

Q5

Pratyush Singh
1022750140

Q5

```
Ans # use 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];
    return count;
}
```

```
void generateKeyTable (char key[], int ks,  
                      char keyT[5][5])
```

```
{
```

```
    int i, j, k, flag = 0, *dirty;
```

```
    dirty = (int *) calloc(26, size of(int));
```

```
    for (i = 0; i < ks; i++) {
```

```
        if (key[i] != 'j')
```

```
            dirty[key[i] - 97] = 2;
```

```
    }
```

```
        dirty['j' - 97] = 1;
```

```
    i = 0;  
    j = 0;
```

```
    for (k = 0; k < ks; k++) {
```

```
        if (dirty[key[k] - 97] == 2)
```

```
        {
```

```
            dirty[key[k] - 97] -= 1;
```

```
            keyT[i][j] = key[k];
```

```
            j++;
```

```
            if (j == 5) {
```

```
                i++;
```

```
                j = 0;
```

```
            }
```

```
    }
```

```

for (k=0; k<26; k++){
    if (dirty[k]==0)
    {
        key T[i][j] = (char) (k+97);
        j++;
        if (j==5){
            i++;
            j=0;
        }
    }
}
}
}

```

```

void search (char key T[5][5], char a, char b,
             int arr[])

```

```

{
    int i, j;

```

```

    if (a=='i')

```

```

        a='i';

```

```

    else if (b=='j')

```

```

        b='i';

```

```

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

```

```
{ for (i = 0; j < 5; j++) {
```

```
    if (key T[i][j] == a)
```

```
    {
```

```
        arr[0] = i;
```

```
        arr[1] = j;
```

```
    }
```

```
    else key if (key T[i][j] == b)
```

```
    {
```

```
        arr[2] = i;
```

```
        arr[3] = j;
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
int mod5 (int a)
```

```
{
```

```
    return (a % 5);
```

```
}
```

```
int prepare (char str[], int ptrs)
```

```
{
```

```
    if (ptrs % 2 != 0) {
```

```
        str[ptrs++] = 'Z';
```

```
        str[ptrs] = '\0';
```

```
}
```


return str;

}

void encrypt (char str[], char keyT[5],
int ps)

{ 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[mod5(a[2]+1)][a[1]];

}

else {

str[i] = keyT[a[0]][a[3]];

str[i+1] = keyT[a[2]][a[1]];

}

}

}

void encryptByPlayfairCipher (char str[],
char key[] char key[])

{

```
char ps, ks, keyT[5][5];
```

```
key
```

```
ks = stolen(key);
```

```
ks = removeSpace(key, ks);
```

```
toLowercase(key, ks);
```

```
ps = stolen(ster);
```

```
toLowercase(ster, ps);
```

```
ps = prepare(ster, ps);
```

```
generateKeyTable(key, ks, keyT);
```

```
encrypt(ster, keyT, ps);
```

```
}
```

```
int main()
```

```
{ char str[Size], key[Size];
```

```
strcpy(key, "Monarchy");
```

```
printf("key test: %s\n", key);
```

```
strcpy(ster, "instruments");
```

```
printf("Plain text: %s\n", ster);
```

```
encryptByPlayfairCipher(ster, key);
```

```
printf("cipher test: %s\n", ster);
```

```
return 0;
```

```
}
```