Name: Tanmay Vig

Roll Number: 19BCS061

Batch: 5th Sem

Lab 4: Write a program to encrypt and decrypt Plain text using Playfair Cipher

Answer:

Source Code:

```
#include<iostream>

#include<string>

#include<set>


using namespace std;


string encrypt(string text, char grid[5][5]){

    int xi,yi,xj,yj,t=0;

    string cipher,temp;

    // saperating same letters

    while(t<text.length()-1){

        if(text[t]==text[t+1]){

            for(int i=0;i<=t;i++){

                temp+=text[i];

            }

            temp+='x';

            for(int i=t+1;i<text.length();i++){

                temp+=text[i];
```

```
        }
        text=temp;
    }t+=2;
}

if(text.size()%2) text+='z';// making length even

cout<<"Plain Text: "<<text<<endl;
for(int i=0,j=1;i<text.size()-1 && j<text.size();i+=2,j+=2){
    if(text[i]==text[j]) text[j]='x';// replacing for unique value

    // storing positions of both chars
    for(int k=0;k<5;k++){
        for(int l=0;l<5;l++){
            if(grid[k][l]==text[i]){
                xi=k;
                yi=l;
            }else if(grid[k][l]==text[j]){
                xj=k;
                yj=l;
            }
        }
    }

    //substituting values;
    if(xi==xj){
```

```
            yi=(yi+1)%5;
            yj=(yj+1)%5;
            cipher+=grid[xi][yi];
            cipher+=grid[xj][yj];
        }else if(yi==yj){
            xi=(xi+1)%5;
            xj=(xj+1)%5;
            cipher+=grid[yi][yi];
            cipher+=grid[xj][yj];
        }else{
            cipher+=grid[xi][yj];
            cipher+=grid[xj][yi];
        }
    }
    return cipher;
}

string decrypt(string text,char grid[5][5]){
    int xi,yi,xj,yj,t=0;
    string cipher,temp;
    for(int i=0,j=1;i<text.size()-1 && j<text.size();i+=2,j+=2){
        if(text[i]==text[j]) text[j]='x';// replacing for unique value

        // storing positions of both chars
        for(int k=0;k<5;k++){
            for(int l=0;l<5;l++){
```

```
            if(grid[k][l]==text[i]){
                xi=k;
                yi=l;
            }else if(grid[k][l]==text[j]){
                xj=k;
                yj=l;
            }
        }
    }

    //substituting values;
    if(xi==xj){
        yi=(yi+4)%5;
        yj=(yj+4)%5;
        cipher+=grid[xi][yi];
        cipher+=grid[xj][yj];
    }else if(yi==yj){
        xi=(xi-1) + (xi>1) ? 0 : 5;
        xj=(xj-1) + (xj>1) ? 0 : 5;
        cipher+=grid[yi][yi];
        cipher+=grid[xj][yj];
    }else{
        cipher+=grid[xi][yj];
        cipher+=grid[xj][yi];
    }
}
```

```
    return cipher;
}

int main(){
    char grid[5][5];
    string text, key,keyun="";
    cout<<"Enter Plain Text\t";
    cin>>text;
    cout<<"Enter key\t";
    cin>>key;
    // removing repetitions in key
    for(int i=0;i<key.length();i++){
        bool ad = true;
        for(int j=0; j<keyun.size();j++){
            if(key[i]==keyun[j]){
                ad=false;
                break;
            }
        }
        if(ad) keyun+=key[i];
    }

    cout<<"Key without Repetitions: "<<keyun<<endl;

    //making grid
    set<char> repeat;
```

```
char f='a';
for(int i=0;i<5;i++){
    for(int j=0;j<5;j++){
        if((i*5+j)<keyun.size()){
            grid[i][j]=keyun[i*5+j];
            repeat.insert(keyun[i*5+j]);
        }else{
            if(repeat.find(f)==repeat.end() && f!='j'){
                grid[i][j]=f;
                repeat.insert(f++);
            }else{
                f++;
                j--;
            }
        }
    }
}
//printing key grid
for(int i=0;i<5;i++){
    for(int j=0;j<5;j++){
        cout<<grid[i][j]<<" ";
    }
    cout<<endl;
}
string enc = encrypt(text,grid);
cout<<"\nEncrypted Text: "<<enc<<endl;
```

```
    cout<<"\nDo you wand to Decrypt??\nIf yes then enter 1 else 0\n";
    int inp=0;
    cin>>inp;
    if(inp){
        cout<<"\nDecrypted Text: "<<decrypt(enc,grid)<<endl;
    }
    return 0;
}
```

Output: