**Shift Cipher Brute force attack**

**Man in a middle attack for ceaser cipher**

#include<bits/stdc++.h>

using namespace std;

char str[100],str1[100];

void encrypt(int key)

{

for(int i=0;i<strlen(str);i++)

{

if((int)str[i]+key>122)

{

str[i] = (char)((int)str[i]+key-26);

continue;

}

else

{str[i]=(char)((int)str[i]+key);}

}

cout<<"Encrypted String is : "<<endl;puts(str);

}

void decrypt(int key)

{

for(int i=0;i<strlen(str);i++)

{

if((int)str[i]-key<97) str1[i] = (char)((int)str[i]-key+26);

else str1[i]=(char)((int)str[i]-key);

}

cout<<"Key is : "<<key <<"\nDecrypted String is : ";puts(str1);

}

int main()

{

int key;

cout<<"Enter String : "<<endl;

gets(str);

cout<<"Enter Key : "<<endl;

cin >> key;

encrypt(key);

decrypt(key);

cout<<"Attack starts : "<<endl;

for(int i=0;i<26;i++)

{

decrypt(i);

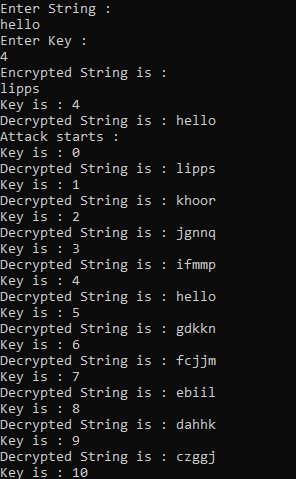
}

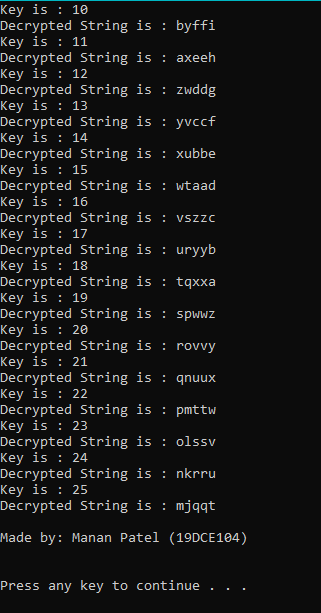
cout<<endl<<"Made by: ========"<<endl;

return 0;

}

**Output :**

****

****

**Brute force attack on Affine Cipher**

[**Attacks-on-Affine-Cipher**](https://github.com/sukhdev01/Attacks-on-Affine-Cipher)

https://github.com/sukhdev01/Attacks-on-Affine-Cipher/blob/master/Brute\_force\_attacks\_on\_Affine\_cipher.ipynb

**Affine Cipher Enc/Decrpy**

<https://www.geeksforgeeks.org/implementation-affine-cipher/>

**RSA encp and Decrpy**

import math

def checkPrime(n):

k = math.sqrt(n)

for i in range(3,int(k+1)):

if n%i == 0:

return False

return True

def gcd(a,b):

if b == 0:

return a

return gcd(b,a%b)

def get\_public\_key(phi):

for i in range(2,10000):

e = gcd(i,phi)

if e == 1:

return i

return False

def get\_private\_key(phi, e, t1, t2):

global d

if e == 0:

d = t1

return t1

get\_private\_key(e, phi%e, t2, t1-(phi//e)\*t2)

def encrpyt(M, e, n):

return (M\*\*e)%n

def decrypt(C, d, n):

return (C\*\*d)%n

p = int(input("Enter First Prime number: "))

while not checkPrime(p):

print(f'{p} is not a prime number. Try Again.')

p = int(input("\nEnter First Prime number: "))

q = int(input("\nEnter Second Prime number: "))

while not checkPrime(q):

print(f'{q} is not a prime number. Try Again.')

q = int(input("\nEnter Second Prime number: "))

n = p\*q

phi = (p-1)\*(q-1)

e = get\_public\_key(phi)

if e:

get\_private\_key(phi, e, 0, 1)

if d<0:

d = d+phi

print(f'\nPublic ckey e = {e}, Private key d = {d}')

M = int(input("\nEnter Your Message (Number): "))

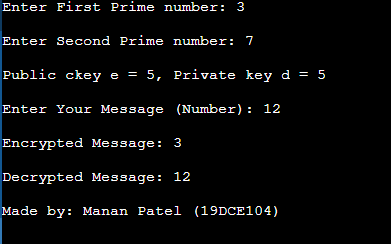
C = encrpyt(M, e, n)

print(f"\nEncrypted Message: {C}")

original\_message = decrypt(C, d, n)

print(f"\nDecrypted Message: {original\_message}")

print("\nMade by: ========= (19DC====)")



**RailFence Cipher encp and Decrpy**

#include <bits/stdc++.h>

using namespace std;

string s ={}, s1={};

int key;

void encryption(){

cout<<endl<<"--------encryption---------"<<endl;

int len;

cout<<"Enter Text :- ";

getline (cin,s);

len=s.length();

cout<<"Enter Key :- ";

cin>>key;

char ans[key][len]={};

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

{for(int j = 0; j < len; ++j)

{ans[i][j] = '\n';}}

int j=-1,row=0;

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

{

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

j=j\*(-1);

}

ans[row][i] = s[i];

row=row+j;

}

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

{

for (int j = 0; j < len; j++) {

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

// cout<<ans[i][j];

s1+=ans[i][j];

}

else{ans[i][j]='-';}

}

}

cout<<endl;

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

{

for (int j = 0; j < len; j++) {

cout<<ans[i][j];

}

cout<<endl;

}

cout<<endl;

cout<<"encryption message : - ";

cout<<s1;

}

void decryption(){

cout<<endl<<"--------decryption---------"<<endl;

int len,m=0;

len=s1.length();

char ans[key][len]={};

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

{for(int j = 0; j < len; ++j)

{ans[i][j] = '\n';}}

int j=-1,row=0;

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

{

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

j=j\*(-1);

}

ans[row][i] = '\*';

row=row+j;

}

cout<<endl;

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

{

for (int j = 0; j < len; j++) {

if(ans[i][j]=='\*'){ans[i][j]=s1[m++];}

}

}

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

{

for (int j = 0; j < len; j++) {

if(ans[i][j]=='\n'){cout<<"-";}

else {cout<<ans[i][j];}

}

cout<<endl;

}

cout<<endl;

cout<<"decryption message : - ";

row =0, j=-1;

for(int i = 0; i < len; i++){

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

{j= j \* (-1);}

cout<<ans[row][i];

row = row + j;

}

}

int main()

{

encryption();

decryption();

cout<<endl<<"Made by: ======== (19DC===) "<<endl;

return 0;

}

