TUTORIAL / PRACTICAL NO.

INDEX

S.No	PRACTICALS	DATE	SIGN.	REMARKS
1.	Write program to implement caesar cipher technique.			
2.	Write program to implement polyalphabetic cipher.			
3.	Write program to implement vernam cipher.			
4.	Write program to implement Rail Fence technique			
5.	Write program to implement simple columnar technique			
6.	Write program to implement RSA algorithm			
7.	Write program to implement Euclidean algorithm			
8	Write steps to ensure security of web browser			

Objective: Write a program to implement caesar cipher technique.

Program:

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{
  cout << " Enter the message : ";
  cin.getline(msg,100);
  int i, j, len, ch, key;
  cout << " Enter key : ";
  cin >> key;
  len = strlen(msg);
  cout << " Enter your choice \n 1. Encryption
\n 2. Decryption \n ";
  cin >> ch;
  if (ch==1)
     char c;
    for (int i=0 ; msg[i] != '\0' ; ++i)
     {  c = msg[i];
      if( c >='a' && c<='z')
      {   c = c+key;
          c = c -'z' + 'a' - 1;
      }
          msg[i] = c;
    }
  else if (c >='A' && c<='Z')
{    c = c + key;
     If (c >'Z')
```

OUTPUT

for encryption

```
Enter the message : how ARE you
Enter Key : 3
Enter your choice
1. Encryption
2. Decryption
1
Encrypted message : KRZ DUH brx
```

for decryption

```
Enter the message : KRZ DUH brx
Enter Key : 3
Enter your choice
1. Encryption
2. Decryption
2
Decrypted message : how ARE you
```

```
    {   c = c - 'Z' + 'A' - 1 ;
    }   msg[i] = ch ;
  } }
  cout << " Encrypted message : " << msg << endl
} else if ( ch == 2)
{ char c ;
 for ( int i=0 ; msg[i] != '\0' ; ++i )
 {  c= msg[i];
  if ( c >='a' && c <='z') {
        c = c-key ;
        if (c<'a') {
          c = c + 'z' - 'a' + 1 ; }
          msg [i] = ch; }
  else if ( c >='A' && c<='Z') {
      c = c -key ;
      if ( c <'A')
       {  c = c+'Z' - 'A' + 1 ;
        } msg[i] = c ;
      }
    }
    cout << " Decrypted message : " << msg
      << endl ;
    }
}
```

OUTPUT

Enter the plain text
todayismybirthday
Enter key
november
new key is : novembernovembern
Cipher Text : gcyck jwdlpdvf ihrl
plain Text : todayismybirthday

TUTORIAL / PRACTICAL NO. 2

Objective :- Write a program to implement polyalphabetic cipher

Program :-

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main()
{ string pt, ct, key, ot;
  int i, j;
cout << "Enter plain text" << cout << endl;
cin >> pt;
cout << "Enter key" << endl;
cin >> key;
 j=0;
for( i= strlen(key); i< strlen(pt); i++)
 { if( j== strlen(key))
      { j=0; } key[j];
      key[i] =
      j++; }
  cout << "new key is :" << key;
for( i=0; i< strlen(pt); i++)
 { ct[i] = (((pt[i]-97) + (key[i]-97)) %.26) +97;
} cout << "cipher text :" << ct << endl;
for( i=0; i< strlen(ct); i++)
 { if (ct[i] < key[i])
    { ot[i] = 26 + ((ct[i]-97) -(key[i]-97)) + 97;
    } else ot[i] = (((ct[i]-97) -(key[i]-97))%26
                                          +97);
} cout << "plain text is :" << ot;
  return 0; }
```

OUTPUT

Plain Text : HELLO
Key : MONEY
Cipher Text : TSYPM
Message : HELLO

Objective: Write a program to implement Vernam Cipher.

Program:

```
#include <bits/stdc++.h>
using namespace std;
string Encryption(string text, string key)
{ string cipherText = " ";
  int cipher[key.length()];
for(int i=0; i<key.length(); i++)
  { cipher[i] = text.at(i) - 'A' + key.at(i) - 'A';
  }
 for(int i=0; i<key.length(); i++)
{   if(cipher[i] > 25)
     { cipher[i] = cipher[i] - 26;
     }
   }
for(int i=0; i<key.length(); i++)
  { int x = cipher[i] + 'A';
    ciphertext += (char)x;
  }return cipherText;
}
Static string Decryption(string s, string key)
{ string plainText = " ";
int plain[key.length()];
for(int i=0; i<key.length(); i++)
  {   if(plain[i]<0)
       { plain[i] = plain[i] + 26;
       }
  }
 for(int i=0, i<key.length(); i++)
```

```
    { int x = plain[i] + 'A';
      plainText += (char) x; }
return plainText;
}
int main()
{ string plainText = "HELLO";
  string key = "MONEY";
for (int i=0; i<plainText.length(); i++)
{ plainText[i] = toupper(plainText[i]);
}
for (int i=0; i<key.length(); i++)
  { key[i] = toupper(key[i]);
}string encryptedText = Encryption(plainText, key);
cout << "Cipher Text" << encryptedText << endl;
cout << "Plain Text" << Decryption(encryptedText,
  key);
return 0;
}
```

