TW1-Substitution Cipher

```
#include<stdio.h>
int main()
{
 char message[100], ch, str;
 int i, key, x;
 printf("Enter a message : ");
 gets(message);
 printf("Enter key: ");
 scanf("%d", &key);
 printf("\nPlease choose following options:\n");
 printf("1 = Encrypt the string.\n");
 printf("2 = Decrypt the string.\n");
 scanf("%d", &x);
switch(x)
 {
 case 1:
 for(i = 0; message[i] != '\0'; ++i)
 {
 ch = message[i];
 if(ch >= 'a' && ch <= 'z')
 {
 ch = ch + key;
 if(ch > 'z')
 {
 ch = ch - 'z' + 'a' - 1;
 }
 message[i] = ch;
 }
 else if(ch >= 'A' && ch <= 'Z')
 {
 ch = ch + key;
 if(ch > 'Z')
 {
 ch = ch - 'Z' + 'A' - 1;
 }
 message[i] = ch;
 }
 }
 printf("Encrypted message: %s", message);
 break;
 case 2:
 for(i = 0; message[i] != '\0'; ++i)
 {
 ch = message[i];
 if(ch >= 'a' && ch <= 'z')
 {
 ch = ch - key;
 if(ch < 'a')
 {
 ch = ch + 'z' - 'a' + 1;
 }
 message[i] = ch;
 }
 else if(ch >= 'A' && ch <= 'Z')
 {
 ch = ch - key;
 if(ch < 'A')
 {
 ch = ch + 'Z' - 'A' + 1;
 }
 message[i] = ch;
 }
 }
 printf("Decrypted message: %s", message);
 break;
 default:
 printf("\nError\n");
 }
return 0;
}
```

Sample Output

```
Enter a message : I am studying Data Encryption
Enter key: 4
Please choose following options:
1 = Encrypt the string.
2 = Decrypt the string.
1
Encrypted message: M eq wxyhcmrk Hexe Irgvctxmsr
Enter a message : M eq wxyhcmrk Hexe Irgvctxmsr
Enter key: 4
Please choose following options:
1 = Encrypt the string.
2 = Decrypt the string.
2
Decrypted message: I am studying Data Encryption
```

TW2-RSA

```
#include <stdio.h>
#include<stdlib.h>
int gcd(int a,int b)
{
int c;
while(a!=b)
{
if(a<b)
{
c=a;a=b;b=c;
}
a-=b;
}
return a;
}
int mod(int m,int e,int n)
{
int a=1;
while(e)
{
a=(a*m)%n;
e--;
}
return a;
}
int main()
{
int p,q,n,e,m,c,d,x,z;
int en[100],de[100],j=0;
printf("\nEnter the value of P & Q\n");
scanf("%d%d",&p,&q);
n=p*q;
z=(p-1)*(q-1);
for(e=1;e<n;e++)
{
if(gcd(e,z)==1)
{
en[j]=e;
printf(" %d",en[j++]);
}
}
printf("\nChoose e\n");
scanf("%d",&e);
if(gcd(e,z)!=1)
{
printf("\nThe value not from list\n");
exit(0);
}
printf("Enter the message(integer value) to be encrypted:\n");
scanf("%d",&m);
printf("Before encryption:%d\n",m);
c=mod(m,e,n);
printf("After encryption:%d\n",c);
printf("The possible Decryption keys Are:");
for(d=0;d<n;d++)
{
if((d*e)%z==1)
{
de[j]=d;
printf(" %d",de[j++]);
}
}
printf("\nChoose D\n");
scanf("%d",&d);
x=mod(c,d,n);
printf("After decryption=%d\n",x);
return 0;
}
```

Sample Output

```
Enter the value of P & Q
7 11
1 7 11 13 17 19 23 29 31 37 41 43 47 49 53 59 61 67 71 73
Choose e
7
Enter the message (integer value) to be encrypted:
9
Before encryption:9
After encryption:37
The possible Decryption Keys Are: 43
Choose D
43
After decryption=9
```

