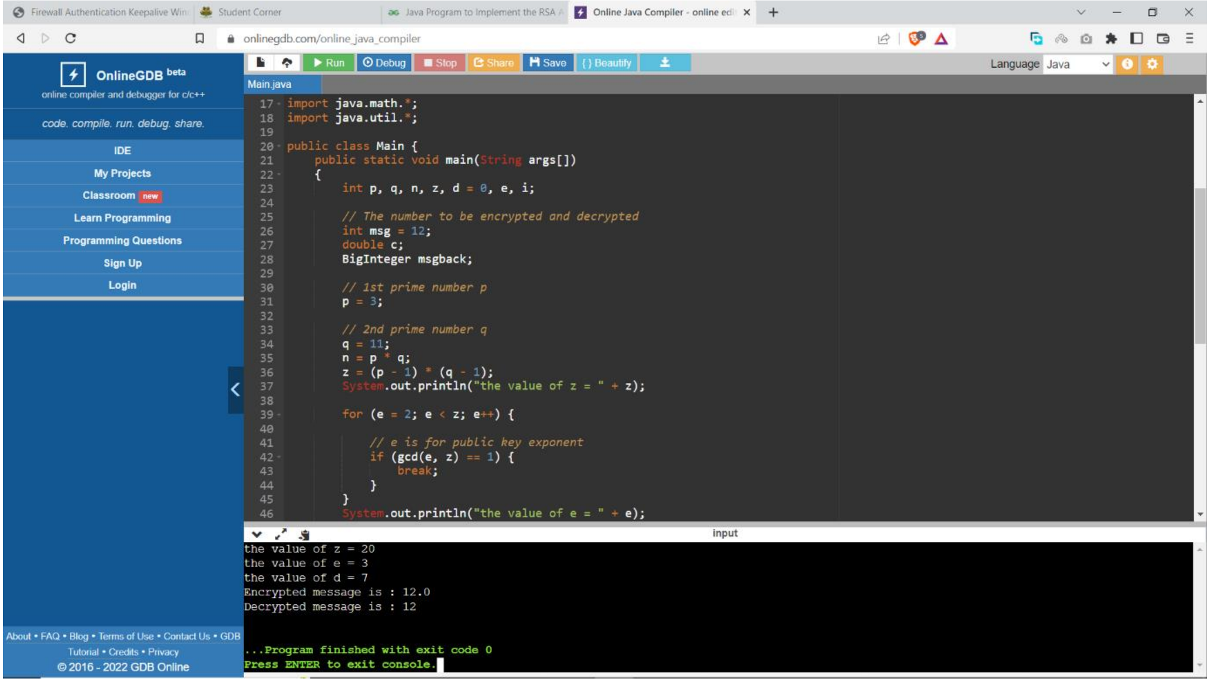
U roll num—191500505

```
import java.security.Key;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.SecureRandom;
import java.security.Security;

import javax.crypto.Cipher;

public class MainClass {
  public static void main(String[] args) throws Exception {
    Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider())
;

    byte[] input = "ab".getBytes();
    Cipher cipher = Cipher.getInstance("ElGamal/None/NoPadding", "BC");
    KeyPairGenerator generator = KeyPairGenerator.getInstance("ElGamal", "BC");
    SecureRandom random = new SecureRandom();

    generator.initialize(128, random);

    KeyPair pair = generator.generateKeyPair();
    Key pubKey = pair.getPublic();
    Key privKey = pair.getPrivate();
    cipher.init(Cipher.ENCRYPT_MODE, pubKey, random);
    byte[] cipherText = cipher.doFinal(input);
    System.out.println("cipher: " + new String(cipherText));

    cipher.init(Cipher.DECRYPT_MODE, privKey);
    byte[] plainText = cipher.doFinal(cipherText);
    System.out.println("plain : " + new String(plainText));
  }
}
```

Name-Nikhil Srivastava

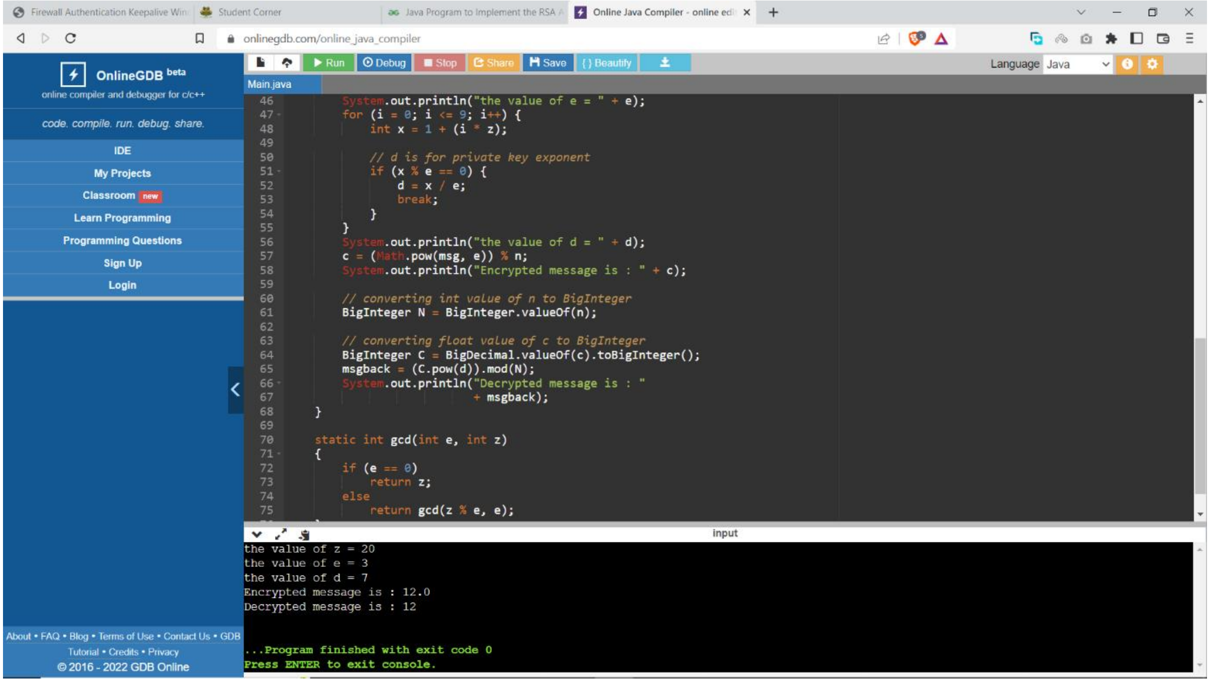
Sec-c

U roll num—191500505


Firewall Authentication Keepalive Win
Student Corner
Java Program to Implement the RSA A
Online Java Compiler - online edi
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

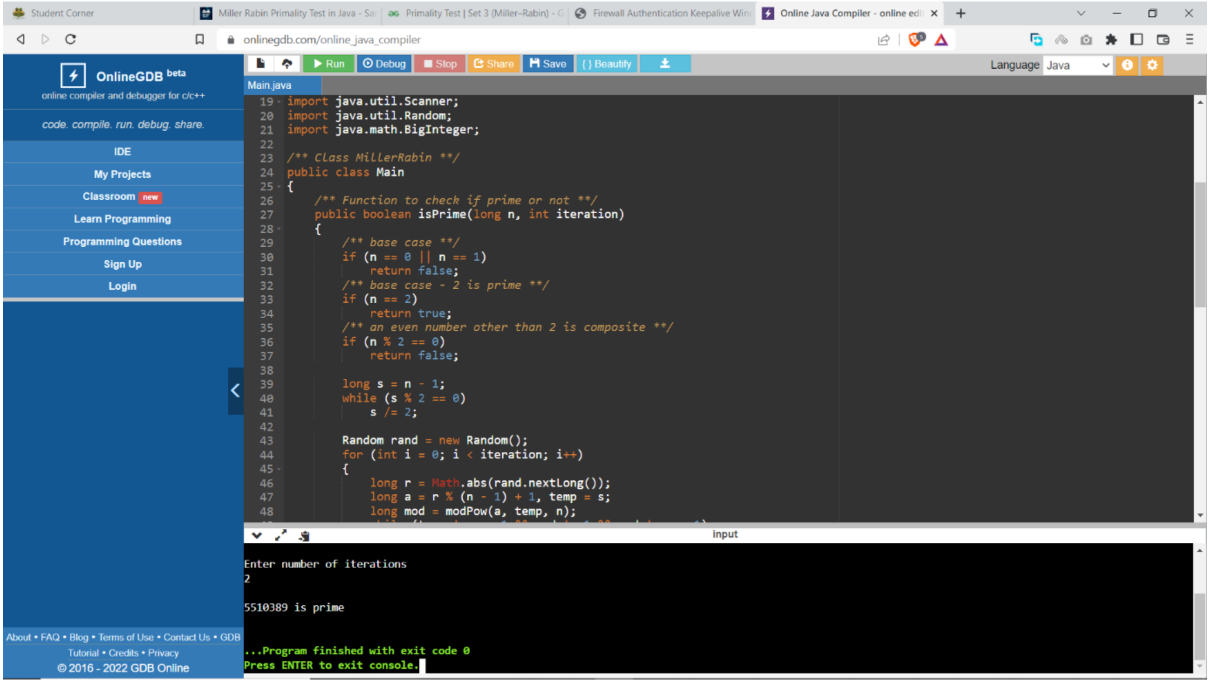
```java
import java.math.*;
import java.util.*;

public class Main {
    public static void main(String args[])
    {
        int p, q, n, z, d = 0, e, i;

        // The number to be encrypted and decrypted
        int msg = 12;
        double c;
        BigInteger msgback;

        // 1st prime number p
        p = 3;

        // 2nd prime number q
        q = 11;
        n = p * q;
        z = (p - 1) * (q - 1);
        System.out.println("the value of z = " + z);

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

            // e is for public key exponent
            if (gcd(e, z) == 1) {
                break;
            }
        }
        System.out.println("the value of e = " + e);
```

```
the value of z = 20
the value of e = 3
the value of d = 7
Encrypted message is : 12.0
Decrypted message is : 12


...Program finished with exit code 0
Press ENTER to exit console.
```


