

CNS MODEL EXAM – 10.11.2020

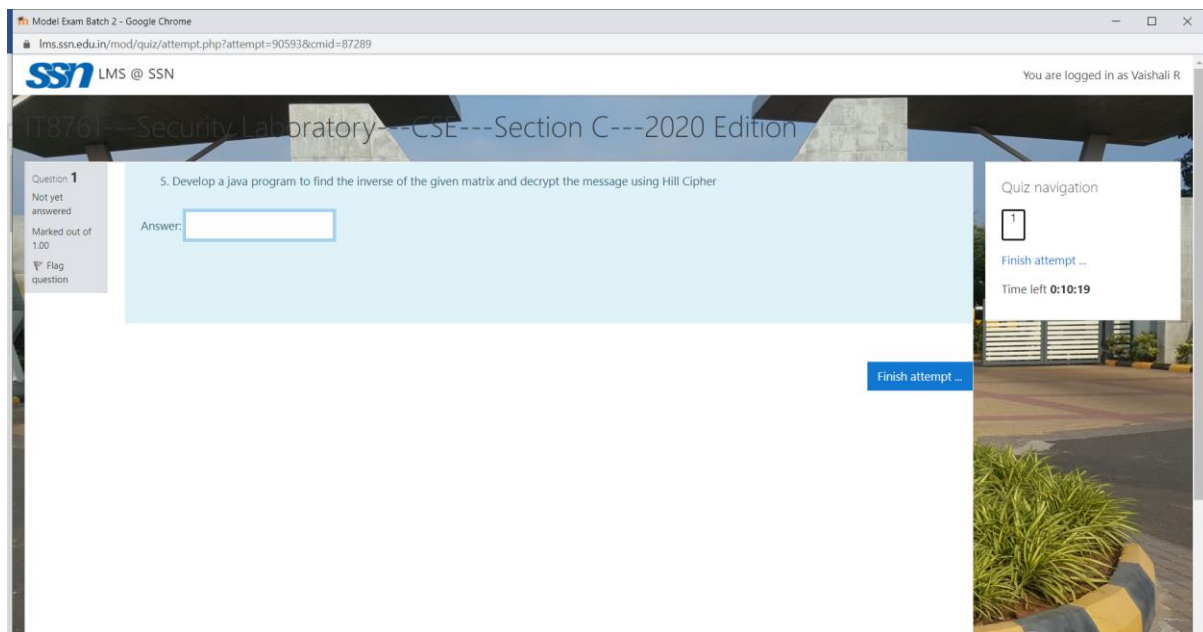
BATCH 2

NAME : VAISHALI R

REG.NO: 312217104181

CSE-C

5) HILL CIPHER



CODE:

```
import java.util.Scanner;

class Main {
    String cipherText;
    String plainText;
    String key;
    int cipherMat[][]=new int[3][1];
    int plainMat[][]= new int [3][1];
    int keyMat[][]=new int[3][3];
    int inv_keyMat[][]=new int[3][3];
    int adj[][]=new int[3][3];
    int d;
    int detInv;
```

```

void printCipherMat(){
    int idx=0;
    System.out.println("The plain text matrix is: ");
    for(int i=0;i<3;i++){
        cipherMat[i][0]=cipherText.charAt(idx++)-65;
        System.out.println(cipherMat[i][0]);
    }
}

void printPlainMat(){
    int idx=0;
    System.out.println("The cipher text matrix is: ");
    for(int i=0;i<3;i++){
        plainMat[i][0]=plainText.charAt(idx++)-65;
        System.out.println(plainMat[i][0]);
    }
}

void printKeyMat(){
    int idx=0;
    System.out.println("The key matrix is: ");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            keyMat[i][j]=key.charAt(idx++)-65;
            System.out.print(keyMat[i][j]+" ");
        }
        System.out.println();
    }
}

public boolean validateKey(String key){
    return key.length()==9;
}

void findAdjointMat(){
    int sign=1;
    int cofactor[][]=new int [3][3];
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            findCofactorMat(keyMat,3,cofactor,i,j);
            if((i+j)%2==0)
                sign=1;
            else
                sign=-1;
            adj[j][i]=sign*findDet(cofactor,2);
            while(adj[j][i]<0){
                adj[j][i]+=26;
            }
        }
    }
}

```

```

    }
}
System.out.println("\nThe adjoint matrix is ");
for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
        System.out.print(adj[i][j]+" ");
    }
    System.out.println();
}
}

void findInverseKeyMat(){
    detInv=findInverse(d,26);
    System.out.println("\nThe det value is "+d);
    System.out.println("\nThe det inv value is "+detInv);
    System.out.println("The inverse key matrix is: ");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            inv_keyMat[i][j]=(adj[i][j]*detInv)%26;
            System.out.print(inv_keyMat[i][j]+" ");
        }
        System.out.println();
    }
}

void findCofactorMat(int mat[][],int n,int cofactor[][],int r,int
c){
    int x=0,y=0;
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if(i!=r && j!=c){
                cofactor[x][y++]=mat[i][j];
                if(y==n-1){
                    x++;
                    y=0;
                }
            }
        }
    }
}

int findDet(int mat[][],int n){
    int det=0,sign=1;
    int cofactor[][] =new int[3][3];

    if(n==1) return mat[0][0];

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

```

```

        findCofactorMat(mat,n,cofactor,0,i);
        det+=(sign*mat[0][i]*findDet(cofactor,n-1));
        sign=-1*sign;
    }
    while(det<0){
        det+=26;
    }
    det=det%26;
    return det;
}

int findInverse(int x,int m){
    x=x%m;
    for(int i=0;i<m;i++){
        if((x*i)%m==1)
            return i;
    }
    return 1;
}

boolean isInvertible(){
    d=findDet(keyMat,3);
    if(d==0 || d%13==0 || d%2==0)
        return false;
    return true;
}

String decrypt(){
    System.out.println("\nThe plain text matrix is: ");
    for(int i=0;i<3;i++){
        for(int j=0;j<1;j++){
            plainMat[i][j]=0;
            for(int k=0;k<3;k++){
                plainMat[i][j]+=inv_keyMat[i][k]*cipherMat[k][j];
            }
            plainMat[i][j]=plainMat[i][j]%26;
            System.out.print(plainMat[i][j]+" ");
        }
        System.out.println();
    }
    plainText="";
    for(int i=0;i<3;i++){
        plainText+=(char)(plainMat[i][0] + 65);
    }
    //System.out.println("The plaintext is "+plainText);
    return plainText;
}

```

```

String encrypt(){
System.out.println("\nThe cipher text matrix is: ");
for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
        cipherMat[i][j]=0;
        for(int k=0;k<3;k++){
            cipherMat[i][j]+=keyMat[i][k]*plainMat[k][j];
        }
        cipherMat[i][j]=cipherMat[i][j]%26;
        System.out.print(cipherMat[i][j]+" ");
    }
    System.out.println();
}
cipherText="";
for(int i=0;i<3;i++){
    cipherText+=(char)(cipherMat[i][0] + 65);
}
//System.out.println("The cipher text is "+cipherText);
return cipherText;
}

```

```

public static void main(String[] args) {
    Main hc=new Main();
    Scanner sc= new Scanner(System.in);
    System.out.println("HILL CIPHER ENCRYPTION");
    System.out.print("\nEnter the plain text: ");
    String pt =sc.next();
    System.out.print("\nEnter the key: ");
    hc.key=sc.next();
    while(!hc.validateKey(hc.key)){
        System.out.println("Invalid key");
        System.out.print("Enter the key: ");
        hc.key=sc.next();
    }
    String ct="";
    hc.printKeyMat();

    for(int i=0;i<pt.length();i=i+3){
        hc.plainText=pt.substring(i,i+3);
        hc.printPlainMat();
        ct+=hc.encrypt();
    }
}

```

```

System.out.println("The cipher text is "+ct);

```

```

System.out.println("\nHILL CIPHER DECRYPTION");
System.out.print("\nEnter the cipher text: ");

```

```

ct=sc.next();
System.out.print("\nEnter the key: ");
hc.key=sc.next();
while(!hc.validateKey(hc.key)){
    System.out.println("Invalid key");
    System.out.print("\nEnter the key: ");
    hc.key=sc.next();
}

pt="";
hc.printKeyMat();

if(hc.isInvertible()){
    System.out.println("\nKey Matrix is invertible");
    hc.findAdjointMat();
    hc.findInverseKeyMat();
    for(int i=0;i<ct.length();i=i+3){
        hc.cipherText=ct.substring(i,i+3);
        hc.printCipherMat();
        pt+=hc.decrypt();
    }
    System.out.println("The plain text is "+pt);
}
else{
    System.out.println("Key Matrix is not invertible");
}
}
}

```

OUTPUT:

EXAMPLE 1:

PLAINTEXT: PAY

KEY: BACKUPABC

```
❏ java -classpath ./run_dir/junit-4.12.jar:target/dependency/* Main  
HILL CIPHER ENCRYPTION
```

Enter the plain text: PAY

Enter the key: BACKUPABC

The key matrix is:

1 0 2
10 20 15
0 1 2

The cipher text matrix is:

15
0
24

The cipher text matrix is:

11
16
22

The cipher text is LQW

HILL CIPHER DECRYPTION

Enter the cipher text: LQW

Enter the key: BACKUPABC

The key matrix is:

1 0 2
10 20 15
0 1 2

Key Matrix is invertible

The adjoint matrix is

25 2 12
6 2 5
10 25 20

The det value is 19

The det inv value is 11

HILL CIPHER DECRYPTION

Enter the cipher text: LQW

Enter the key: BACKUPABC

The key matrix is:

1 0 2
10 20 15
0 1 2

Key Matrix is invertible

The adjoint matrix is

25 2 12
6 2 5
10 25 20

The det value is 19

The det inv value is 11

The inverse key matrix is:

15 22 2
14 22 3
6 15 12

The plain text matrix is:

11
16
22

The plain text matrix is:

15
0
24

The plain text is PAY



EXAMPLE 2:

PLAINTEXT: PAYMOREMONEY

KEY:ALPHABETA


```
❖ java -classpath ./run_dir/junit-4.12.jar:target/dependency/* Main  
HILL CIPHER ENCRYPTION
```

```
Enter the plain text: PAYMOREMONEY
```

```
Enter the key: ALPHABETA
```

```
The key matrix is:
```

```
0 11 15
```

```
7 0 1
```

```
4 19 0
```

```
The cipher text matrix is:
```

```
15
```

```
0
```

```
24
```

```
The cipher text matrix is:
```

```
22
```

```
25
```

```
8
```

```
The cipher text matrix is:
```

```
12
```

```
14
```

```
17
```

```
The cipher text matrix is:
```

```
19
```

```
23
```

```
2
```

```
The cipher text matrix is:
```

```
4
```

```
12
```

```
14
```

```
The cipher text matrix is:
```

```
4
```

```
16
```

```
10
```

```
The cipher text matrix is:
```

```
13
```

```
4
```

```
24
```

The cipher text matrix is:
14
11
24
The cipher text is WZITXCEQKOLY

HILL CIPHER DECRYPTION

Enter the cipher text: WZITXCEQKOLY

Enter the key: ALPHABETA

The key matrix is:

0 11 15

7 0 1

4 19 0

Key Matrix is invertible

The adjoint matrix is

7 25 11

4 18 1

3 18 1

The det value is 11

The det inv value is 19

The inverse key matrix is:

3 7 1

24 4 19

5 4 19

The plain text matrix is:

22

25

8

```
The plain text matrix is:
15
0
24
The plain text matrix is:
19
23
2

The plain text matrix is:
12
14
17
The plain text matrix is:
4
16
10

The plain text matrix is:
4
12
14
The plain text matrix is:
14
11
24

The plain text matrix is:
13
4
24
The plain text is PAYMOREMONEY
❖
```