Ex. No.: 3

Date:

PLAYFAIR CIPHER

Problem Statement:

The Playfair cipher was the first practical digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher. In playfair cipher unlike traditional cipher we encrypt a pair of alphabets (digraphs) instead of a single alphabet.

.

The Algorithm consists of 2 steps:

I. **Generate the key Square(5×5):**
   - The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I.
   - The initial alphabets in the key square are the unique alphabets of the key in the order in which

II. **Algorithm to encrypt the plain text:** The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.

Rules for Encryption:

- **If both the letters are in the same column**: Take the letter below each one (going back to the top if at the bottom).
- **If both the letters are in the same row**: Take the letter to the right of each one (going back to the leftmost if at the rightmost position).
- **If neither of the above rules is true**: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Aim:

To implement Playfair Cipher technique using C.

Algorithm:

1. Initialize the contents of the table to zero.
2. Get the length of the key
3. Get the key string from the user.
4. Insert each element of the key into the table.
5. Fill the remaining entries of the table with the character not already entered into the table.
6. Enter the length of the plaintext.
7. Get the plaintext string.

Program Code:

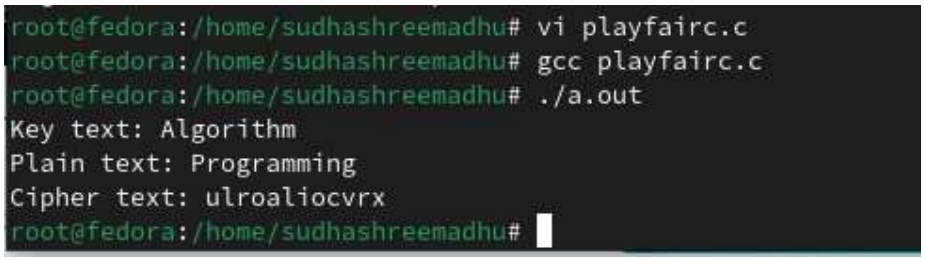
```
#include<stdio.h>
int check(char table[5][5],char k)
{
	int i,j;
	for(i=0;i<5;++i)
	for(j=0;j<5;++j)
	{
		if(table[i][j]==k)
			return 0;
	}
	return 1;
}

void main()
{
	int i,j,key_len;
	char table[5][5];
	for(i=0;i<5;++i)
	    for(j=0;j<5;++j)
		table[i][j]='0';
printf("**********Playfair Cipher**********\n\n");
printf("Enter the length of the Key. ");
scanf("%d",&key_len);

char key[key_len];
printf("Enter the Key. ");
for(i=-1;i<key_len;++i)
{
	scanf("%c",&key[i]);
	if(key[i]=='j')
		key[i]='i';
}
int flag;
int count=0;
// inserting the key into the
table for(i=0;i<5;++i)
{
	for(j=0;j<5;++j)
	{
		flag=0;
		while(flag!=1)
		{
			if(count>key_len)
				goto l1;
			flag=check(table,key[count]);
			++count;
		}// end of while
		table[i][j]=key[(count-1)];
	}// end of inner for
}// end of outer for

l1:printf("\n");
```

```
int val=97;

//inserting other alphabets
for(i=0;i<5;++i)
{
  for(j=0;j<5;++j)
  {
          if(table[i][j]>=97 && table[i][j]<=123)
          {}
          else
          {
                  flag=0;
                  while(flag!=1)
                  {
                          if('j'==(char)val)
                                  ++val;
                          flag=check(table,(char)val);
                          ++val;
                  }// end of while
                  table[i][j]=(char)(val-1);
          }//end of else
  }// end of inner for
}// end of outer for

printf("The table is as follows:\n");
for(i=0;i<5;++i)
{
  for(j=0;j<5;++j)
  {
          printf("%c ",table[i][j]);
  }
printf("\n");
}

int l=0;
printf("\nEnter the length of plain text.(without spaces) ");
scanf("%d",&l);
printf("\nEnter the Plain text. ");
char p[l];
for(i=-1;i<l;++i)
{
          scanf("%c",&p[i]);
}

for(i=-1;i<l;++i)
{
          if(p[i]=='j')
                  p[i]='i';
}

printf("\nThe replaced text(j with i)");
for(i=-1;i<l;++i)
printf("%c ",p[i]);
count=0;
for(i=-1;i<l;++i)
```

```
{
if(p[i]==p[i+1])
count=count+1;
}
printf("\nThe cipher has to enter %d bogus char.It is either 'x' or 'z'\n",count);
int length=0;
if((l+count)%2!=0)
length=(l+count+1);
else
length=(l+count);
printf("\nValue of length is %d.\n",length);
char p1[length];

//inserting bogus characters.
char temp1;
int count1=0;
for(i=-1;i<l;++i)
{
p1[count1]=p[i];
if(p[i]==p[i+1])
{
        count1=count1+1;
        if(p[i]=='x')
                p1[count1]='z';
        else    p1[count1]='x';
}
count1=count1+1;
}
//checking for length
char bogus;
if((l+count)%2!=0)
{
if(p1[length-1]=='x')
p1[length]='z';
else
p1[length]='x';
}

printf("The final text is:");
for(i=0;i<=length;++i)
printf("%c ",p1[i]);
char cipher_text[length];
int r1,r2,c1,c2;
int k1;
for(k1=1;k1<=length;++k1)
{
for(i=0;i<5;++i)
{
for(j=0;j<5;++j)
{
if(table[i][j]==p1[k1])
{
r1=i;
c1=j;
}
```

```
else
if(table[i][j]==p1[k1+1])
{
r2=i;
c2=j;
}
}//end of for with j
}//end of for with i
(r1==r2)
{
cipher_text[k1]=table[r1][(c1+1)%5];
cipher_text[k1+1]=table[r1][(c2+1)%5];
}
else
if(c1==c2)
{
cipher_text[k1]=table[(r1+1)%5][c1];
cipher_text[k1+1]=table[(r2+1)%5][c1];
}
else
{
cipher_text[k1]=table[r1][c2];
cipher_text[k1+1]=table[r2][c1];
}

k1=k1+1;
}//end of for with k1

printf("\n\nThe Cipher text is:\n ");
for(i=1;i<=length;++i)
printf("%c ",cipher_text[i]);
}
```

Output:



Result: