

# NIS LAB-1

Roll NO. : CE146

Name : Shingai Shubham P.

ID NO. : 19CEU05159

\* AIM :- Write a program to implement

- 1) additive / shift / caesar cipher
- 2) monoalphabetic substitution cipher.

1) additive / shift / caesar cipher

→ source code :

```
#include <iostream>
#include <string>
using namespace std;

int mod410 (int a, int n) {
    int mod410N = a % n;
    if (mod410N < 0)
        mod410N += n;
    return mod410N
}
```

```
string Encrypt (int key, string plainText)
{
    string cipherText;
    int lengthOfText = plainText.length(),
    textMap;
```



```
for (int i=0 ; i < lengthofText ; i++)
```

```
{
```

```
    if (isupper (plainText[i]))
```

```
        textMap = plainText[i] - 'A' ;
```

```
    else
```

```
        textMap = plainText[i] - 'a' ;
```

```
    cipherText += modulo (textMap+key, 26)
```

```
    + 'A' ;
```

```
    return cipherText ;
```

```
}
```

```
string Decrypt (int key, string cipherText)
```

```
{
```

```
    string plainText ;
```

```
    int lengthofcipher = cipherText.length(),
```

```
        textMap ;
```

```
    for (int i=0 ; i < lengthofcipher ; i++)
```

```
{
```

```
        textMap = cipherText[i] - 'A' ;
```

```
        plainText += modulo (textMap - key,
```

```
        26) + 'a' ;
```

```
}
```

```
    return plainText ;
```

```
}
```



```
int main ()
```

```
{
```

```
    string plainText;
```

```
    cout << " Enter Plain Text:" << endl;
```

```
    cin >> plainText;
```

```
    int key;
```

```
    cout << " Enter key ;" ;
```

```
    cin >> key ;
```

```
    string encryptText = Encrypt (key, plainText);
```

```
    cout << "Encrypt Text:" << encryptText << endl;
```

```
    cout << "Decrypt Text:" << Decrypt (key,
    encryptText) << endl << endl;
```

```
    // cryptanalysis using brute force.
```

```
    cout << " Enter cipher Text ;" ;
```

```
    string cipherText;
```

```
    cin >> cipherText;
```

```
    for (int i=1; i<26; i++)
```

```
        cout << " key:" << i << " " << .
```

```
        Decrypt (i, cipherText) << endl;
```

```
    return 0;
```

```
}
```



Input: Enter Plain Text : Caesar cipher.  
Enter Key : 5

Output: Encrypt Text : HFJXFWHNUMJW  
Decrypt Text : caesar cipher

Input: Enter Cipher Text : HFJXFWHNUMJW

Output: Key: 1 geiWGVgmtIIV

Key: 2 AdhvdvPISKhY

Key: 3 Ec-gNcteK&ygt

Key: 4 idbFtbSdjqifS

Key: 5 caesar cipher

Key: 6 bzdrZ9bhogd9

Key: 7 elyC9yPagnfCP

Key: 25 igKy9ociOVnKx



## 2) monoalphabetic substitution cipher.

→ source code:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
vector<int> generateKey() {
```

```
    vector<int> key;
```

```
    int n = 26, i;
```

```
    for (i = 0; i < n; i++)
```

```
        key.push_back(i);
```

```
    for (i = n-1; i > 0; i--)
```

```
    {
```

```
        srand((unsigned int)time(NULL));
```

```
        int j = rand() % i;
```

```
        int temp = key[i];
```

```
        key[i] = key[j];
```

```
        key[j] = temp;
```

```
    }
```

```
    cout << "in map : ";
```

```
    for (i = 0; i < n; i++)
```

```
        cout << char(i + 'A') << " ";
```

```
    for (i = 0; i < n; i++)
```

```
        cout << char(key[i] + 'A') << " ";
```

```
    cout << "in m";
```

```
    return key;
```

```
}
```



```
string monoalphabeticCipher (string plainText)
{
```

```
    vector<int> Key = generateKey();
```

```
    string encryptText;
```

```
    for (int i=0; i< plainText.length(); i++)
```

```
    {
```

```
        if (isupper(plainText[i])) {
```

```
            int q = plainText[i] - 'A';
```

```
            encryptText += Key[q] + 'A';
```

```
        }
```

```
        else {
```

```
            int q = plainText[i] - 'a';
```

```
            encryptText += Key[q] + 'a';
```

```
        }
```

```
    }
    return encryptText;
```

```
}
```

```
int main ()
```

```
{
```

```
    string plainText, encryptText;
```

```
    cout << "Enter plain text : ";
```

```
    cin >> plainText;
```

```
    encryptText = monoalphabeticCipher(plainText);
```

```
    cout << "Encrypt Text : " <<
```

```
        encryptText;
```

```
    return 0;
```

```
}
```



## \* Test Case - 1:

Input: Enter Plain Text :  
monocryptetic Cipher

Output:

map: A B C D E F G H I J K L M N O P Q R S T U  
key: B D P E H Q I N J M L W S O K F R T Y X K

V W X Y Z

V Z V C A

Encrypt Text : SKOKBWFNB D H X I P P J F N H T

## \* Test Case - 2:

INPUT: Enter plain Text :  
Shybam Shingaiq

map: A B C D E F G H I J K L M N  
key: G A E J H Z M P Y C D K T F

map: O P Q R S T U V W X Y Z  
key: Q W B X L U N I V O R S

Encrypt Text : LPNAPQTLPYFMGKG