TUTORIAL / PRACTICAL NO. 4

Objective: Write a program to implement Rail Fence technique.

Program:

```
#include <bits/stdc++.h>
using namespace std;
string encryptRailFence(string text, int key)
{ char rail[key][(text.length())];
for(int i=0; i<key; i++)
  { for(int j=0; j<text.length(); j++)
      rail[i][j] = '\n';
bool dir_down = false;
int row=0, col=0;
for(int i=0; i<text.length(); i++)
 { if(row==0 || row==key-1)
     dir_down = !dir_down;
rail[row][col++] = text[i];
dir_down ? row++ : row--;
}
string result;
for(int i=0; i<key; i++)
  for(int j=0; j<text.length(); j++)
    if(rail[i][j] != '\n')
      result.push_back(rail[i][j]);
return result;
}
string decryptRailFence(string cipher, int key)
{ char rail[key][cipher.length()];
for(int i=0; i<key; i++)
    for(int j=0; j<cipher.length(); j++)
      rail[i][j] = '\n';
bool dir_down;
```

```
    int row = 0 , col = 0 ;
for ( int i = 0 ; i < cipher.length() ; i++)
 {  if ( row == 0)
          dir_down = true ;
       if ( row == key - 1)
          dir_down = false ;
   rail[row][col++] = '*' ;
   dir_down ? row ++ : row -- ;
 }
int index = 0 ;
for (int i=0 ; i< key ; i++)
    for ( int j=0 ; j<cipher.length() ; j++)
     if (rail[i][j] == '*' && index < cipher.length())
     rail[i][j] = cipher[index++] ;
string result ;
row=0, col=0 ;
for ( int i=0 ; i< cipher.length() ; i++)
  {  if (row == 0)
          dir_down = true ;
       if ( row == key - 1)
          dir_down = false ;
   if (rail[row][col] != '*')
     result.push_back (rail[row][col++]) ;
   dir_down ? row ++ : row -- ;
} return result ;
  }
int main ()
{
 cout << encryptRailfence("attack at once", 2)
    << endl ;
```

OUTPUT

Plain Text :

attack at once

defend the east wall

Cipher Text :

atc toctaka ne

dnhaveedtes aff tl

```
    cout << encryptRailFence("defend the east wall", 3) << endl;
    cout << decryptRailFence("atc toctaka ne", 2) << endl;
    cout << decryptRailFence("dnhaweedtees alf tl", 3) << endl;
    return 0;
}
```

TUTORIAL / PRACTICAL NO. 5

Objective : Write program to implement simple columnar transposition technique.

Program :

```
#include<bits/stdc++.h>
using namespace std;
string Encryption (int no_rows, int len_key, int
    len_msg, string msg, int col_val[])
{  int x=0;
   char enc_mat[no_rows+1][len_key];
   for (int i=0; i<no_rows+1; i++)
    {  for (int j=0; j<len_key; j++)
       {  if (x>= len_msg)
          {  enc_mat[i][j] ='_';
          }
          else {
             enc_mat[i][j] = msg[x];
          }
          x++; } }

int t=1;
 string cipher =" ";
 while (t<= len_key)
 {   for (int i=0; i<len_key ; i++)
     {  int k = col_val[i];
        if (k==t)
        {
           for(int j=0; j<no_rows+1 ; j++)
           {  cipher += enc_mat[j][i];
           } t++;
        }
     } }
     return cipher;
 }
```

