

Walchand College Of Engineering, Sangli
Department of Computer Science and Engineering
Subject: C&NS Lab

Batch: B4

Name: Gayatri Sopan Gade

PRN:2020BTECS00210

Assignment 5

Title: Implementation of Transposition Cipher

Introduction:

Transposition Cipher is a cryptographic algorithm where the order of alphabets in the plaintext is rearranged to form a cipher text.

Two types of Transposition cipher:

- Rail Fence cipher.
- Columnar Cipher

1.Rail Fence Cipher:

The rail fence cipher (also called a zigzag cipher) is a form of transposition cipher. It derives its name from the way in which it is encoded.

Example:

Encryption

Input : "attack at once"

Key = 2

Output : atc toctaka ne

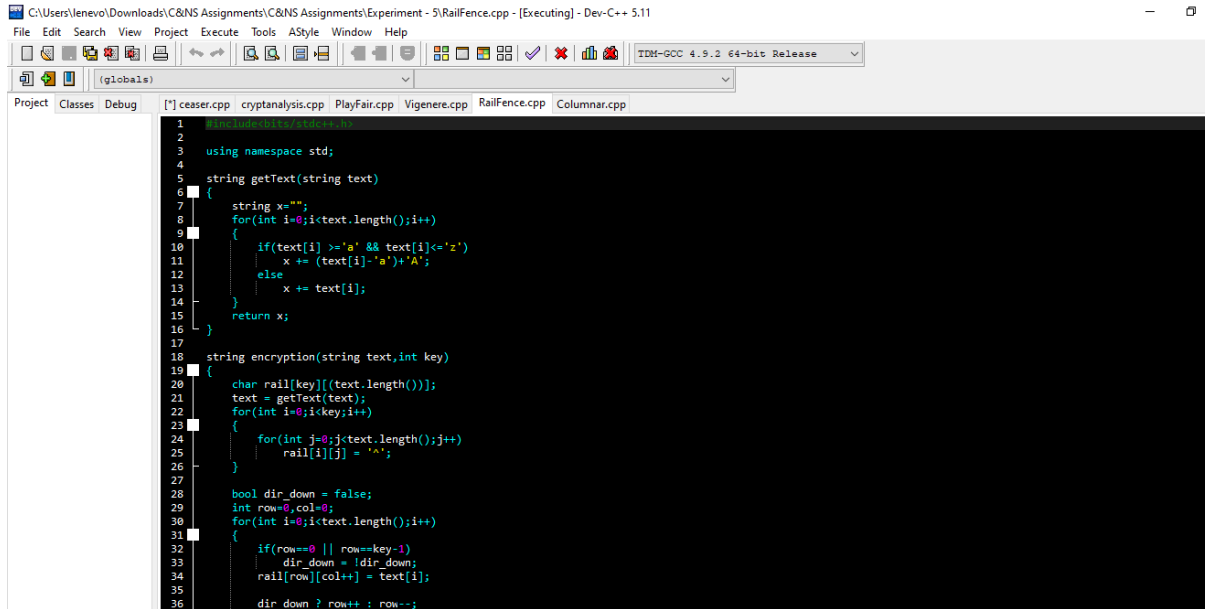
Decryption

Input : "atc toctaka ne"