Firewall Authentication Keepalive Win
Student Corner
Java Program to Implement the RSA A
Online Java Compiler - online edi
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

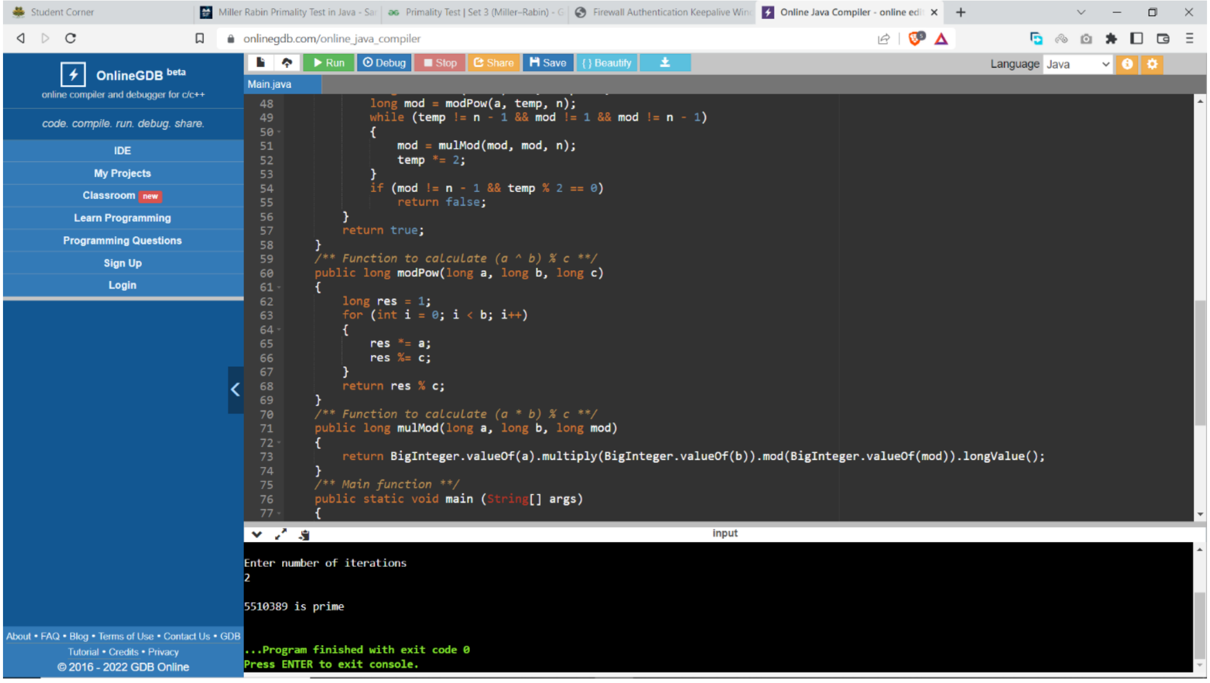
```java
        System.out.println("the value of e = " + e);
        for (i = 0; i <= 9; i++) {
            int x = 1 + (i * z);

            // d is for private key exponent
            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);

        // converting int value of n to BigInteger
        BigInteger N = BigInteger.valueOf(n);

        // converting float value of c to BigInteger
        BigInteger C = BigDecimal.valueOf(c).toBigInteger();
        msgback = (C.pow(d)).mod(N);
        System.out.println("Decrypted message is : "
                           + msgback);
    }

    static int gcd(int e, int z)
    {
        if (e == 0)
            return z;
        else
            return gcd(z % e, e);
```

```
the value of z = 20
the value of e = 3
the value of d = 7
Encrypted message is : 12.0
Decrypted message is : 12


...Program finished with exit code 0
Press ENTER to exit console.
```


Student Corner
Miller Rabin Primality Test in Java - Sar
Primality Test | Set 3 (Miller–Rabin) - G
Firewall Authentication Keepalive Win
Online Java Compiler - online edi
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

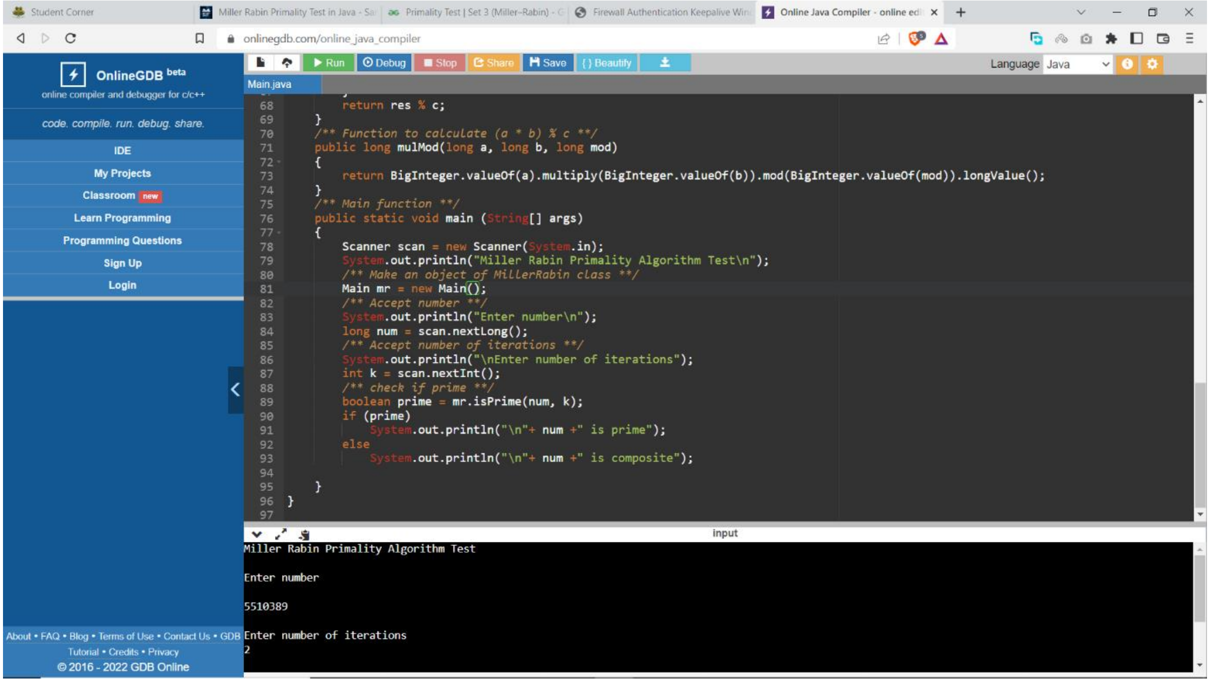
```java
import java.util.Scanner;
import java.util.Random;
import java.math.BigInteger;

/** Class MillerRabin **/
public class Main
{
    /** Function to check if prime or not **/
    public boolean isPrime(long n, int iteration)
    {
        /** base case **/
        if (n == 0 || n == 1)
            return false;
        /** base case - 2 is prime **/
        if (n == 2)
            return true;
        /** an even number other than 2 is composite **/
        if (n % 2 == 0)
            return false;

        long s = n - 1;
        while (s % 2 == 0)
            s /= 2;

        Random rand = new Random();
        for (int i = 0; i < iteration; i++)
        {
            long r = Math.abs(rand.nextLong());
            long a = r % (n - 1) + 1, temp = s;
            long mod = modPow(a, temp, n);
```

```
Enter number of iterations
2

5510389 is prime


...Program finished with exit code 0
Press ENTER to exit console.
```


Student Corner
Miller Rabin Primality Test in Java - Sar
Primality Test | Set 3 (Miller–Rabin) - G
Firewall Authentication Keepalive Win
Online Java Compiler - online edi
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

```java
            long mod = modPow(a, temp, n);
            while (temp != n - 1 && mod != 1 && mod != n - 1)
            {
                mod = mulMod(mod, mod, n);
                temp *= 2;
            }
            if (mod != n - 1 && temp % 2 == 0)
                return false;
        }
        return true;
    }
    /** Function to calculate (a ^ b) % c **/
    public long modPow(long a, long b, long c)
    {
        long res = 1;
        for (int i = 0; i < b; i++)
        {
            res *= a;
            res %= c;
        }
        return res % c;
    }
    /** Function to calculate (a * b) % c **/
    public long mulMod(long a, long b, long mod)
    {
        return BigInteger.valueOf(a).multiply(BigInteger.valueOf(b)).mod(BigInteger.valueOf(mod)).longValue();
    }
    /** Main function **/
    public static void main (String[] args)
    {
```

```
Enter number of iterations
2

5510389 is prime


...Program finished with exit code 0
Press ENTER to exit console.
```



Student Corner
Miller Rabin Primality Test in Java - Sa
Primality Test | Set 3 (Miller–Rabin) - G
Firewall Authentication Keepalive Win
Online Java Compiler - online ed
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language Java
Main.java
input

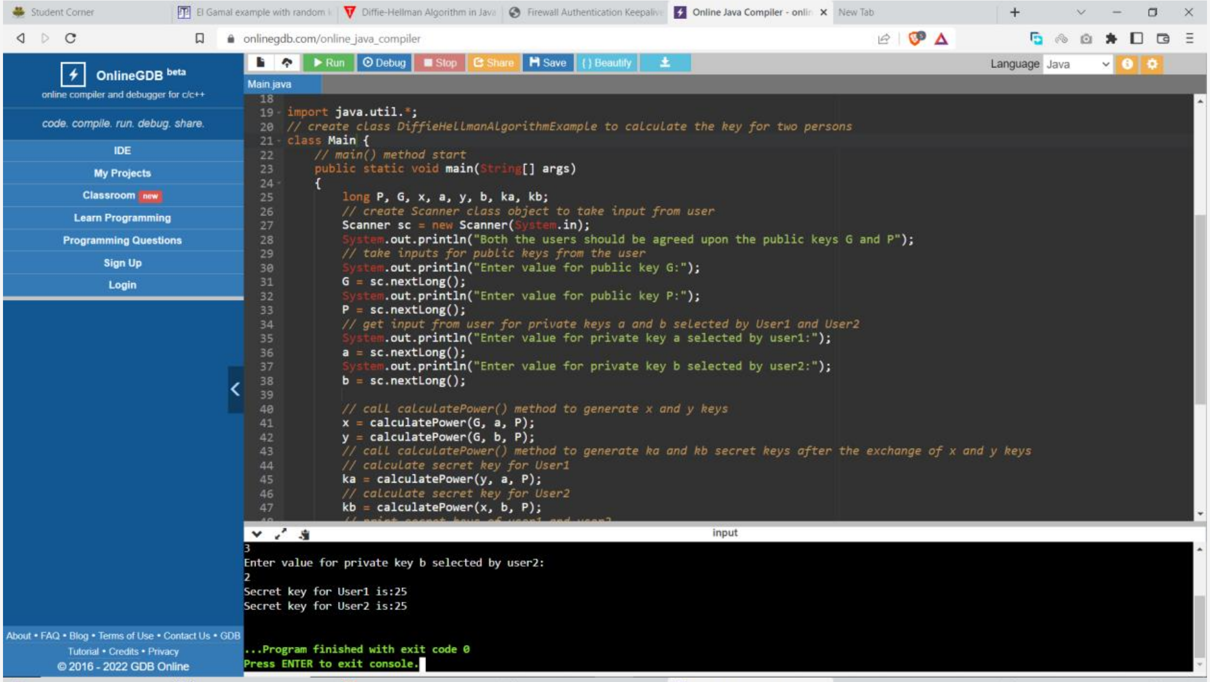
```
        return res % c;
    }
    /** Function to calculate (a * b) % c **/
    public long mulMod(long a, long b, long mod)
    {
        return BigInteger.valueOf(a).multiply(BigInteger.valueOf(b)).mod(BigInteger.valueOf(mod)).longValue();
    }
    /** Main function **/
    public static void main (String[] args)
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Miller Rabin Primality Algorithm Test\n");
        /** Make an object of MillerRabin class **/
        Main mr = new Main();
        /** Accept number **/
        System.out.println("Enter number\n");
        long num = scan.nextLong();
        /** Accept number of iterations **/
        System.out.println("\nEnter number of iterations");
        int k = scan.nextInt();
        /** check if prime **/
        boolean prime = mr.isPrime(num, k);
        if (prime)
            System.out.println("\n"+ num +" is prime");
        else
            System.out.println("\n"+ num +" is composite");

    }
}
```

```
Miller Rabin Primality Algorithm Test

Enter number

5510389

Enter number of iterations
2
```

Name-Nikhil Srivastava

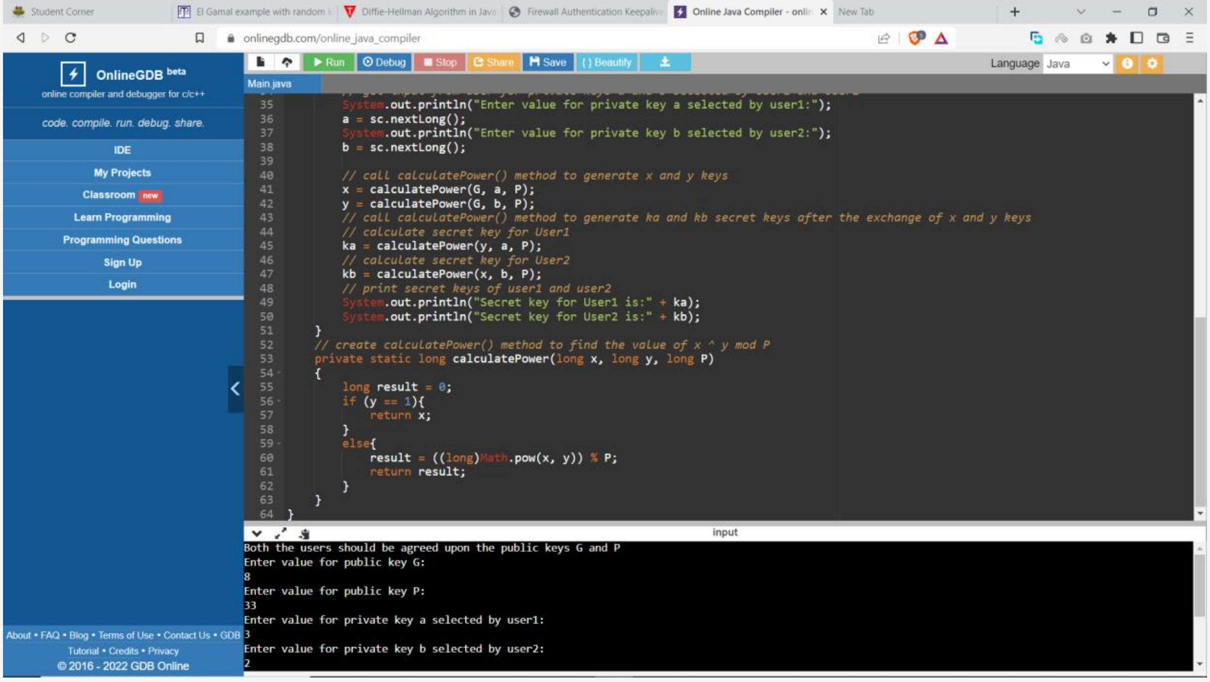
Sec-c

U roll num—191500505


Student Corner
El Gamal example with random
Diffie-Hellman Algorithm in Java
Firewall Authentication Keepalive
Online Java Compiler - onli
New Tab
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Java
Main.java
input

```java
import java.util.*;
// create class DiffieHellmanAlgorithmExample to calculate the key for two persons
class Main {
    // main() method start
    public static void main(String[] args)
    {
        long P, G, x, a, y, b, ka, kb;
        // create Scanner class object to take input from user
        Scanner sc = new Scanner(System.in);
        System.out.println("Both the users should be agreed upon the public keys G and P");
        // take inputs for public keys from the user
        System.out.println("Enter value for public key G:");
        G = sc.nextLong();
        System.out.println("Enter value for public key P:");
        P = sc.nextLong();
        // get input from user for private keys a and b selected by User1 and User2
        System.out.println("Enter value for private key a selected by user1:");
        a = sc.nextLong();
        System.out.println("Enter value for private key b selected by user2:");
        b = sc.nextLong();

        // call calculatePower() method to generate x and y keys
        x = calculatePower(G, a, P);
        y = calculatePower(G, b, P);
        // call calculatePower() method to generate ka and kb secret keys after the exchange of x and y keys
        // calculate secret key for User1
        ka = calculatePower(y, a, P);
        // calculate secret key for User2
        kb = calculatePower(x, b, P);
```

```
3
Enter value for private key b selected by user2:
2
Secret key for User1 is:25
Secret key for User2 is:25


...Program finished with exit code 0
Press ENTER to exit console.
```


Student Corner
El Gamal example with random
Diffie-Hellman Algorithm in Java
Firewall Authentication Keepalive
Online Java Compiler - onli
New Tab
onlinegdb.com/online_java_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Java
Main.java
input

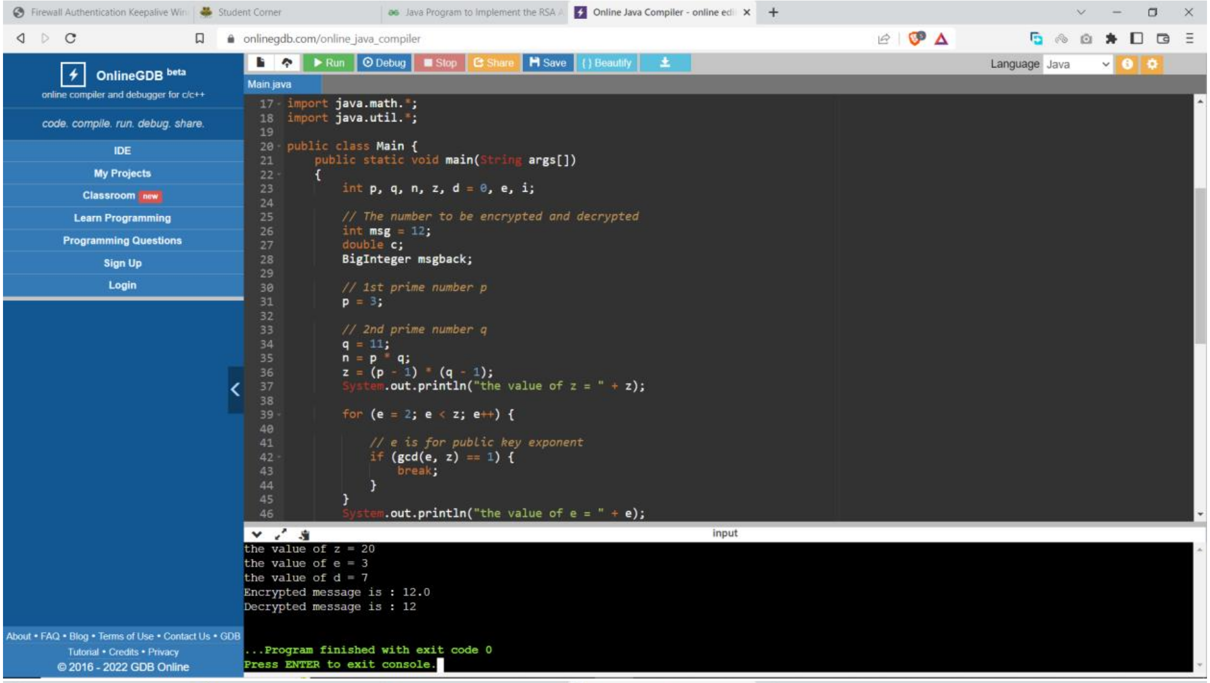
```java
        System.out.println("Enter value for private key a selected by user1:");
        a = sc.nextLong();
        System.out.println("Enter value for private key b selected by user2:");
        b = sc.nextLong();

        // call calculatePower() method to generate x and y keys
        x = calculatePower(G, a, P);
        y = calculatePower(G, b, P);
        // call calculatePower() method to generate ka and kb secret keys after the exchange of x and y keys
        // calculate secret key for User1
        ka = calculatePower(y, a, P);
        // calculate secret key for User2
        kb = calculatePower(x, b, P);
        // print secret keys of user1 and user2
        System.out.println("Secret key for User1 is:" + ka);
        System.out.println("Secret key for User2 is:" + kb);
    }
    // create calculatePower() method to find the value of x ^ y mod P
    private static long calculatePower(long x, long y, long P)
    {
        long result = 0;
        if (y == 1){
            return x;
        }
        else{
            result = ((long)Math.pow(x, y)) % P;
            return result;
        }
    }
}
```

```
Both the users should be agreed upon the public keys G and P
Enter value for public key G:
8
Enter value for public key P:
33
Enter value for private key a selected by user1:
3
Enter value for private key b selected by user2:
2
```

Name-Nikhil Srivastava

Sec-C

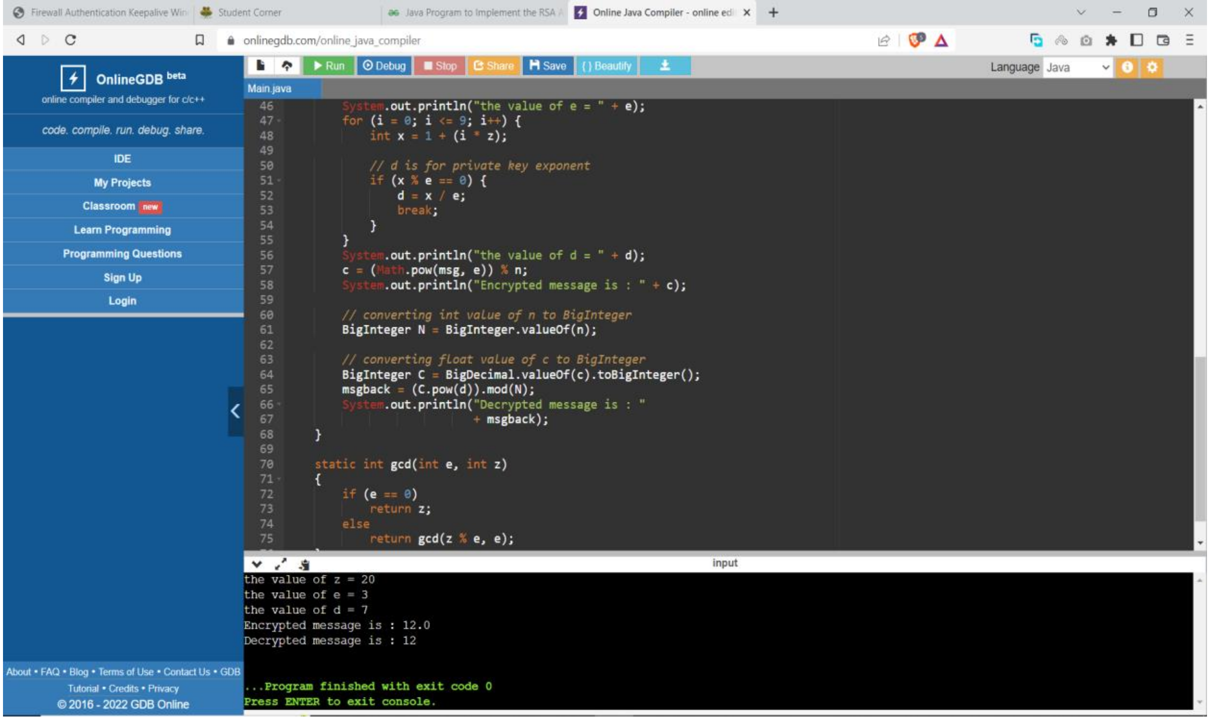
U.roll no. 191500505



Firewall Authentication Keepalive Win
Student Corner
Java Program to Implement the RSA
Online Java Compiler - online ed
onlinegdb.com/online_java_compiler
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Java
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Main.java
input

```java
import java.math.*;
import java.util.*;

public class Main {
    public static void main(String args[])
    {
        int p, q, n, z, d = 0, e, i;

        // The number to be encrypted and decrypted
        int msg = 12;
        double c;
        BigInteger msgback;

        // 1st prime number p
        p = 3;

        // 2nd prime number q
        q = 11;
        n = p * q;
        z = (p - 1) * (q - 1);
        System.out.println("the value of z = " + z);

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

            // e is for public key exponent
            if (gcd(e, z) == 1) {
                break;
            }
        }
        System.out.println("the value of e = " + e);
```

```
the value of z = 20
the value of e = 3
the value of d = 7
Encrypted message is : 12.0
Decrypted message is : 12


...Program finished with exit code 0
Press ENTER to exit console.
```



Firewall Authentication Keepalive Win
Student Corner
Java Program to Implement the RSA
Online Java Compiler - online ed
onlinegdb.com/online_java_compiler
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Java
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Main.java
input

```java
        System.out.println("the value of e = " + e);
        for (i = 0; i <= 9; i++) {
            int x = 1 + (i * z);

            // d is for private key exponent
            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);

        // converting int value of n to BigInteger
        BigInteger N = BigInteger.valueOf(n);

        // converting float value of c to BigInteger
        BigInteger C = BigDecimal.valueOf(c).toBigInteger();
        msgback = (C.pow(d)).mod(N);
        System.out.println("Decrypted message is : "
                           + msgback);
    }

    static int gcd(int e, int z)
    {
        if (e == 0)
            return z;
        else
            return gcd(z % e, e);
```

```
the value of z = 20
the value of e = 3
the value of d = 7
Encrypted message is : 12.0
Decrypted message is : 12


...Program finished with exit code 0
Press ENTER to exit console.
```

Name-Nikhil Srivastava

Sec-c

U roll num—191500505

```
import java.security.Key;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.SecureRandom;
import java.security.Security;

import javax.crypto.Cipher;

public class MainClass {
  public static void main(String[] args) throws Exception {
    Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider())
;

    byte[] input = "ab".getBytes();
    Cipher cipher = Cipher.getInstance("ElGamal/None/NoPadding", "BC");
    KeyPairGenerator generator = KeyPairGenerator.getInstance("ElGamal", "BC");
    SecureRandom random = new SecureRandom();

    generator.initialize(128, random);

    KeyPair pair = generator.generateKeyPair();
    Key pubKey = pair.getPublic();
    Key privKey = pair.getPrivate();
    cipher.init(Cipher.ENCRYPT_MODE, pubKey, random);
    byte[] cipherText = cipher.doFinal(input);
    System.out.println("cipher: " + new String(cipherText));

    cipher.init(Cipher.DECRYPT_MODE, privKey);
    byte[] plainText = cipher.doFinal(cipherText);
    System.out.println("plain : " + new String(plainText));
  }
}
```