**Program :**

#include <bits/stdc++.h>

using namespace std;

string encryptNormal(string text, int s)

{

    string result = "";

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

        if (text[i] == ' ')

            result += ' ';

        else if (isupper(text[i]))

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

        else

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

    }

    return result;

}

string decryptNormal(string text, int s)

{

    string result = "";

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

        if (text[i] == ' ')

            result += ' ';

        else if (isupper(text[i]))

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

        else

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

    }

    return result;

}

string encryptSpecial(string text, int s)

{

    string result = "";

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

        if (text[i] == ' ')

            result += ' ';

        else

            result += char(int(text[i] + s - 33) % 93 + 33);

    }

    return result;

}

string decryptSpecial(string text, int s)

{

    string result = "";

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

        if (text[i] == ' ')

            result += ' ';

        else

            result += char(int(text[i] - s - 33) % 93 + 33);

    }

    return result;

}

int main()

{

    string text;

    int key, choice;

    cout<<"Enter plain text :";

    getline(cin, text);

    cout<<"Enter the key :";

    cin>>key;

    cout<<"\nFor simple conversion Enter 0\n"<<"For special character conversion Enter 1"<<endl;

    cin>>choice;

    if(choice){

        string encrypted = encryptSpecial(text, key);

        cout << "\nCipher text: " << encrypted<<endl;

        cout << "Derypted text: " << decryptSpecial(encrypted, key)<<endl;

    }

    else{

        string encrypted = encryptNormal(text, key);

        cout << "\nCipher text: " << encrypted<<endl;

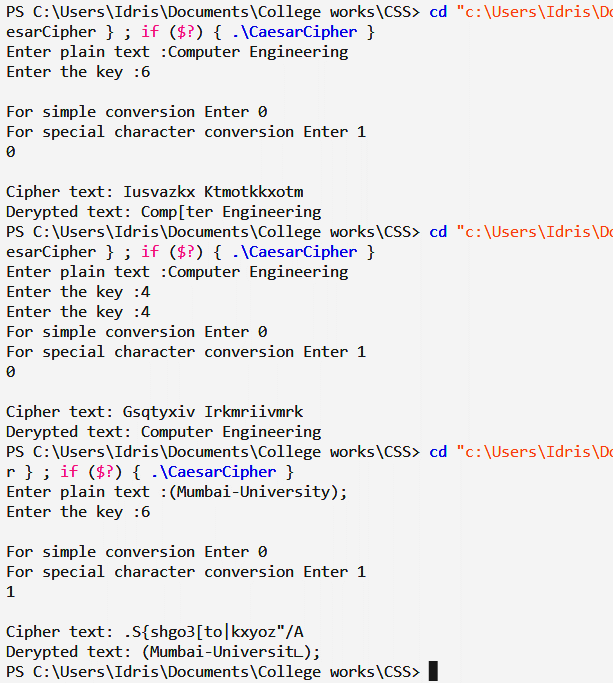
        cout << "Derypted text: " << decryptNormal(encrypted, key)<<endl;

    }

    return 0;

}

**Output :**

****