19SE02IT058

HILL CIPHER :

CODE:

```
#include<iostream>
#include<math.h>
using namespace std;
float en[3][1], de[3][1], a[3][3], b[3][3], msg[3][1], m[3][3];
void getKeyMatrix() { //get key and message from user
  int r, s;
  char mes[3];
  cout<<"Enter 3x3 matrix for key (should have inverse):\n";
  for(r = 0; r < 3; r++)
  for(s = 0; s < 3; s++) {
    cin>>a[r][s];
    m[r][s] = a[r][s];
  }
  cout<<"\nEnter a string of 3 letter(use A through Z): ";
  cin>>mes;
  for(r = 0; r < 3; r++)
  msg[r][0] = mes[r] - 65;
}
void encrypt() { //encrypts the message
  int r, s, k;
  for(r = 0; r < 3; r++)
  for(s = 0; s < 1; s++)
  for(k = 0; k < 3; k++)
  en[r][s] = en[r][s] + a[r][k] * msg[k][s];
  cout<<"\nEncrypted string is: ";
  for(r = 0; r < 3; r++)
  cout<<(char)(fmod(en[r][0], 26) + 65); //modulo 26 is taken for each element of the matrix
obtained by multiplication
```

```
}
void inversematrix() { //find inverse of key matrix
  int r, s, k;
  float p, q;
  for(r = 0; r < 3; r++)
  for(s = 0; s < 3; s++) {
    if(r == s)
      b[r][s]=1;
    else
      b[r][s]=0;
  }
  for(k = 0; k < 3; k++) {
    for(r = 0; r < 3; r++) {
      p = m[r][k];
      q = m[k][k];
      for(s = 0; s < 3; s++) {
        if(r != k) {
          m[r][s] = m[r][s]*q - p*m[k][s];
          b[r][s] = b[r][s]*q - p*b[k][s];
        }
      }
    }
  }
  for(r = 0; r < 3; r++)
  for(s = 0; s < 3; s++)
  b[r][s] = b[r][s] / m[r][r];
  cout<<"\n\nInverse Matrix is:\n";
  for(r = 0; r < 3; r++) {
    for(s = 0; s < 3; s++)
    cout<<b[r][s]<<" ";
    cout<<"\n";
```

```
  }
}
void decrypt() { //decrypt the message
  int r, s, k;
  inversematrix();
  for(r = 0; r < 3; r++)
  for(s = 0; s < 1; s++)
  for(k = 0; k < 3; k++)
  de[r][s] = de[r][s] + b[r][k] * en[k][s];
  cout<<"\nDecrypted string is: ";
  for(r = 0; r < 3; r++)
  cout<<(char)(fmod(de[r][0], 26) + 65); //modulo 26 is taken to get the original message
  cout<<"\n";
}
int main() {
  getKeyMatrix();
  encrypt();
  decrypt();
}
```

OUTPUT:

19SE02IT058