OUTPUT

Encrypted Message: comwomomeretorhoo

Decrypted Message: comehometomarrow

```
string Decryption (int no_rows, int len_key)
{ char dec_mat [no_rows+1][len_key];
  int x=0, t=1;
  while (t<= len_key)
  { for (int i=0; i< len_key ; i++)
    { int k= col_val[i];
      if (k==t)
      { for (int j=0 ; j< no_rows+1 ; j++)
        { dec_mat[j][i] = cipher[x];
          x++;
        } t++; } } }
  string message = "";
for (int i=0 ; i< rows+1 ; i++)
  { for (int j=0; j< len_key ; j++)
    { if (dec_mat[i][j] == '_')
      { dec_mat[i][j] = ' ';
      }
      message += dec_mat[i][j];
    }
  } return message;
}
int main()
{ string msg = "come home tomorrow";
  string key = "HACK";
  int len_key = key.length();
  int len_msg = msg.length();
  int val = 1, count = 0, ind;
  int col_val [len_key];
```

```
while (count < len_key)
{  int min = 999;
   for(int i=0 ; i<len_key ; i++)
   {  if ((min > int(key[i])) && col_val[i]==0)
      {  min = int(key[i]);
         end = i;
      }
   }
   int no_rows = len_msg / len_key ;
   string cipher_text = " ";
   cipher_text = Encryption (no_rows, len_key,
len_msg, msg, col_val);
   cout << " Encrypted message : " << cipher_text
   << endl;
   string original_msg = " ";
originalmsg = Decryption (no_rows, len_key,
   cipher_text, col_val);
cout << " Decrypted message " << original_msg;
}
return 0;
}
```

TUTORIAL / PRACTICAL NO. 6

Objective: Write a program to implement simple RSA algorithm with small numbers.

Program:

```
#include<bits/stdc++.h>
using namespace std;
int gcd (int a, int h)
{ int temp;
  while (1)
  { temp = a % h;
    if (temp == 0)
        return h;
    a = h;
    h = temp;
  }
}
int main()
{ double p = 3;
  double q = 7;
  double n = p*q;
  double e = 2;
  double phi = (p-1) * (q-1);
  while (e < phi)
  { if (gcd(e, phi) == 1)
      break;
    else
      e++;
  }
  int k = 2;
  double d = (1 + (k*phi))/e;
  double msg = 12;
  cout << "Message data " << msg;
```

OUTPUT

Message data = 12.000000

Encrypted data = 3.000000

Original Message Sent = 12.000000

```
        double c = pow (msg, e);
        c = fmod (c,n);
cout << "Encrypted data :" << c;
    double m= pow (c, d);
     m= fmod (m,n);
  cout << "original Message Sent" << m;
return 0;
  }
```

OUTPUT

GCD (10, 15) = 5

GCD (35, 10) = 5

GCD (31, 2) = 1

UTORIAL / PRACTICAL NO. 7

Objective : Write a program to implement Euclidean algorithm.

Program :

```
#include <stdio.h>
#include <bits/stdc++.h>
using namespace std;
int gcd (int a, int b)
{   if (a == 0)
        return b;
    return gcd (b%a, a);
}
int main ()
{   int a = 10, b = 15;
cout << "GCD (" << a << ", " << b << ") = " <<
    gcd (a,b) << endl;
    a = 35, b = 10;
cout << "GCD (" << a << ", " << b << ") = " <<
    gcd (a,b) << endl;
a = 31, b = 2;
cout << "GCD (" << a << ", " << b << ") = " <<
gcd (a,b) << endl;
return 0;
}
```

TUTORIAL / PRACTICAL NO. 8

Objective: Write various steps to ensure security of any web browser. (Firefox, Google Chrome)

Firefox ⇒

STEPS

- Auto install updates ⇒ For both Mac and PC - go to Firefox menu > Preferences (Mac) options (PC) General tab > Firefox Update section. Select "Automatically install updates".
- Block unwanted pop-ups ⇒ For both Mac and PC - go to Firefox menu > Preferences (Mac) Options (PC) > Privacy + Security > Permission section. Check "Warn you when website try to install add-ons";
- Don't save passwords ⇒ For both Mac & PC - go to Firefox menu > Browser Privacy section. Uncheck the "Ask to save login & passwords for websites" box.
- Cookies + Site Data ⇒ Select "Custom" and set cookies to block "Third-party trackers". Also place checks to block Cryptominers + Fingerprinters.

Google Chrome ⇒

STERS

- Auto-download updates ⇒ To make sure that you're protected by latest security updates, google chrome automatically updates whenever it detects new version is available

- Block unwanted plugins → Go to settings > Unsandboxed plugins acces & turn on " Ask when a site wants to use a plugin to access your computer.

- Don't save passwords → Go to settings > Passwords & turn off " Offer to save passwords.

- Handle cookies → Turn on " Allow site to save and read cookie data (recommended) and " Block - Third party cookies ".

- Camera Acces → Go to Settings > Advanced > Content settings > Camera & turn on " Ask before accessing (recommended)".