DM Project Report
Title: Caeser Cipher Encryption and Decryption

- GROUP MEMBERS:
  - MUHAMMAD KASHIF 11762
  - FAHAD IMRAN 13018
  - WAHEED MANZOR AFRIDI 11723
- Class ID
  - 108940

- DISCRIPTION:

Encryption

Encryption is the process by which a readable message is converted to an unreadable form to prevent unauthorized parties from reading it.. The original message is called the plaintext message.

We are using Caeser Cipher Method to encrypt the messages. In this Method a code number is given through which the letters are changed for example a given number is 4 and the plain text is “Maths” than using Caeser Cipher Method, Every character will be changed with the next 4^{th} character.

Equation

En (x) = (x + n) mod 26
Where X is Serial number of character.
And n is Code number (number of jump)

Serial Number	Character	Serial Number	Character
0	A	13	N
1	B	14	O
2	C	15	P
3	D	16	Q
4	E	17	R
5	F	18	S
6	G	19	T
7	H	20	U
8	I	21	V
9	J	22	W
10	K	23	X
11	L	24	Y
12	M	25	Z

Code

```
private string Encryption(string txt,int key)
```

```
{
    string plainCapital = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    string plainSmall = "abcdefghijklmnopqrstuvwxyz ";
    this.shift = key;
    string cipher = "";
    char space = ' ';
    for (int i= 0;i<txt.Length;i++)
    {
        if(char.IsUpper(txt[i]))
        {
            for (int j = 0; j < plainCapital.Length; j++)
            {
                if (txt[i] == plainCapital[j])
                {
                    cipher = cipher + plainCapital[(j + shift) % 26];
                }
            }
        }
        else if (txt[i] == space)
        {
            cipher = cipher + " ";
        }
        else
        {
            for (int j = 0; j < plainSmall.Length; j++)
            {
                if (txt[i] == plainSmall[j])
                {
                    cipher = cipher + plainSmall[(j + shift) % 26];
                }
            }
        }
    }
    return cipher;
       }
```

Decryption

Decryption is the process of converting an encrypted message back to its original (readable) format. The original message is called the plaintext message.

In Decryption we are also using Caeser Cipher Method to decrypt the messages. In this Method a code number is given through which the letters are changed for example a given number is 4 and the plain text is “Maths” than

using Caeser Cipher Method, Every character will be changed with the previous 4th character.

Equation

En (x) = (x - n) mod 26

Where X is Serial number of character.

And n is Code number (number of jump)

Serial Number	Character	Serial Number	Character
0	A	13	N
1	B	14	O
2	C	15	P
3	D	16	Q
4	E	17	R
5	F	18	S
6	G	19	T
7	H	20	U
8	I	21	V
9	J	22	W
10	K	23	X
11	L	24	Y
12	M	25	Z

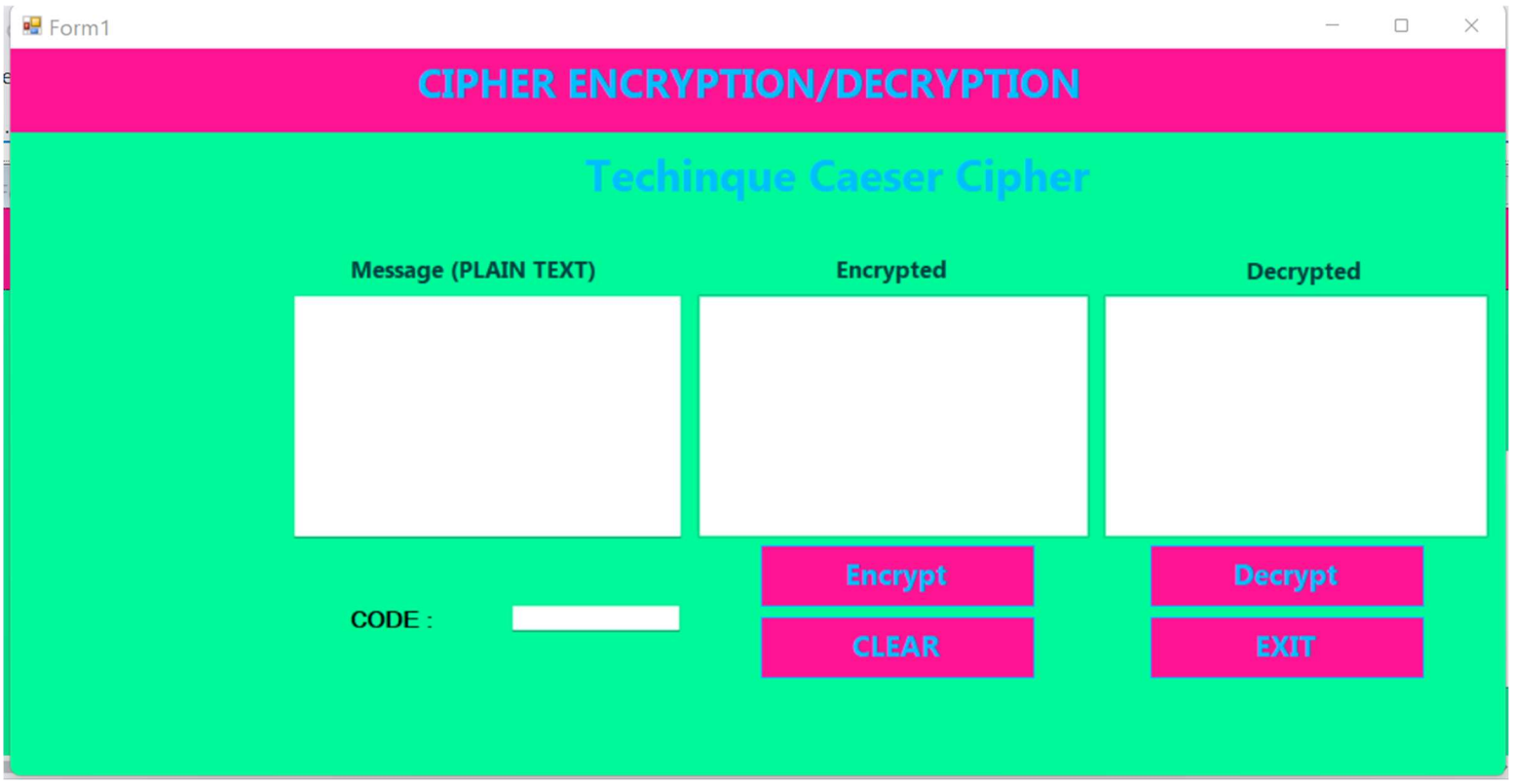
Code

```
private string Decryption(string txt)
        {
            string plainCapital = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
            string plainSmall = "abcdefghijklmnopqrstuvwxyz";
            string CipherDec = "";
            char space = ' ';
            for (int i = 0; i < txt.Length; i++)
            {
                if (char.IsUpper(txt[i]))
                {
                    for (int j = 0; j < plainCapital.Length; j++)
                    {
                        if (txt[i] == plainCapital[j])
                        {
                            if (j >= shift)
                            {
                                CipherDec = CipherDec + plainCapital[(j - shift) % 26];
                            }
                            else
                            {
```