TW3-Password

```
int main()
{
 int i,n,a=0,d=0,s=0;
 char p[10];
 printf("Enter the Password: ");
 gets(p);
 n=strlen(p);
 if(n>=6)
 {
 for(i=0;i<n;i++)
 {
 if(isalpha(p[i]))
 {
 a+=1;
 }
 else if(isdigit(p[i]))
 {
 d+=1;
 }
 else
 {
 s+=1;
 }
 }
 if(a>=1 && d>=1 && s>=1)
 {
 printf("Strong Password");
 }
 else if((a>=1 && d>=1) || (a>=1 && s>1) || (d>=1 && s>=1))
 {
 printf("Moderate Password");
 }
 else
 {
 printf("Weak Password");
 }
 }
 else
 {
printf("Invalid Password");
 }
}
```

Sample Output

```
Enter password : aw1
Invalid Password
Enter password : adckex
Weak Password
Enter password : abc123
Moderate Password
Enter password : abc2#@
Strong Password
```

TW-4 Rail fence

```c
#include<stdio.h>
#include<string.h>
void encryptMsg(char msg[], int key)
{
 int msgLen = strlen(msg), i, j, k = -1, row = 0, col = 0;
 char railMatrix[key][msgLen];
 for(i = 0; i < key; ++i)
 for(j = 0; j < msgLen; ++j)
 railMatrix[i][j] = '\n';
 for(i = 0; i < msgLen; ++i)
 {
 railMatrix[row][col++] = msg[i];
 if(row == 0 || row == key-1)
 k= k * (-1);
 row = row + k;
 }
 printf("\nOutput :");
 char nlet[100];
 for(i = 0; i < key; ++i)
 for(j = 0; j < msgLen; ++j)
 if(railMatrix[i][j] != '\n')
 {
 printf("%c", railMatrix[i][j]);
 strncat(nlet, &railMatrix[i][j], 1);
 }
}
int main()
{
 char msg[100];
 int key;
 printf("Encryption:");
 printf("\nInput:");
 scanf("%[^\n]s",msg);
 printf("Key = ");
 scanf("%d",&key);
 encryptMsg(msg, key);
 return 0;
}
```

Sample Output

```
Encryption
Input : attack at once
Key = 2
Output : atc toctaka ne
```

TW-5 Diffie Hellman

```
#include <stdio.h>
#include <stdlib.h>

long int power(int a ,int b ,int mod)
{
    long long int t;
    if(b==1)
        return a;
    t=power(a, b / 2, mod);
    if(b%2 == 0)
        return (t * t) % mod;
    else
        return (((t * t) % mod) * a) % mod;
}
long long int key(int a,int x,int q)
{
    return power(a,x,q);
}
int main()
{
    int alpha,q,a,x,b,y;
    printf("enter the prime number and prime root\n");
    scanf("%d%d",&q,&alpha);
    printf("enter the private key of a\n");
    scanf("%d",&x);
    a=power(alpha,x,q);
    printf("enter the private key of b\n");
    scanf("%d",&y);
    b=power(alpha,y,q);
    printf("key calculated by a=%lld\n",key(b,x,q));
     printf("key calculated by b=%lld\n",key(a,y,q));
     return 0;
}
```

Sample Output

```
Enter the prime number and prime root : 23 9
Enter the private key of A : 4
Enter the private key of B : 3
A computes key K : 9
B computes key K : 9
```

TW-6 Hill cipher

```
void getKeyMatrix(char key[6], int keyMatrix[][3])
{
 int k = 0;
 for (int i = 0; i < 3; i++)
 {
 for (int j = 0; j < 3; j++)
 {
 keyMatrix[i][j] = (key[k]) % 65;
 k++;
 }
 }
}
void encrypt(int cipherMatrix[][1],
 int keyMatrix[][3],
 int messageVector[][1])
{
 int x, i, j;
 for (i = 0; i < 3; i++)
 {
 for (j = 0; j < 1; j++)
 {
 cipherMatrix[i][j] = 0;
 for (x = 0; x < 3; x++)
 {
 cipherMatrix[i][j] +=keyMatrix[i][x] * messageVector[x][j];
 }
 cipherMatrix[i][j] = cipherMatrix[i][j] % 26;
 }
 }
}
void HillCipher(char message[3], char key[9])
{

 int keyMatrix[3][3];
 getKeyMatrix(key, keyMatrix);
 int messageVector[3][1];

 for (int i = 0; i < 3; i++)
 messageVector[i][0] = (message[i]) % 65;
 int cipherMatrix[3][1];
encrypt(cipherMatrix, keyMatrix, messageVector);
 char CipherText[3];
 for (int i = 0; i < 3; i++)
 CipherText[i] = cipherMatrix[i][0] + 65;
printf("%s", CipherText);
}
int main()
{
 // Get the message to be encrypted
 char message[3],key[9];
 printf("Enter the plaintext :");
 scanf("%s",message);
 // Get the key
 printf("Enter the key :");
 scanf("%s",key);
 HillCipher(message, key);
 return 0;
}
```