Key = 2

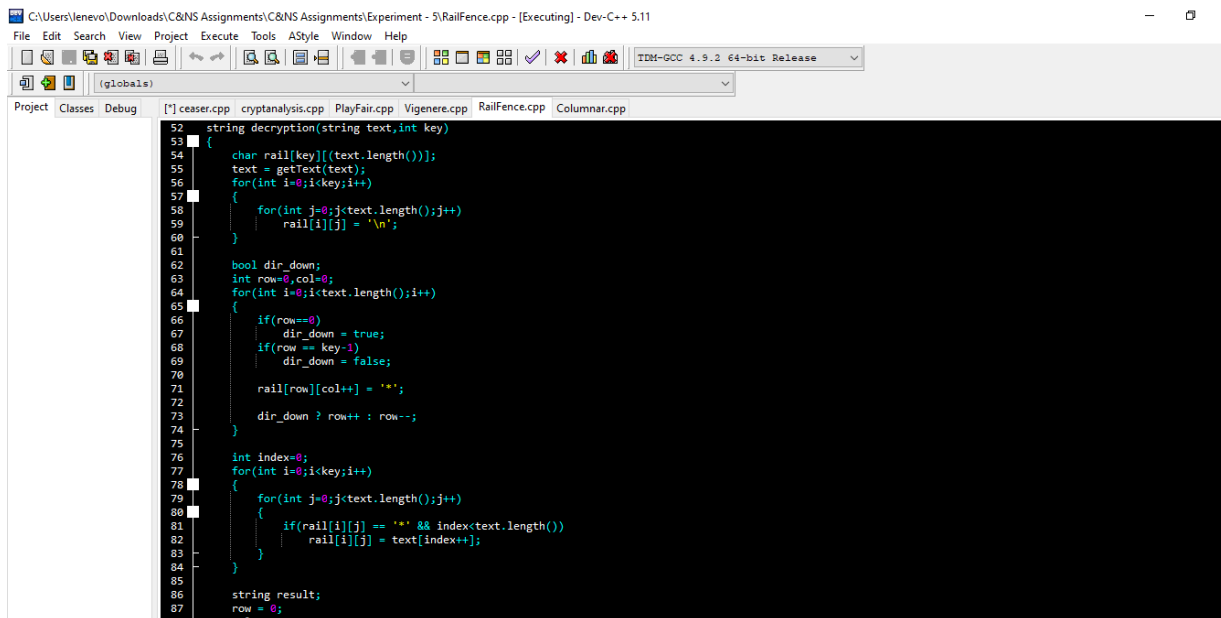
Output : attack at once

Code Screenshot:



The screenshot shows a C++ IDE with the file `RailFence.cpp` open. The code defines a `getText` function that converts lowercase letters to uppercase and a `encryption` function that implements the Rail Fence cipher. The `encryption` function uses a `rail` array to store characters in a zig-zag pattern based on the key.

```
1 #include<bits/stdc++.h>
2
3 using namespace std;
4
5 string getText(string text)
6 {
7     string x="";
8     for(int i=0;i<text.length();i++)
9     {
10         if(text[i] >='a' && text[i]<='z')
11             x += (text[i]-'a')+'A';
12         else
13             x += text[i];
14     }
15     return x;
16 }
17
18 string encryption(string text,int key)
19 {
20     char rail[key][text.length()];
21     text = getText(text);
22     for(int i=0;i<key;i++)
23     {
24         for(int j=0;j<text.length();j++)
25             rail[i][j] = 'A';
26     }
27
28     bool dir_down = false;
29     int row=0,col=0;
30     for(int i=0;i<text.length();i++)
31     {
32         if(row==0 || row==key-1)
33             dir_down = !dir_down;
34         rail[row][col++] = text[i];
35
36         dir_down ? row++ : row--;
```



This screenshot shows the continuation of the `encryption` function. It completes the zig-zag filling of the `rail` array and then constructs the final encrypted string by reading the characters row by row.

```
52     string decryption(string text,int key)
53     {
54         char rail[key][text.length()];
55         text = getText(text);
56         for(int i=0;i<key;i++)
57         {
58             for(int j=0;j<text.length();j++)
59                 rail[i][j] = '\n';
60         }
61
62         bool dir_down;
63         int row=0,col=0;
64         for(int i=0;i<text.length();i++)
65         {
66             if(row==0)
67                 dir_down = true;
68             if(row == key-1)
69                 dir_down = false;
70
71             rail[row][col++] = text[i];
72
73             dir_down ? row++ : row--;
74         }
75
76         int index=0;
77         for(int i=0;i<key;i++)
78         {
79             for(int j=0;j<text.length();j++)
80             {
81                 if(rail[i][j] == "" && index<text.length())
82                     rail[i][j] = text[index++];
83             }
84         }
85
86         string result;
87         row = 0;
88         col = 0;
```

```
154 cout << "Data is from\n 1. Manual Entering\n 2. File\nEnter Choice: ";
155 cin>>datachoice;
156 if(datachoice == 1)
157 {
158     cout<<"Enter data to be Encrypted:\n";
159     cin.ignore();
160     getline(cin,sample);
161     cout<<"Enter the key: ";
162     cin>>key;
163     cout<<"Encrypted String:\n";
164     cout<<encryption(sample,key)<<endl;;
165 }
166 else
167 {
168     cout<<"Enter File Name:\n";
169     cin.ignore();
170     getline(cin,sample);
171     cout<<"Enter the key: ";
172     cin>>key;
173     fstream myfile;
174     myfile.open(sample.c_str());
175     string str,s;
176     if(!myfile.is_open())
177     {
178         cout << "Error while Opening File";
179         while(getline(myfile,str))
180             s+=str;
181         myfile.close();
182         s=decryption(s,key);
183         myfile.open("Plaintext.txt",ios_base::out);
184         if(myfile.is_open())
185             myfile.write(s.data(),s.size());
186         cout<<"File Decrypted\n";
187         myfile.close();
188     }
189     break;
```

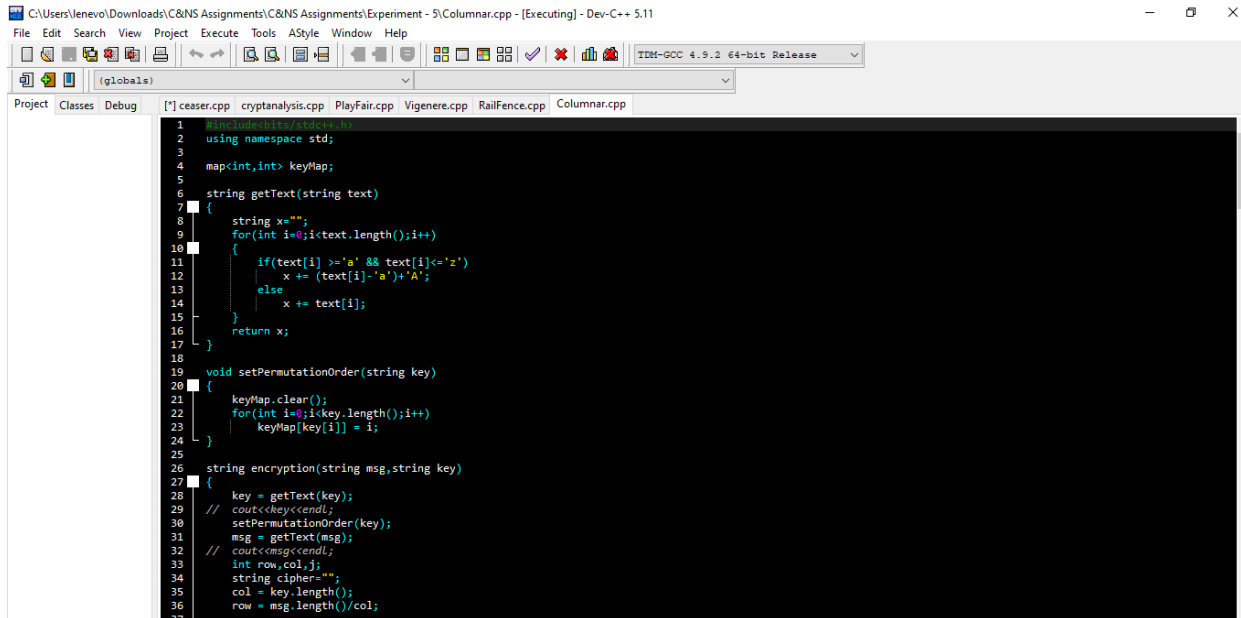
Output:

```
3. Exit
Enter Choice: 1
Data is from
 1. Manual Entering
 2. File
Enter Choice: 1
Enter data to be Encrypted:
Gayatri
Enter the key: 2
Encrypted String:
GYTIAAR
Rail Fence Cipher
 1. Encryption
 2. Decryption
 3. Exit
Enter Choice: 2
Data is from
 1. Manual Entering
 2. File
Enter Choice: 1
Enter data to be Decrypted:
GYTIAAR
Enter the key: 2
Decrypted String:
GAYATRI
Rail Fence Cipher
 1. Encryption
 2. Decryption
 3. Exit
Enter Choice:
```

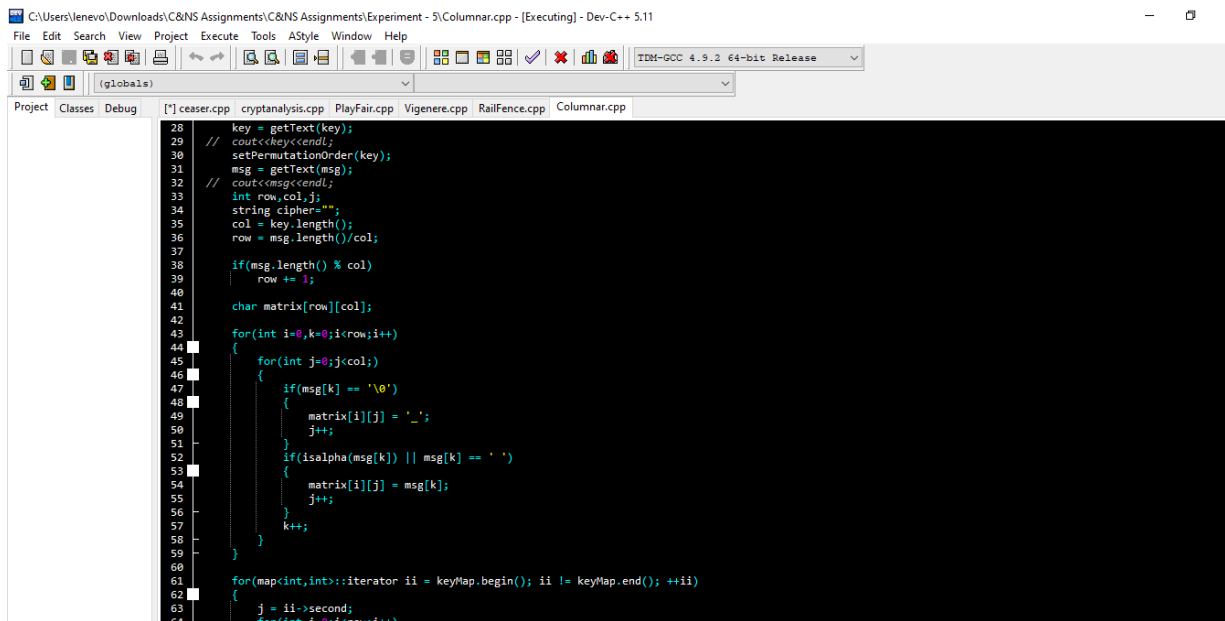
Columnar Transposition:

The Columnar Transposition Cipher is a form of transposition cipher just like Rail Fence Cipher. Columnar Transposition involves writing the plaintext out in rows, and then reading the ciphertext off in columns one by one.

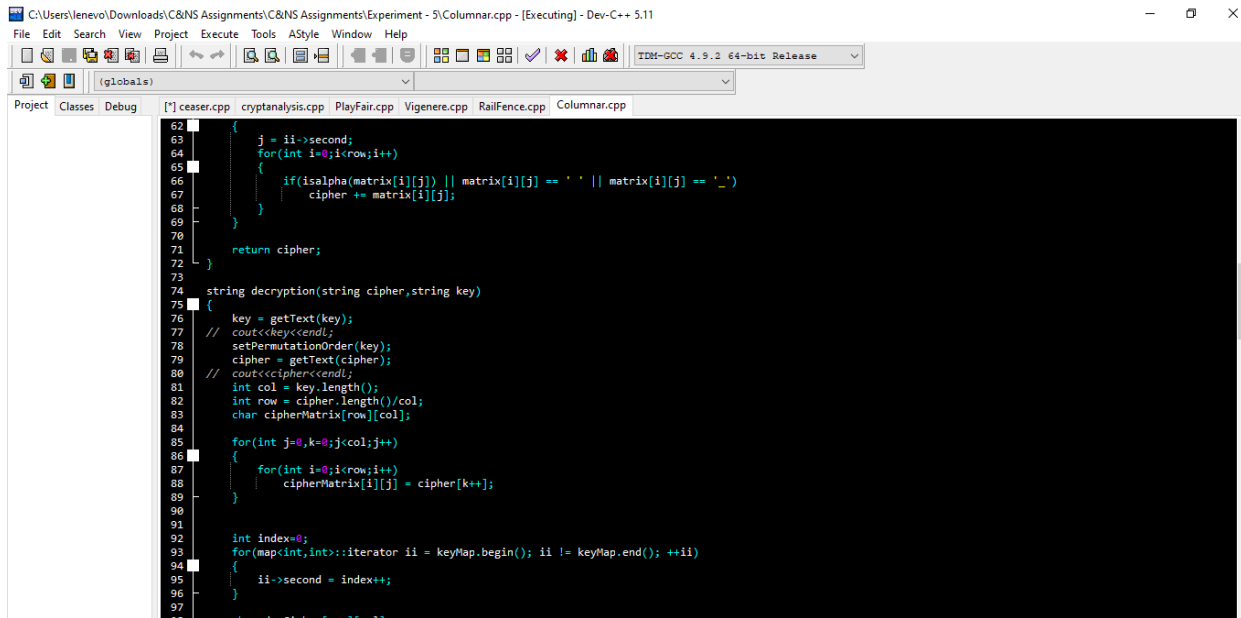
Code Screenshots:



```
1 // Columnar.cpp
2 using namespace std;
3
4 map<int,int> keyMap;
5
6 string getText(string text)
7 {
8     string x="";
9     for(int i=0;i<text.length();i++)
10     {
11         if(text[i] >='a' && text[i] <='z')
12             x += (text[i]-'a')+'A';
13         else
14             x += text[i];
15     }
16     return x;
17 }
18
19 void setPermutationOrder(string key)
20 {
21     keyMap.clear();
22     for(int i=0;i<key.length();i++)
23         keyMap[key[i]] = i;
24 }
25
26 string encryption(string msg,string key)
27 {
28     key = getText(key);
29     // cout<<key<<endl;
30     setPermutationOrder(key);
31     msg = getText(msg);
32     // cout<<msg<<endl;
33     int row,col,j;
34     string cipher="";
35     col = key.length();
36     row = msg.length()/col;
```

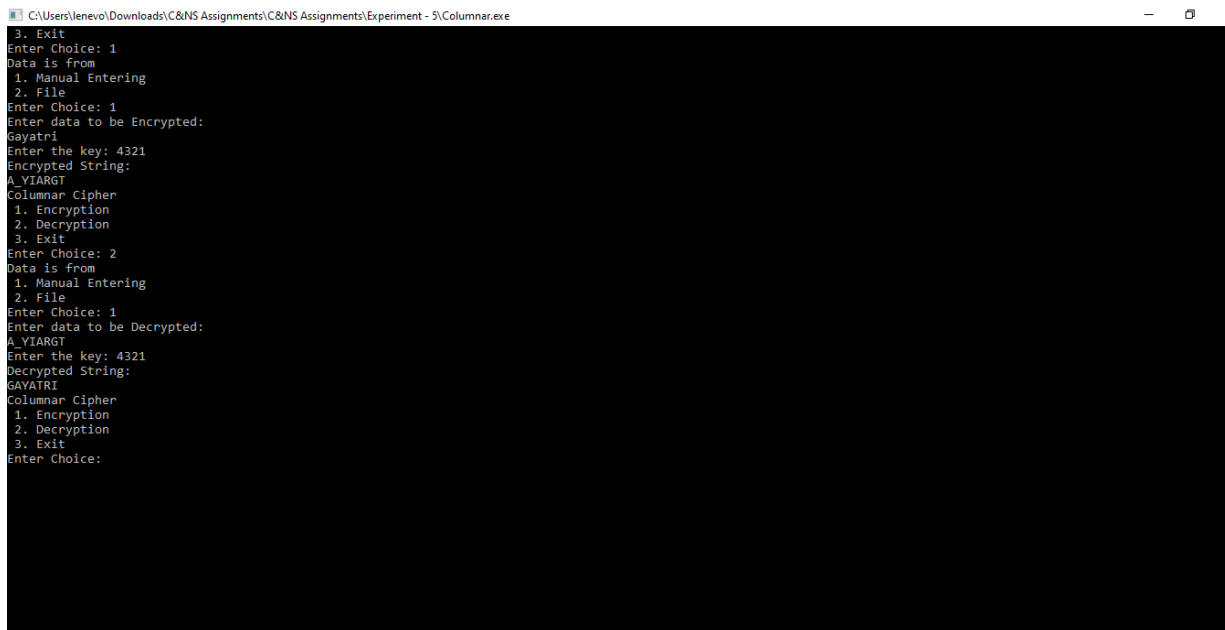


```
28     key = getText(key);
29     // cout<<key<<endl;
30     setPermutationOrder(key);
31     msg = getText(msg);
32     // cout<<msg<<endl;
33     int row,col,j;
34     string cipher="";
35     col = key.length();
36     row = msg.length()/col;
37
38     if(msg.length() % col)
39         row += 1;
40
41     char matrix[row][col];
42
43     for(int i=0,k=0;i<row;i++)
44     {
45         for(int j=0;j<col;j++)
46         {
47             if(msg[k] == '\0')
48             {
49                 matrix[i][j] = '_';
50                 j++;
51             }
52             if(isalpha(msg[k]) || msg[k] == ' ')
53             {
54                 matrix[i][j] = msg[k];
55                 j++;
56             }
57             k++;
58         }
59     }
60
61     for(map<int,int>::iterator ii = keyMap.begin(); ii != keyMap.end(); ++ii)
62     {
63         j = ii->second;
64         for(int i=0;i<row;i++)
```



```
62 {
63     j = ii->second;
64     for(int i=0;i<row;i++)
65     {
66         if(isalpha(matrix[i][j]) || matrix[i][j] == ' ' || matrix[i][j] == '_')
67             cipher += matrix[i][j];
68     }
69 }
70
71 return cipher;
72 }
73
74 string decryption(string cipher,string key)
75 {
76     key = getText(key);
77     // cout<<key<<endl;
78     setPermutationOrder(key);
79     cipher = getText(cipher);
80     // cout<<cipher<<endl;
81     int col = key.length();
82     int row = cipher.length()/col;
83     char cipherMatrix[row][col];
84
85     for(int j=0,k=0;j<col;j++)
86     {
87         for(int i=0;i<row;i++)
88             cipherMatrix[i][j] = cipher[k++];
89     }
90
91     int index=0;
92     for(map<int,int>::iterator ii = keyMap.begin(); ii != keyMap.end(); ++ii)
93     {
94         ii->second = index++;
95     }
96
97     string result="";
98     for(int i=0;i<row;i++)
99     {
100         for(int j=0;j<col;j++)
101             result += cipherMatrix[i][j];
102     }
103     return result;
104 }
```

Output Screenshots:



```
C:\Users\lenevo\Downloads\C&NS Assignments\C&NS Assignments\Experiment - 5\Columnar.exe
3. Exit
Enter Choice: 1
Data is from
1. Manual Entering
2. File
Enter Choice: 1
Enter data to be Encrypted:
Gayatri
Enter the key: 4321
Encrypted String:
A_YIARGI
Columnar Cipher
1. Encryption
2. Decryption
3. Exit
Enter Choice: 2
Data is from
1. Manual Entering
2. File
Enter Choice: 1
Enter data to be Decrypted:
A_YIARGI
Enter the key: 4321
Decrypted String:
GAYATRI
Columnar Cipher
1. Encryption
2. Decryption
3. Exit
Enter Choice:
```