Ex. No : 1(c) **Date :**	**Hill Cipher**

AIM:

To implement a program to encrypt and decrypt using the Hill cipher substitution technique

ALGORITHM:

1. In the Hill cipher Each letter is represented by a number modulo 26.
2. To encrypt a message, each block of n letters is multiplied by an invertible ***n x n*** matrix, again ***modulus 26***.
3. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.
4. The matrix used for encryption is the cipher key, and it should be chosen randomly from the ***set of invertible n × n matrices (modulo 26).***
5. The cipher can, be adapted to an alphabet with any number of letters.
6. All arithmetic just needs to be done modulo the number of letters instead of modulo 26.

PROGRAM:

HillCipher.java

```
class hillCipher {
   /* 3x3 key matrix for 3 characters at once */
   public static int[][] keymat = new int[][] { { 1, 2, 1 }, { 2, 3, 2 },
          { 2, 2, 1 } }; /* key inverse matrix */
   public static int[][] invkeymat = new int[][] { { -1, 0, 1 }, { 2, -1, 0 }, { -2, 2, -1
} };
   public static String key = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

   private static String encode(char a, char b, char c) {
      String ret = "";
      int x, y, z;
      int posa = (int) a - 65;
      int posb = (int) b - 65;
      int posc = (int) c - 65;
      x = posa * keymat[0][0] + posb * keymat[1][0] + posc * keymat[2][0];
      y = posa * keymat[0][1] + posb * keymat[1][1] + posc * keymat[2][1];
      z = posa * keymat[0][2] + posb * keymat[1][2] + posc * keymat[2][2];
      a = key.charAt(x % 26);
      b = key.charAt(y % 26);
```

```
        c = key.charAt(z % 26);
        ret = "" + a + b + c;
        return ret;
    }

    private static String decode(char a, char b, char c) {
        String ret = "";
        int x, y, z;
        int posa = (int) a - 65;
        int posb = (int) b - 65;
        int posc = (int) c - 65;
        x = posa * invkeymat[0][0] + posb * invkeymat[1][0] + posc *
invkeymat[2][0];
        y = posa * invkeymat[0][1] + posb * invkeymat[1][1] + posc *
invkeymat[2][1];
        z = posa * invkeymat[0][2] + posb * invkeymat[1][2] + posc *
invkeymat[2][2];
        a = key.charAt((x % 26 < 0) ? (26 + x % 26) : (x % 26));

        b = key.charAt((y % 26 < 0) ? (26 + y % 26) : (y % 26));
        c = key.charAt((z % 26 < 0) ? (26 + z % 26) : (z % 26));
        ret = "" + a + b + c;
        return ret;
    }

    public static void main(String[] args) throws java.lang.Exception {
        String msg;
        String enc = "";
        String dec = "";
        int n;
        msg = ("SecurityLaboratory");
        System.out.println("simulation of Hill Cipher\n ------------------------ ");
        System.out.println("Input message : " + msg);
        msg = msg.toUpperCase();
        msg = msg.replaceAll("\\s", "");
        /* remove spaces */ n = msg.length() % 3;
        /* append padding text X */ if (n != 0) {
            for (int i = 1; i <= (3 - n); i++) {
                msg += 'X';
            }
        }
        System.out.println("padded message : " + msg);
        char[] pdchars = msg.toCharArray();
        for (int i = 0; i < msg.length(); i += 3) {
            enc += encode(pdchars[i], pdchars[i + 1], pdchars[i + 2]);
        }
        System.out.println("encoded message : " + enc);
        char[] dechars = enc.toCharArray();
        for (int i = 0; i < enc.length(); i += 3) {
            dec += decode(dechars[i], dechars[i + 1], dechars[i + 2]);
        }
        System.out.println("decoded message : " + dec);
    }
}
```

OUTPUT:

```
Simulating Hill Cipher
-------------------------------
Input Message : SecurityLaboratory
Padded Message : SECURITYLABORATORY
Encrypted Message : EACSDKLCAEFQDUKSXU
Decrypted Message : SECURITYLABORATORY
```

RESULT:

Thus the program for hill cipher encryption and decryption algorithm has been implemented and the output verified successfully.