PROGRAM:

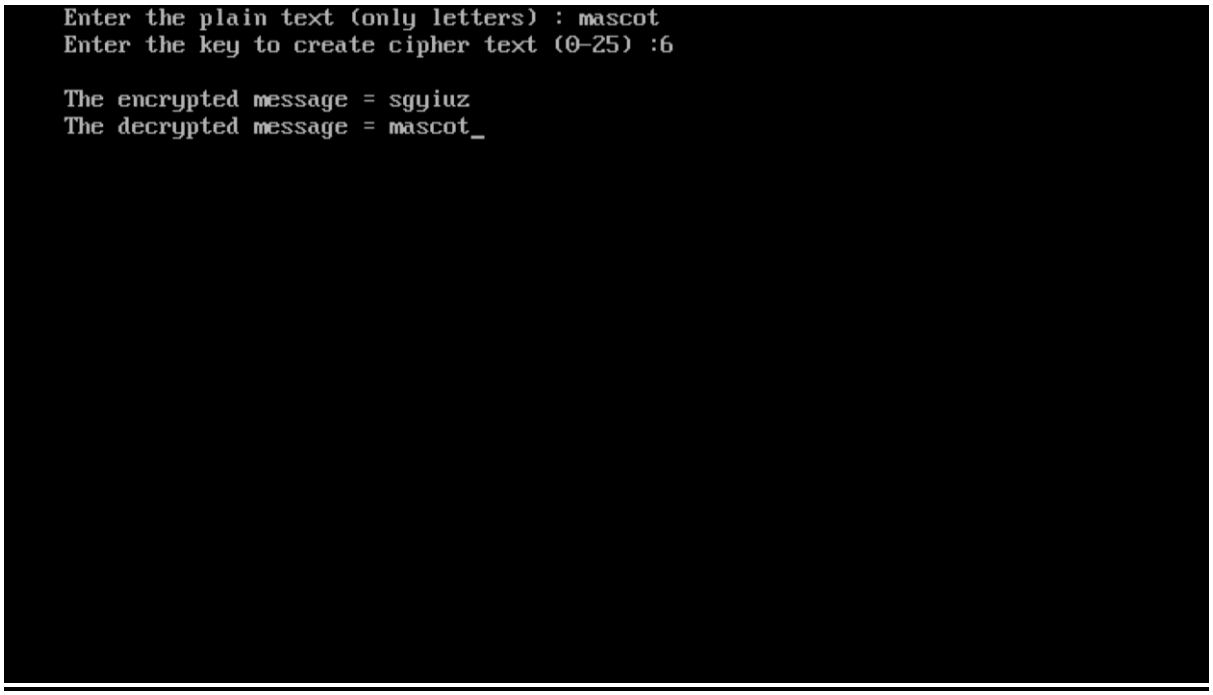
```
#include<stdio.h>
#include<conio.h>
char *encrypt(char *plain,int key)
{
	char cipher[100];
	int i=0,cip,num;
	while(plain[i]!='\0')
	{
		if((plain[i]>='A')&&(plain[i]<='Z'))
		{
			num=plain[i]-'A';
			cip=(num+key)%26;
			cip=cip+'A';
		}
		else if((plain[i]>='a')&&(plain[i]<='z'))
		{
			num=plain[i]-'a';
			cip=(num+key)%26;
			cip=cip+'a';
		}
		cipher[i]=cip;
		i++;

	}
	cipher[i]='\0';
	return cipher;}
```

```
char *decrypt(char *cipher,int key)
{
	char *plain;
	int i=0,cip,num;
	while(cipher[i]!='\0')
	{
		if((cipher[i]>='A')&&(cipher[i]<='Z'))
		{
			num=cipher[i]-'A';
			cip=(num-key)%26;
			if(cip<0)
				cip=cip+26;
			cip=cip+'A';
		}
		else if((cipher[i]>='a')&&(cipher[i]<='z'))
		{
			num=cipher[i]-'a';
			cip=(num-key)%26;
			if(cip<0)
				cip=cip+26;
			cip=cip+'a';
		}
		plain[i]=cip;
		i++;
	}
	plain[i]='\0';
	return plain;
```

```
}
int main()
{
        char message[100];
        int key;
        clrscr();
        printf("Enter the plain text (only letters) : ");
        scanf("%s",message);
        printf("Enter the key to create cipher text (0-25) :");
        scanf("%d",&key);
        printf("\nThe encrypted message = %s",encrypt(message,key));
        printf("\nThe decrypted message = %s",decrypt(encrypt(message,key),key));
        getch();
        return 0;
}
```

OUTPUT:



PROGRAM:

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>


char mat[5][6],message[100],key[10],two[20][3],cipher[100];

int size;


void keygen()

{

   int i=0,j=-1,l,dup=0,k;

   char alp='a';


   for(k=0;key[k]!='\0';k++)

   {

      dup=0;

      for(l=k-1;l>=0;l--)

      {

            if(key[k]==key[l])

            {

                  dup=1;

                  break;

            }

      }

      if(dup==0)

      {
```

```
        if(j>=4)
        {
                i++;
                j=0;
        }
        else
                j++;
        mat[i][j]=key[k];
    }
}
while(alp!=123)
{
        dup=0;
        for(l=0;key[l]!='\0';l++)
        {
                if((alp==key[l])||(alp=='j'))
                {
                        dup=1;
                        alp++;
                        break;
                }
        }
        if(dup==0)
        {
                if(j>=4)
                {
                i++;
```

```
                j=0;
                }
                else
                j++;
                mat[i][j]=alp++;
        }
    }
    for(i=0;i<5;i++)
    {
        for(j=0;j<5;j++)
        printf("%c\t",mat[i][j]);
        printf("\n");
    }

}
void split()
{
        int i,len,k=0;
        len=strlen(message);
        for(i=0;i<len;i=i+2)
        {
                if(message[i]!=message[i+1])
                {
                        two[k][0]=message[i];
                        if(message[i+1]=='\0')
                        two[k][1]='x';
                        else
```

```
                two[k][1]=message[i+1];

            }
            else
            {
                two[k][0]=message[i];
                two[k][1]='x';
                i--;

            }
            two[k][2]='\0';
            k++;
        }
        size=k;
        for(i=0;i<size;i++)
        printf("\nsplit=%s",two[i]);
}
void encrypt()
{
        int i,j,k,m,n,temp1,temp2,x=0,o,p;
        for(k=0;k<size;k++)
        {
            for(i=0;i<5;i++)
            for(j=0;j<5;j++)
            {
                if(two[k][0]==mat[i][j])
                {
```

```
for(m=0;m<5;m++)
{
        if(two[k][1]==mat[i][m])
        {
                n=(j+1)%5;
                cipher[x++]=mat[i][n];
                n=(m+1)%5;
                cipher[x++]=mat[i][n];
                break;
        }
        else if(two[k][1]==mat[m][j])
        {
                n=(i+1)%5;
                cipher[x++]=mat[n][j];
                n=(m+1)%5;
                cipher[x++]=mat[n][j];
                break;
        }
        else
        {
                for(temp2=0;temp2<5;temp2++)
                if(two[k][1]==mat[m][temp2])
                {
                        cipher[x++]=mat[i][temp2];
                        cipher[x++]=mat[m][j];
                        break;
                }
```

```
                                        }
                                    }
                            }
                    }
            }
            cipher[x]='\0';
            printf("\nCipher= %s",cipher);
}

int main()
{
            clrscr();
            printf("\nEnter the message in small case letter ");
            scanf("%s",message);
            printf("\nEnter the key in small case letter");
            scanf("%s",key);
            keygen(key);
            split();
            encrypt();
            getch();
            return 0;

}
```

OUTPUT:

```
Enter key only 4 characters in small letters :
hill

Key in 2*2 matrix
7       8
11      11

 Enter the message in even character: helo

Encrypted String is :drhp
The determinent is 67
The inverse of determinent is 7

Inverse key is
25      22
1       23

Decrypted String is : helo
```

PROGRAM:

```
#include<stdio.h>

#include<math.h>

#include<string.h>


char message[20],key[10],encrypt[20],decrypt[20];

int ke[2][2],sp[10][2],e[10][2],length;

int in[2][2],adj[2][2],d[10][2],esp[10][2];


void encryption();

void decryption();

void getKeyMessage();

void inverse();

void splitmessage();

void splitcipher();

void main()

{

        clrscr();

        getKeyMessage();

        encryption();

        inverse();

        getch();


}


void encryption()

{
```

```
        int i=0,j,k;
        splitmessage();
        for(i=0;i<length;i++)
        for(j=0;j<2;j++)
                e[i][j]=0;
        for(i=0;i<length;i++)
                for(j=0;j<2;j++)
                {
                        for(k=0;k<2;k++)
                                e[i][j]=e[i][j]+ke[j][k]*sp[i][k];
                        e[i][j]=e[i][j]%26;


                }
        printf("\nEncrypted String is :");
        k=0;
        for(i=0;i<length;i++)
        for(j=0;j<2;j++)
                encrypt[k++]=e[i][j]+97;
        encrypt[k]='\0';
        printf("%s",encrypt);
}
void decryption()
{
        int i=0,j,k;
        splitcipher();
        for(i=0;i<length;i++)
        for(j=0;j<2;j++)
```

```
		d[i][j]=0;
	for(i=0;i<length;i++)
	for(j=0;j<2;j++)
	{
		for(k=0;k<2;k++)
			d[i][j]=d[i][j]+in[j][k]*esp[i][k];
		d[i][j]=d[i][j]%26;

	}
	printf("\nDecrypted String is : ");
	k=0;
	for(i=0;i<length;i++)
	for(j=0;j<2;j++)
		decrypt[k++]=d[i][j]+97;
	encrypt[k]='\0';
	printf("%s",decrypt);
}

void getKeyMessage()
{
	int i,j=0,k=0;
	printf("Enter key only 4 characters in small letters :  \n");
	scanf("%s",key);

	for(i=0;i<2;i++)
	for(j=0;j<2;j++)
		ke[i][j]=key[k++]-97;
```

```
        printf("\nKey in 2*2 matrix \n");
        for(i=0;i<2;i++)
        {
                for(j=0;j<2;j++)
                        printf("%d\t",ke[i][j]);
                printf("\n");

        }
        printf("\n Enter the message in even character: ");
        scanf("%s",message);
}

void splitmessage()
{
        int i=0,k=0,j=0;
        while(message[i]!='\0')
        {
                sp[j][k]=message[i++]-97;
                k=(k+1)%2;
                if(k==0)
                        j++;

        }
        length=j;
}

void splitcipher()
```

```
{
        int i=0,j=0,k=0;
        while(encrypt[i]!='\0')
        {
                esp[j][k]=encrypt[i++]-97;
                k=(k+1)%2;
                if(k==0)
                        j++;

        }
}

void inverse()
{
        int i,j,k,det,idet=0;
        det=((ke[0][0]*ke[1][1])-(ke[0][1]*ke[1][0])%26);
        if(det==0)
        {
                printf("Determinent cannot be ZERO");

        }
        else
        {
                if(det<0)
                        det=det+26;
                printf("\nThe determinent is %d\t",det);
```

```
adj[0][0]=ke[1][1];
adj[1][1]=ke[0][0];
adj[0][1]=-ke[0][1]+26;
adj[1][0]=-ke[1][0]+26;

for(i=1;i<26;i++)
if(((det*i)%26)==1)
{
	idet=i;
	printf("\nThe inverse of determinent is %d\n",idet);
	break;
}
if(idet==0)
	printf("SORRY, Inverse is not possible");
else
{
	printf("\nInverse key is \n");
	for(i=0;i<2;i++)
	{
		for(j=0;j<2;j++)
		{
			in[i][j]=(adj[i][j]*idet)%26;
			printf("%d\t",in[i][j]);
		}
		printf("\n");
	}
	decryption();
```

```
            }

        }

}
```

OUTPUT:

```
Enter the key only 4 charactersin small letters :
hill

Key in 2*2 matrix
7       8
11      11

Enter the message in even character : helo

Encrypted String is : drhp
The determinant is 67

The inverse of determinent is 7

Inverse key is
25      22
1       23

Decrypted String is : helo_
```

PROGRAM:

```
#include<stdio.h>

#include<conio.h>

#include<string.h>


char message[100],k[20],key[100],plain[100],cipher[20];


void genkey()

{

	int i,j=0;

	for(i=0;message[i]!='\0';i++)

	{

		if(k[j]!='\0')

			key[i]=k[j++];

		else

		{

			j=0;i--;

		}

	}

	key[i]='\0';

	printf("\nThe generated key to the length of message is %s\n",key);

}

void encryption()

{

	int i=0,cip,num,k;

	while(message[i]!='\0')

	{
```

```
		num=message[i]-'a';
		k=key[i]-'a';
		cip=(num+k)%26;
		cip=cip+'a';
		cipher[i]=cip;
		i++;
	}
	cipher[i]='\0';
	printf("\nThe encrypted message is %s",cipher);

}
void decryption()
{
	int i=0,cip,num,k;
	while(cipher[i]!='\0')
	{
		num=cipher[i]-'a';
		k=key[i]-'a';
		cip=(num-k)%26;
		if(cip<0)
			cip=cip+26;
		cip=cip+'a';
		plain[i]=cip;
		i++;
	}
	plain[i]='\0';
	printf("\nThe decrypted message is %s\n",plain);
```

```
}
int main()
{
        clrscr();
        printf("Enter the plain text (only lower case letters) ");
        scanf("%s",message);
        printf("Enter the key in lower case letter ");
        scanf("%s",k);
        genkey();
        encryption(message,key);
        decryption(cipher,key);
        getch();
        return 0;
}
```

OUTPUT: