**Information Security**

* **Practical-3: Playfair Cipher**

**CODE:**

#include<stdio.h>

#include<string.h>

#include<ctype.h>

int removerepeated(int size,int a[]);

int insertelementat(int position,int a[],int size);

main()

{

int i,j,k,numstr[100],numcipher[100],numkey[100],lenkey,templen,tempkey[100],flag=-1,size,cipherkey[5][5],lennumstr,row1,row2,col1,col2;

char str[100],key[100];

printf("Enter a string : ");

gets(str);

for(i=0,j=0;i<strlen(str);i++)

{

if(str[i]!=' ')

{

str[j]=toupper(str[i]);

j++;

}

}

printf("\n---------------------------\n");

str[j]='\0';

printf("Entered String is %s",str);

size=strlen(str);

for(i=0;i<size;i++)

{

if(str[i]!=' ')

numstr[i]=str[i]-'A';

}

lennumstr=i;

printf("\n---------------------------\n");

printf("Enter the key : ");

gets(key);

for(i=0,j=0;i<strlen(key);i++)

{

if(key[i]!=' ')

{

key[j]=toupper(key[i]);

j++;

}

}

key[j]='\0';

printf(" Entered key is %s\n",key);

k=0;

for(i=0;i<strlen(key)+26;i++)

{

if(i<strlen(key))

{

if(key[i]=='J')

{

flag=8;

printf("%d",flag);

}

numkey[i]=key[i]-'A';

}

else

{

if(k!=9 && k!=flag)

{

numkey[i]=k;

}

k++;

}

}

templen=i;

lenkey=removerepeated(templen,numkey);

printf("\n---------------------------\n");

printf("Entered key converted according to Play Fair Cipher rule\n");

for(i=0;i<lenkey;i++)

{

printf("%c",numkey[i]+'A');

}

printf("\n");

k=0;

for(i=0;i<5;i++)

{

for(j=0;j<5;j++)

{

cipherkey[i][j]=numkey[k];

k++;

}

}

printf("\n---------------------------\n");

printf("Arranged key\n");

for(i=0;i<5;i++)

{

for(j=0;j<5;j++)

{

printf("%c |",cipherkey[i][j]+'A');

}

printf("\n");

}

for(i=0;i<lennumstr;i+=2)

{

if(numstr[i]==numstr[i+1])

{

insertelementat(i+1,numstr,lennumstr);

lennumstr++;

}

}

if(lennumstr%2!=0)

{

insertelementat(lennumstr,numstr,lennumstr);

lennumstr++;

}

printf("\n---------------------------\n");

printf("Entered String/Message After Processing according to Play fair cipher rule\n");

for(i=0;i<lennumstr;i++)

{

printf("%c",numstr[i]+'A');

}

for(k=0;k<lennumstr;k+=2)

{

for(i=0;i<5;i++)

{

for(j=0;j<5;j++)

{

if(numstr[k]==cipherkey[i][j])

{

row1=i;

col1=j;

}

if(numstr[k+1]==cipherkey[i][j])

{

row2=i;

col2=j;

}

}

}

if(row1==row2)

{

col1=(col1-1)%5;

col2=(col2-1)%5;

if(col1<0)

{

col1=5+col1;

}

if(col2<0)

{

col2=5+col2;

}

numcipher[k]=cipherkey[row1][col1];

numcipher[k+1]=cipherkey[row2][col2];

}

if(col1==col2)

{

row1=(row1-1)%5;

row2=(row2-1)%5;

if(row1<0)

{

row1=5+row1;

}

if(row2<0)

{

row2=5+row2;

}

numcipher[k]=cipherkey[row1][col1];

numcipher[k+1]=cipherkey[row2][col2];

}

if(row1!=row2&&col1!=col2)

{

numcipher[k]=cipherkey[row1][col2];

numcipher[k+1]=cipherkey[row2][col1];

}

}

printf("\n---------------------------\n");

printf("\nCipher Text is\n");

for(i=0;i<lennumstr;i++)

{

if((numcipher[i]+'A')!='X')

printf("%c",numcipher[i]+'A');

}

printf("\n");

}

int removerepeated(int size,int a[])

{

int i,j,k;

for(i=0;i<size;i++)

{

for(j=i+1;j<size;)

{

if(a[i]==a[j])

{

for(k=j;k<size;k++)

{

a[k]=a[k+1];

}

size--;

}

else

{

j++;

}

}

}

return(size);

}

int insertelementat(int position,int a[],int size)

{

int i,insitem=23,temp[size+1];

for(i=0;i<=size;i++)

{

if(i<position)

{

temp[i]=a[i];

}

if(i>position)

{

temp[i]=a[i-1];

}

if(i==position)

{

temp[i]=insitem;

}

}

for(i=0;i<=size;i++)

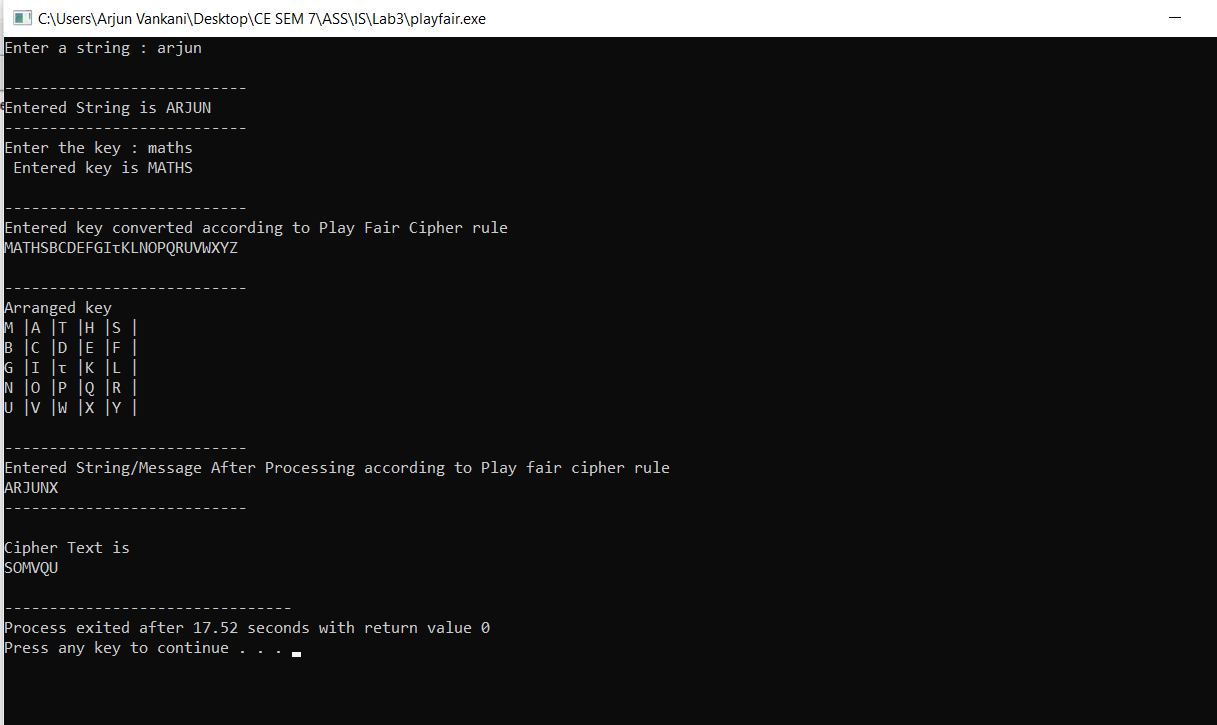
{

a[i]=temp[i];

}

}

**Output:**

****