**Ceasar Cipher**

**Code**

#include <iostream>

using namespace std;

string encrypt(string text, int shift)

{

string result;

result = "";

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

if (isupper(text[i]))

result += char(int(text[i] + shift - 65) % 26 + 97);

else

result += char(int(text[i] + shift - 97) % 26 + 65);

}

return result;

}

string decrypt(string a, int shift)

{

string result;

result = "";

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

if (isupper(a[i]))

result += char(int(a[i] - shift - 65) % 26 + 97);

else

result += char(int(a[i] - shift - 97) % 26 + 65);

}

return result;

}

int main()

{

string text,a;

int shift;

cout << "Enter a Text here: ";

cin >> text;

cout << "Enter a Shift count: ";

cin >> shift;

cout << "Text : " << text;

a = encrypt(text,shift);

cout << "\nCipher: " << a;

cout << "\nMain text: "<< decrypt(a,shift);

return 0;

}

**Output**

Enter a Text here: CSBS

Enter a Shift count: 3

Text : CSBS

Cipher: fvev

Main text: CSBS

**Rail Fence**

**Code**

#include <bits/stdc++.h>

using namespace std;

string encryptRailFence(string text, int key) {

char rail[key][text.length()];

for (int i = 0; i < key; i++)

for (int j = 0; j < text.length(); j++)

rail[i][j] = '\n';

bool dir\_down = false;

int row = 0, col = 0;

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

if (row == 0 || row == key - 1)

dir\_down = !dir\_down;

rail[row][col++] = text[i];

dir\_down ? row++ : row--;

}

string result;

for (int i = 0; i < key; i++)

for (int j = 0; j < text.length(); j++)

if (rail[i][j] != '\n')

result.push\_back(rail[i][j]);

return result;

}

string decryptRailFence(string cipher, int key) {

char rail[key][cipher.length()];

for (int i = 0; i < key; i++)

for (int j = 0; j < cipher.length(); j++)

rail[i][j] = '\n';

bool dir\_down;

int row = 0, col = 0;

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

if (row == 0)

dir\_down = true;

if (row == key - 1)

dir\_down = false;

rail[row][col++] = '\*';

dir\_down ? row++ : row--;

}

int index = 0;

for (int i = 0; i < key; i++)

for (int j = 0; j < cipher.length(); j++)

if (rail[i][j] == '\*' && index < cipher.length())

rail[i][j] = cipher[index++];

string result;

row = 0, col = 0;

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

if (row == 0)

dir\_down = true;

if (row == key - 1)

dir\_down = false;

if (rail[row][col] != '\*')

result.push\_back(rail[row][col++]);

dir\_down ? row++ : row--;

}

return result;

}

int main() {

string a,c;

int b;

cout << "Enter Plain Text: ";

cin >> a;

cout << "Enter shift value: ";

cin >> b;

c=encryptRailFence(a, b);

cout<<"Encryption"<<endl;

cout << c << endl;

cout<<"decryption"<<endl;

cout << decryptRailFence(c, b) << endl;

return 0;

}

**Output**

Enter Plain Text: csbs

Enter key: 3

Encryption

cssb

decryption

csbs