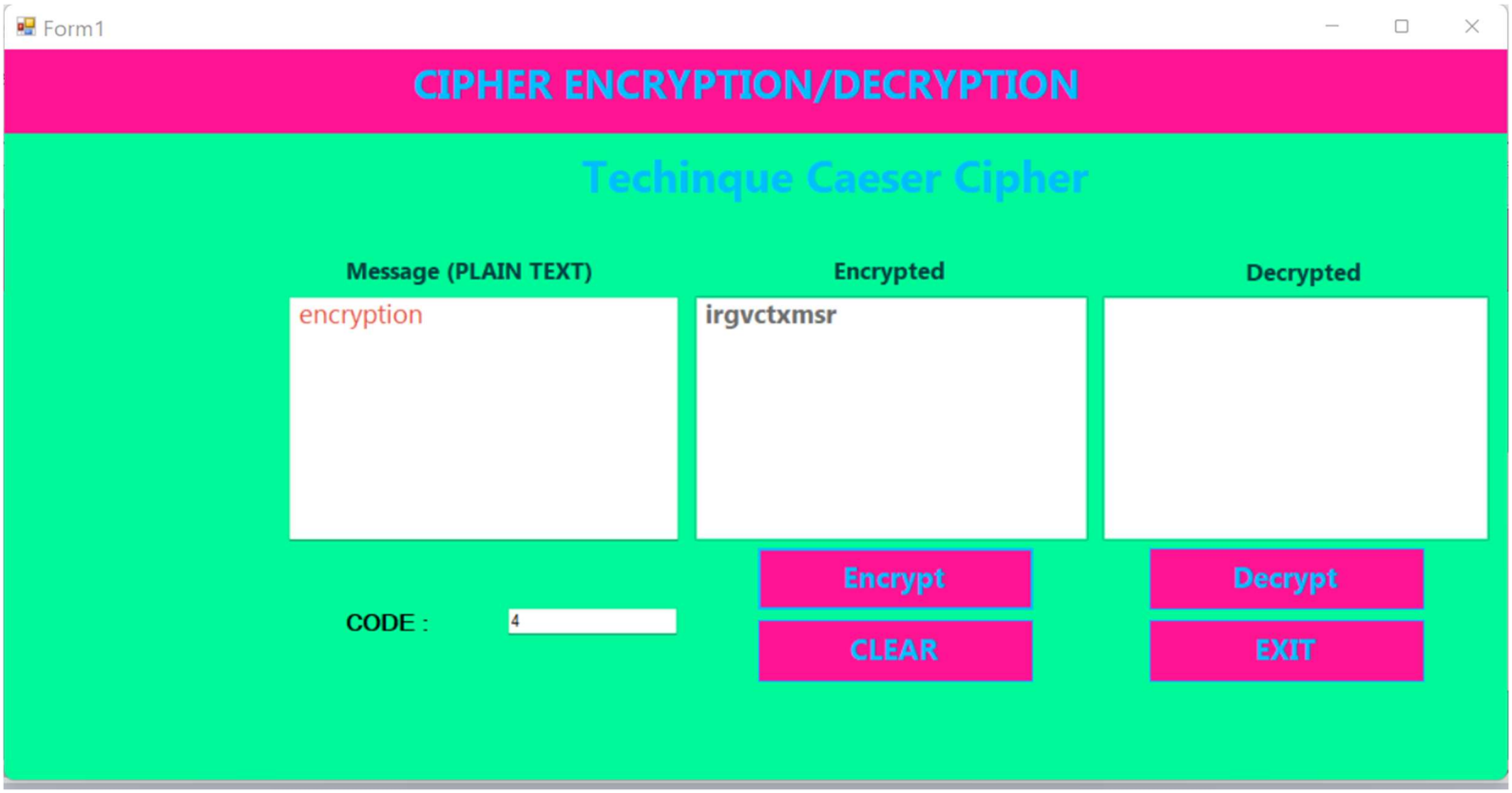
```
                                int NonNegativeIndex = ((shift-j) % 26);
                                CipherDec = CipherDec + plainCapital[26-
NonNegativeIndex];
                            }

                        }
                    }
                }
                else if (txt[i] == space)
                {
                    CipherDec = CipherDec + " ";
                }
                else
                {
                    for (int j = 0; j < plainSmall.Length; j++)
                    {
                        if (txt[i] == plainSmall[j])
                        {
                            if (j >= shift)
                            {
                                CipherDec = CipherDec + plainSmall[(j - shift) % 26];
                            }
                            else
                            {
                                int NonNegativeIndex = ((shift - j) % 26);
                                CipherDec = CipherDec + plainSmall[26 -
NonNegativeIndex];
                            }

                        }
                    }
                }
            }
            return CipherDec;
           }
```

ScreenShots

Simple



Encryption



Decryption