**Assignment – 7**

**Q1. Write the Encryption and decryption procedure for**

**i) Column Transposition**

**ii) Route cipher**

**iii) Row column transposition**

**iv) Rail fence cipher**

**A:**



*Pseudocode and Explanation –*

1. **Input:** It takes a plain text and a key (used for column transposition).
2. **Grid Setup:** The plain text is arranged in a grid with rows determined by the key length. If the text doesn't fill the grid, extra characters (A, B, C, ...) are added.
3. **Encryption:**
   * Characters from the grid columns are rearranged based on the alphabetical order of the key.
   * A map stores the characters column-wise, indexed by the key.
   * The encrypted text is formed by reading characters column-wise based on the sorted key.
4. **Decryption:**
   * The process is reversed by mapping the characters back to their original columns using the original (unsorted) key.
   * The decrypted text is formed by reading characters row-wise from the grid.

The code outputs the encrypted and decrypted text.

*Code-*

// Column Transposition

#include<bits/stdc++.h>

using *namespace* std;

*int* main()

{

    string plain\_text;

    string key, key2;

    cout<<"Enter the plain text: "<<endl;

    cin>>plain\_text;

    cout<<"Enter the key: "<<endl;

    cin>>key;

    key2 = key;

*int* col = key.length();

*int* row;

    if(plain\_text.length()%col == 0){

        row = plain\_text.length()/col;

    }else{

        row = plain\_text.length()/col + 1;

    }

    // Encryption

*int* p = 0;

*int* h = 0;

*int* q=0;

    cout<<"Row: "<<row<<" Col: "<<col<<endl;

    vector<vector<*char*>> grid(row, vector<*char*>(col));

    for(*int* i=0;i<(row\*col);i++){

        if(q == col){

            p++;

            q=0;

        }

        if(i >= plain\_text.length()){

            grid[p][q] = *char*(65 + h);

            h++;

        }else{

            grid[p][q] = plain\_text[i];

        }

        q++;

    }

    for(*int* i=0;i<row;i++){

        for(*int* j=0;j<col;j++){

            cout<<grid[i][j]<<" ";

        }

        cout<<endl;

    }

    unordered\_map<*char*, vector<*char*>> alpha\_map;

    for(*int* i=0;i<key.length();i++){

        for(*int* j=0;j<row;j++){

            alpha\_map[key[i]].push\_back(grid[j][i]);

        }

    }

    sort(key.begin(),key.end());

    string encrypted\_text;

    for(*int* i=0;i<key.length();i++){

*char* ch = key[i];

        for(*int* j=0;j<row;j++){

            encrypted\_text+=alpha\_map[ch][j];

        }

    }

    cout<<"Encrypted Text: "<<encrypted\_text;

    // Decryption

    vector<vector<*char*>> grid\_2(row, vector<*char*>(col));

    for(*int* i=0;i<col;i++){

        for(*int* j=0;j<row;j++){

            grid\_2[j][i] = alpha\_map[key2[i]][j];

        }

    }

    cout<<endl;

    for(*int* i=0;i<row;i++){

        for(*int* j=0;j<col;j++){

            cout<<grid\_2[i][j]<<" ";

        }

        cout<<endl;

    }

    string decrypted\_text;

    for(*int* i=0;i<row;i++){

        for(*int* j=0;j<col;j++){

            decrypted\_text+=grid\_2[i][j];

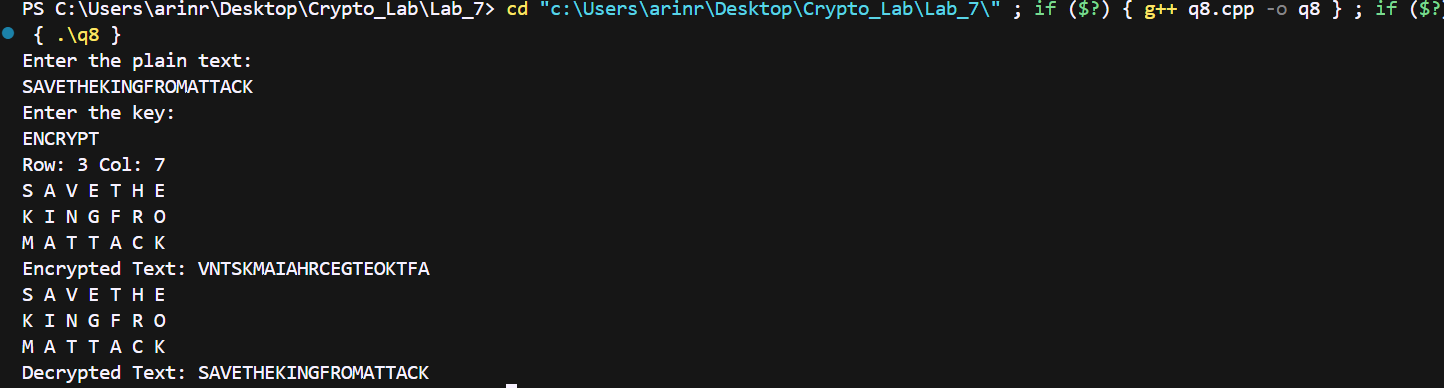
        }

    }

    cout<<"Decrypted Text: "<<decrypted\_text;

}

*Output –*

**

ii)

*Pseudocode and Explanation –*

1. **Input:**
   * The user inputs a plain text message and the grid size (row and col).
2. **Grid Setup:**
   * The plain text is arranged into a grid of dimensions row x col.
   * If the plain text doesn't fill the grid, additional characters (A, B, C, ...) are added to fill the grid.
3. **Encryption (Spiral/Route traversal):**
   * The grid is traversed in a spiral order:
     + Start from the top-right corner.
     + Move downward along the rightmost column.
     + Then, move left along the bottom row.
     + Move upward along the leftmost column.
     + Continue this pattern, spiraling inward until the entire grid is traversed.
   * The characters are collected in the order they are traversed to form the encrypted text.
4. **Output:**
   * The final encrypted text is printed after completing the spiral traversal of the grid.