

**EXPNO:2**

**DATE:**

## **PLAYFAIRCIPHER**

**Aim:** To implement an encryption algorithm using Playfair Cipher technique.

### **Algorithm:**

- Step1: "Algorithm" (as the key) and "ulroaliocvrx" (as the encrypted text).
- Step2: Remove spaces and convert to lowercase.
- Step3: Create a 5x5 key table based on the modified key.
- Step4: Apply Playfair Cipher decryption to the encrypted text using the generated key table.
- Step5: Display the deciphered text.

### **Program:**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h> #define
SIZE 30
void toLowerCase(char plain[], int ps)
{
    int i;
    for (i = 0; i < ps; i++) {
        if (plain[i] > 64 && plain[i] < 91)
            plain[i] += 32;
    }
}
int removeSpaces(char* plain, int ps)
{
    int i, count = 0;
    for (i = 0; i < ps; i++)    if
(plain[i] != " )
plain[count++] = plain[i];
```

```

plain[count] = '\0';    return
count;
}
void generateKeyTable(charkey[],intks,charkeyT[5][5])
{
    inti,j,k,flag=0, *dicty;
    dicty=(int*)calloc(26,sizeof(int));

    for(i =0;i<ks;i++)
    {
        if(key[i]!='j')
        dicty[key[i]-97]=2;
    }
    dicty['j'-97]=1;
    i =0;
    j=0;
    for(k =0;k<ks; k++)
    {
        if(dicty[key[k]-97]==2)
        {
            dicty[key[k]-97]-=1;
            keyT[i][j]=key[k]; j++;
            if(j ==5)
            {
                i++;
                j=0;
            }
        }
    }
    for(k =0;k<26;k++)
    {
        if(dicty[k]==0)
        {
            keyT[i][j]=(char)(k+
97);
            j++;
            if(j ==5)
            {
                i++;
                j =0;
            }
        }
    }
}

```

```

    }
}
}
voidsearch(charkeyT[5][5],chara,charb,intarr[])
{
    inti,j;    if (a
    =='j')      a='i';
    elseif(b=='j') b
    = 'i';

    for(i = 0;i<5;i++)
    {
        for(j =0; j<5;j++)
        {
            if(keyT[i][j]==a)
            {
                arr[0]=i;
arr[1]=j;
            }
            elseif(keyT[i][j]==b){
                arr[2] = i;
arr[3]=j;
            }
        }
    }
}intmod5(inta)
{
    if(a<0)
a += 5;    return
(a % 5);
}
voiddecrypt(charstr[],charkeyT[5][5],intps){ int
    i, a[4];    for (i = 0; i < ps; i += 2)
    {
        search(keyT,str[i],str[i+1],a);
if(a[0]==a[2]){
        str[i]=
keyT[a[0]][mod5(a[1]-1)];
        str[i+1]=keyT[a[0]][mod5(a[3]-1)];
    }
}

```

```

        elseif(a[1]==a[3]){          str[i]=
keyT[mod5(a[0]-1)][a[1]];
        str[i+1]=keyT[mod5(a[2]- 1)][a[1]];
    }    else{          str[i]
=keyT[a[0]][a[3]];
        str[i+1]=keyT[a[2]][a[1]];
    }
}
}

```

```

voiddecryptByPlayfairCipher(charstr[],charkey[])
{    char ps, ks, keyT[5][5];
ks = strlen(key);    ks =
removeSpaces(key, ks);
toLowerCase(key,ks);    ps=
strlen(str);    toLowerCase(str,
ps);ps= removeSpaces(str, ps);

```

```

    generateKeyTable(key,ks,keyT);

```

```

    decrypt(str,keyT, ps);
}

```

```

intmain()

```

```

{
    charstr[SIZE],key[SIZE];

```

```

    strcpy(key,"SRIPRASATH");
printf("Key text: %s\n", key);
strcpy(str, "ulroaliocvrx");
printf("Plain text: %s\n", str);

```

```

    decryptByPlayfairCipher(str, key);

```

```
printf("Decipheredtext:%s\n",str);  
  
return0;  
}
```

### **Output:**

```
Key text: thriloke.....  
Plain text: superman.....  
Cipher text: unqklfos.....  
  
=== Code Execution Successful ===
```

### **Result:**