

Lab2

Shlok Kamath

21BAI1844

Cryptography and Network Security Lab (BCSE309P)

Professor: Dr. RAJESH R

Today's task:

1. Hill Cipher

Hill Cipher:

Code:

```
public class Main {
    static final int N = 3;

    private static int help(int a, int p){
        int[] arr = {0, p, a, 0, 0, 1, 0};
        while(arr[2]!=0){
            arr[0] = arr[1]/arr[2];
            arr[3] = arr[1]%arr[2];
            arr[6] = arr[4] - arr[0]*arr[5];
            arr[1] = arr[2];
            arr[2] = arr[3];
            arr[4] = arr[5];
            arr[5] = arr[6];
        }
        if(arr[4]<0)
            return p+arr[4];
        else
            return arr[4];
    }

    private static void getCofactor(int A[][], int temp[][], int p, int q,
int n)
    {
        int i = 0, j = 0;
        for (int row = 0; row < n; row++)
        {
            for (int col = 0; col < n; col++)
            {
                if (row != p && col != q)
                {
                    temp[i][j++] = A[row][col];
                    if (j == n - 1)
                    {
                        j = 0;
                        i++;
                    }
                }
            }
        }
    }
}
```

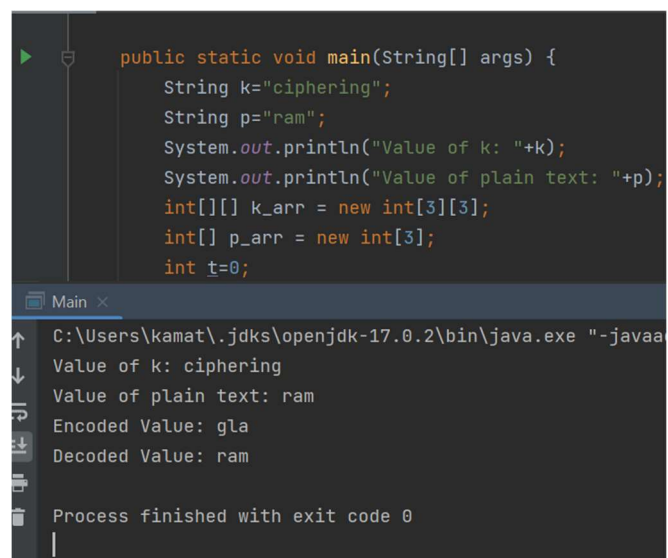
```
    }  
    }  
    }  
private static int determinant(int A[][], int n)  
{  
    int D = 0;  
    if (n == 1)  
        return A[0][0];  
  
    int [][]temp = new int[N][N];  
  
    int sign = 1;  
    for (int f = 0; f < n; f++)  
    {  
        getCofactor(A, temp, 0, f, n);  
        D += sign * A[0][f] * determinant(temp, n - 1);  
        sign = -sign;  
    }  
  
    return D;  
}  
  
private static void adjoint(int A[][],int [][]adj)  
{  
    if (N == 1)  
    {  
        adj[0][0] = 1;  
        return;  
    }  
    int sign = 1;  
    int [][]temp = new int[N][N];  
  
    for (int i = 0; i < N; i++)  
    {  
        for (int j = 0; j < N; j++)  
        {  
            getCofactor(A, temp, i, j, N);  
            sign = ((i + j) % 2 == 0)? 1: -1;  
            adj[j][i] = (sign)*(determinant(temp, N-1));  
        }  
    }  
}  
  
private static String calculate(int [][] k, int[] p){  
    int[] res = new int[3];  
    for(int i=0; i<3; i++){  
        int temp=0;  
        for(int j=0; j<3; j++){  
            temp+=k[i][j]*p[j];  
        }  
        res[i] = temp%26;  
    }  
    String encoded="";  
    for(int i=0; i<3; i++){  
        encoded += (char) (res[i] + 'a');  
    }  
    return encoded;  
}  
  
public static void main(String[] args) {  
    String k="cipherng";
```

```
String p="ram";
System.out.println("Value of k: "+k);
System.out.println("Value of plain text: "+p);
int[][] k_arr = new int[3][3];
int[] p_arr = new int[3];
int t=0;
for(int i=0; i<3; i++){
    for(int j=0; j<3; j++){
        k_arr[i][j] = k.charAt(t++) - 'a';
    }
}
t=0;
for(int i=0; i<3; i++){
    p_arr[i] = p.charAt(t++) - 'a';
}

String encoded = calculate(k_arr, p_arr);

System.out.println("Encoded Value: "+encoded);
int[][] kinv = new int[3][3];
int d = determinant(k_arr, 3);
int dinv = help(d, 26);
adjoint(k_arr, kinv);
for(int i=0; i<3; i++){
    for (int j = 0; j < 3; j++) {
        kinv[i][j] = (kinv[i][j]*dinv)%26;
        if(kinv[i][j]<0) kinv[i][j]+=26;
    }
}
t=0;
int[] res = new int[3];
for(int i=0; i<3; i++){
    res[i] = encoded.charAt(t++) - 'a';
}
String decoded = calculate(kinv, res);
System.out.println("Decoded Value: "+decoded);
}
}
```

Output:



The screenshot shows a Java IDE with a code editor and a console window. The code editor displays the `main` method, which initializes `k="ciphering"` and `p="ram"`, prints their values, and then performs the encoding and decoding process. The console window shows the output of the program, which matches the expected results: "Value of k: ciphering", "Value of plain text: ram", "Encoded Value: gla", and "Decoded Value: ram". The process finished with exit code 0.

```
public static void main(String[] args) {
    String k="ciphering";
    String p="ram";
    System.out.println("Value of k: "+k);
    System.out.println("Value of plain text: "+p);
    int[][] k_arr = new int[3][3];
    int[] p_arr = new int[3];
    int t=0;
    for(int i=0; i<3; i++){
        for(int j=0; j<3; j++){
            k_arr[i][j] = k.charAt(t++) - 'a';
        }
    }
    t=0;
    for(int i=0; i<3; i++){
        p_arr[i] = p.charAt(t++) - 'a';
    }

    String encoded = calculate(k_arr, p_arr);

    System.out.println("Encoded Value: "+encoded);
    int[][] kinv = new int[3][3];
    int d = determinant(k_arr, 3);
    int dinv = help(d, 26);
    adjoint(k_arr, kinv);
    for(int i=0; i<3; i++){
        for (int j = 0; j < 3; j++) {
            kinv[i][j] = (kinv[i][j]*dinv)%26;
            if(kinv[i][j]<0) kinv[i][j]+=26;
        }
    }
    t=0;
    int[] res = new int[3];
    for(int i=0; i<3; i++){
        res[i] = encoded.charAt(t++) - 'a';
    }
    String decoded = calculate(kinv, res);
    System.out.println("Decoded Value: "+decoded);
}
```

Value of k: ciphering
Value of plain text: ram
Encoded Value: gla
Decoded Value: ram
Process finished with exit code 0