Sample Output

```
Enter the plaintext :ACT
Enter the key :GYBNQKURP
POH
```

TW-7 DES

```
#include<stdio.h>

int main() {
    int i, cnt=0, p8[8] = {6,7,8,9,1,2,3,4};
    int p10[10] = {6,7,8,9,10,1,2,3,4,5};
    char input[11], k1[9], temp[11];

    printf("Enter 10 bits input:");
    scanf("%s", input);
    input[10] = '\0';

    // Applying p10...
    for(i = 0; i < 10; i++) {
        cnt = p10[i];
        temp[i] = input[cnt - 1];
    }
    temp[i] = '\0';

    printf("\nYour p10 key is :");
    for(i = 0; i < 10; i++) {
        printf("%d,", p10[i]);
    }

    printf("\nBits after p10 :");
    puts(temp);

    // Performing LS-1 on temp
    char temp_bit = temp[0];
    for(i = 0; i < 4; i++) {
        temp[i] = temp[i + 1];
    }
    temp[4] = temp_bit;

    printf("Output after LS-1 :");
    puts(temp);

    printf("\nYour p8 key is :");
    for(i = 0; i < 8; i++) {
        printf("%d,", p8[i]);
    }

    // Applying p8...
    for(i = 0; i < 8; i++) {
        cnt = p8[i];
        k1[i] = temp[cnt - 1];
    }
    k1[i] = '\0';

    printf("\nYour key k1 is :");
    puts(k1);

    return 0;
}
```

Sample Output

```
Enter 10 bits input:1100011100
Your p10 key is :6,7,8,9,10,1,2,3,4,5,
Bits after p10 :1110011000
Output after LS-1 :1100110001
Your p8 key is :6,7,8,9,1,2,3,4,
Your key k1 is :10001100
```

TW-8 Vigenere Cipher

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
main()
{
 int i,j,k,numstr[100],numkey[100],numcipher[100];
 char str[100],key[100];
 printf("Enter a string\n");
 gets(str);
//converting entered string to Capital letters
 for(i=0,j=0; i<strlen(str); i++)
 {
 if(str[i]!=' ')
 {
 str[j]=toupper(str[i]);
 j++;
 }
 }
 str[j]='\0';
 printf("Entered string is : %s \n",str);
//Storing string in terms of ascii
 for(i=0; i<strlen(str); i++)
 {
 numstr[i]=str[i]-'A';
 }
printf("Enter a key\n");
 gets(key);
 //converting entered key to Capital letters
 for(i=0,j=0; i<strlen(key); i++)
 {
 if(key[i]!=' ')
 {
 key[j]=toupper(key[i]);
 j++;
 }
 }
 key[j]='\0';
 //Assigning key to the string
 for(i=0; i<strlen(str);)
 {
 for(j=0; (j<strlen(key))&&(i<strlen(str)); j++)
 {
```

```
 numkey[i]=key[j]-'A';
 i++;
 }
 }
 for(i=0; i<strlen(str); i++)
 {
 numcipher[i]=numstr[i]+numkey[i];
 }
 for(i=0; i<strlen(str); i++)
 {
 if(numcipher[i]>25)
 {
 numcipher[i]=numcipher[i]-26;
 }
 }
 printf("Vigenere Cipher text is\n");
for(i=0; i<strlen(str); i++)
 {
 printf("%c",(numcipher[i]+'A'));
 }
 printf("\n");
}
```

Sample Output:

```
Enter a string
getupearly
Entered string is : GETUPEARLY
Enter a key
monday
Vigenere Cipher text is
SSGXPCMFYB
```