18/12/2020.

IT 8761 Security laboratory

3122171040 54
Harshavardhini. L

Aim:

To develop a java program to find the inverse of the given matrix and encrypt the message using Hill Cipher.

Algorithm.

1. Read the input key matrix and the plain text.
2. To encrypt the plain text, each block of n-letter corresponding to the matrix is multiplied with the key matrix.
3. The matrix is then reduced by modulo 26 for each numbers.
4. The final matrices obtained is combined together to arrive at the encrypted text.
5. Print the inverse matrix and the encrypted text.

Hill Cipher encryption with Inverse Key Matrix Generation.

Program:

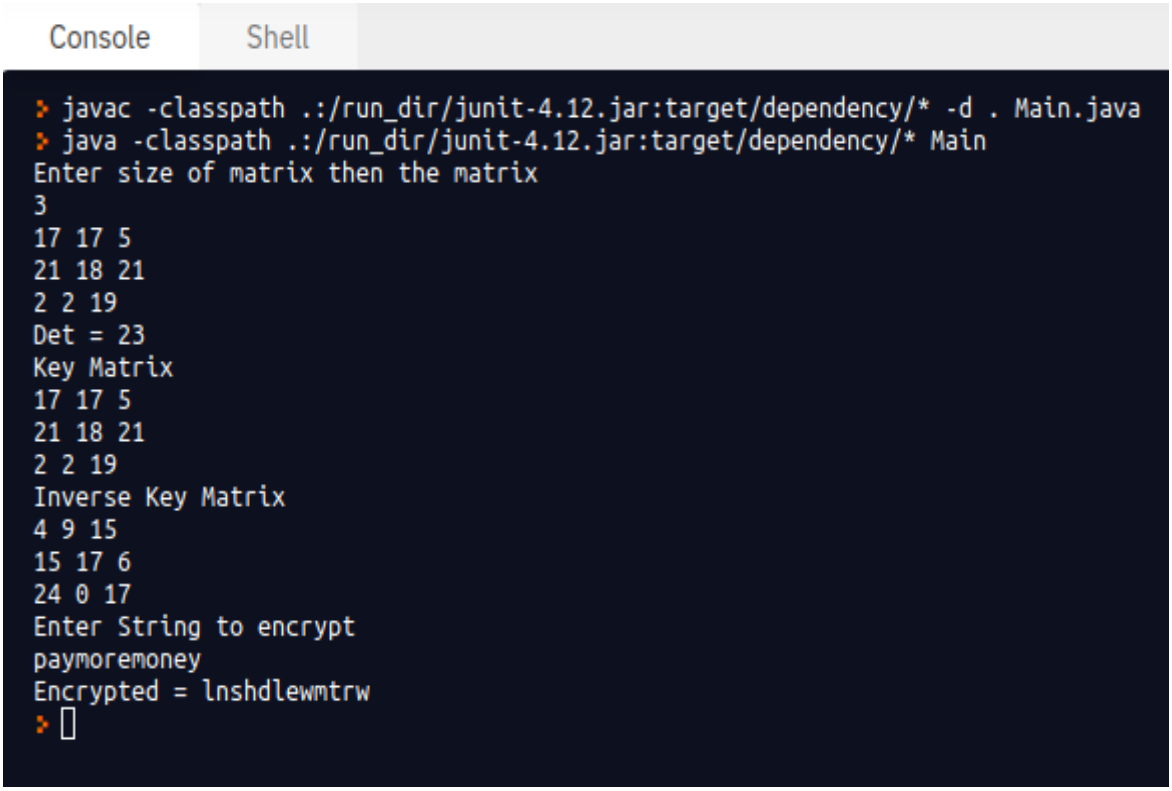
```
import java.util.Scanner;
class Main {
int mat[][];
int matinv[][];
int n;

public static void main(String[] args) {
Scanner in = new Scanner(System.in);
System.out.println("Enter size of matrix then the matrix ");
int n = in.nextInt();
int mat[][] = new int[n][n];
for (int i = 0; i < n; i++) {
for (int j = 0; j < n; j++)
mat[i][j] = in.nextInt();
}
Main m = new Main(mat);
System.out.println("Enter String to encrypt ");
String text = in.next();
System.out.println("Encrypted = " + m.encrypt(text));
in.close();
}

Main(int mat[][]) {
n = mat.length;
this.mat = new int[n][n];
this.matinv = new int[n][n];
for (int i = 0; i < n; i++) {
for (int j = 0; j < n; j++)
this.mat[i][j] = mat[i][j];
}
// System.out.println("Det = " + determinant(this.mat));
int det = determinant(this.mat) % 26;
if (det < 0)
det += 26;
System.out.println("Det = " + det);
if (det == 0 || gcd(det, 26) != 1)
throw new RuntimeException("Not invertible");
matinv = inverse(this.mat);
printmat();
}

private int gcd(int a, int b) {
if (b == 0)
return a;
return gcd(b, a % b);
}
```

```
public int determinant(int[][] a) {
int n = a.length;
if (n == 1)
return a[0][0];
else if (n == 2)
return (a[0][0] * a[1][1]) - (a[0][1] * a[1][0]);
int[][] temp = new int[n - 1][n - 1];
int ans = 0;
for (int i = 0; i < n; i++) {
for (int j = 1; j < n; j++) {
int rn = 0;
for (int k = 0; k < n; k++) {
if (k == i)
continue;
temp[j - 1][rn++] = a[j][k];
}
}
ans += Math.pow(-1.0, 2.0 + i) * a[0][i] * determinant(temp);
}
return ans;
}

public int[][] adjoint(int[][] a) {
int n = a.length;
int adj[][] = new int[n][n];
for (int i = 0; i < n; i++) {
for (int j = 0; j < n; j++) {
int temp[][] = new int[n - 1][n - 1];
int r = 0, c = 0;
for (int i1 = 0; i1 < n; i1++) {
for (int j1 = 0; j1 < n; j1++) {
if (i1 == i || j1 == j)
continue;
temp[r][c++] = a[i1][j1];
if (c == n - 1) {
c = 0;
r++;
}
}
}
adj[j][i] = (int) Math.pow(-1.0, (double) i + j) * determinant(temp);
}
}
return adj;
}

static int modpow(int x, int y, int p) {
int res = 1;
x = x % p;
if (x == 0)
```

```
return 0;
while (y > 0) {
if ((y & 1) == 1)
res = (res * x) % p;
y >>= 1;
x = (x * x) % p;
}
return res;
}

private int[][] inverse(int[][] a) {
int n = a.length;
int det = determinant(a) % 26;
if (det < 0)
det += 26;
int detinv = modpow(det, 11, 26);
int adj[][] = adjoint(a);
for (int i = 0; i < n; i++) {
for (int j = 0; j < n; j++) {
adj[i][j] = ((adj[i][j] * detinv) % 26 + 26) % 26;
}
}
return adj;
}

public String encrypt(String text) {
String ans = "";
int textn = text.length();
for (int e = 0; e < textn; e += n) {
String res = "";
for (int i = 0; i < n; i++) {
int temp = 0;
for (int j = 0; j < n; j++)
temp += mat[i][j] * ((int) text.charAt(e + j) - 97);
temp = temp % 26;
if (temp < 0)
temp += 26;
res += (char) (97 + temp);
}
ans += res;
}
return ans;
}

public void printmat() {
System.out.println("Key Matrix");
for (int i = 0; i < this.n; i++) {
for (int j = 0; j < this.n; j++) {
System.out.print(mat[i][j] + " ");
}
System.out.print("\n");
```

```
}
System.out.println("Inverse Key Matrix");
for (int i = 0; i < this.n; i++) {
for (int j = 0; j < this.n; j++) {
System.out.print(matinv[i][j] + " ");
}
System.out.print("\n");
}
}
}
```

Output:



Result:

Thus a java program to find the inverse of a given matrix and the encryption of a message using hill cipher has been implemented and executed successfully.