DEEN DAYAL UPADHYAYA COLLEGE

UNIVERSITY OF DELHI

PRACTICAL FILE

SUBJECT: COMPUTER NETWORKS

COURSE: B.Sc. MATHEMATICAL SCIENCES

YEAR: THIRD

SEMESTER: SIXTH

Year: 2023-2024

SUBMITTED BY:	SUBMITTED TO:
Sneha Gupta	Mr. DEEPAK MITTAL
(21MTS5735)	(ASSISTANT PROFESSOR)

1. Write a HTML program to design a form which should allow to enter your personal data.
   (Hint: make use of text field, password field, e-mail, lists, radio buttons, checkboxes, submit button).

```
<html>
<head>
<title>Personal Data Form</title>
</head>
<body>

<h2>Enter Your Personal Data</h2>

<form action="#" method="post">
  <label for="name">Name:</label><br>
  <input type="text" id="name" name="name"><br><br>

  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email"><br><br>

  <label for="password">Password:</label><br>
  <input type="password" id="password" name="password"><br><br>
```

```
  <label for="gender">Gender:</label><br>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Male</label>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label><br><br>

  <label for="country">Country:</label><br>
  <select id="country" name="country">
    <option value="India">India</option>
    <option value="USA">USA</option>
    <option value="Canada">Canada</option>
    <option value="UK">UK</option>
    <option value="Australia">Australia</option>
  </select><br><br>

  <label for="interests">Interests:</label><br>
  <input type="checkbox" id="sports" name="interests" value="sports">
  <label for="sports">Sports</label>
  <input type="checkbox" id="music" name="interests" value="music">
  <label for="music">Music</label>
  <input type="checkbox" id="reading" name="interests"
value="reading">
  <label for="reading">Reading</label><br><br>
```

```
  <label for="comments">Comments:</label><br>
  <textarea id="comments" name="comments" rows="4" cols="50"></textarea><br><br>

  <input type="submit" value="Submit">
</form>

</body>
</html>
```

Enter Your Personal Data

Name:

Email:

Password:

Gender:
Male Female

Country:
India

Interests:
Sports Music Reading

Comments:

Submit

2. Write html code to generate following output.

- Coffee
- Tea
    - Black Tea
    - Green Tea
- Milk

```
<html>
<head>
<title>Beverage Menu</title>
</head>
<body>

<h2>Beverage Menu</h2>

<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
```

```
    <li>Black Tea</li>
    <li>Green Tea</li>
   </ul>
  </li>
  <li>Milk</li>
</ul>

</body>
</html>
```

Beverage Menu

- Coffee
- Tea
  - Black Tea
  - Green Tea
- Milk

3. Design an html form to take the information of a customer visiting a departmental store such as name, contact phone no, preferred days of purchasing, favourite item (to be selected from a list of items), suggestions etc. One should provide button to Submit as well as Reset the form contents.

```
<html>
<head>
<title>
```

```
Feedback Form
</title>
</head>
<body>
<h1> Customer Feedback </h1>
<form>
<h3> Name: <input> </h3>
<h3> Contact Number: <input type="Number"> </h3>
<h3> Preferred days of purchasing:</h3>
<h4> Monday <input type="checkbox"><br>
Tuesday <input type="checkbox"><br>
Wednesday <input type="checkbox"><br>
Thursday <input type="checkbox"><br>
Friday <input type="checkbox"><br>
Saturday <input type="checkbox"><br>
Sunday <input type="checkbox"> </h4>
<h3> Favourite Items:</h3>
<h4> Groceries <input type="checkbox"> <br>
Stationary <input type="checkbox"> <br>
Garments <input type="checkbox"> <br>
Footwear <input type="checkbox"> </h4>
<h3> Suggestion: <textarea> </textarea> </h3>
<h3> <input type="SUBMIT"> <input type="RESET"> </h3>
</form>
```

```
</body>
</html>
```

Customer Feedback

Name:

Contact Number:

Preferred days of purchasing:

- **Monday**
- **Tuesday**
- **Wednesday**
- **Thursday**
- **Friday**
- **Saturday**
- **Sunday**

Favourite Items:

- **Groceries**
- **Stationary**
- **Garments**
- **Footwear**

Suggestion:

Submit Reset

4. Design an html form to take the information of an article to be uploaded such as file path, author name, type (technical, literary, general), subject topic (to be selected from a list) etc. One should provide button to Submit as well as Reset the form contents.

```
<html>
<head>
<title>
Article Information
</title>
</head>
<body>
<h1> Article Information </h1>
<form>
 <h3> File: <input type="text" id="file" name="file"></h3>
 <h3> Author Name:<input type="text" id="author" name="author">
</h3>
 <h3> Type:</h3>
 <input type="radio" id="technical" name="type" value="technical">
 <label for="technical">Technical</label>
 <input type="radio" id="literary" name="type" value="literary">
 <label for="literary">Literary</label>
```

```
  <input type="radio" id="general" name="type" value="general">
  <label for="general">General</label><br>

  <h3>Subject:
<select id="subject" name="subject">
    <option value="technology">Technology</option>
    <option value="science">Science</option>
    <option value="health">Health</option>
    <option value="politics">Politics</option>
    <option value="economy">Economy</option>
  </select>
</h3><br>

  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>
</body>
</html>
```

Article Information

File:

Author Name:

Type:

○ Technical ○ Literary ○ General

Subject: Technology

Submit Reset

Design an HTML document using Table related tags align the images.


```
<html>
<head>
  <title>Company Logos</title>
</head>
<body>
  <table border="1">
```

```
    <tr>
      <td><img src="vodafone_logo.png" alt="Vodafone"></td>
      <td><img src="reliance_logo.png" alt="Reliance"></td>
      <td><img src="indicom_logo.png" alt="Indicom"></td>
      <td><img src="airtel_logo.png" alt="Airtel"></td>
    </tr>
    <tr>
      <td><img src="tata_docomo_logo.png" alt="Tata Docomo"></td>
      <td><img src="table_with_images_logo.png" alt="Table With
Images"></td>
      <td><img src="aircel_logo.png" alt="AIRCEL"></td>
    </tr>
    <tr>
      <td><img src="uninor_logo.png" alt="uninor"></td>
      <td><img src="mtc_logo.png" alt="MTC"></td>
      <td><img src="hutch_logo.png" alt="HUTCH"></td>
      <td><img src="tata_docomo_logo.png" alt="Tata Docomo"></td>
    </tr>
  </table>
</body>
</html>
```

5. Develop static pages (using only HTML) of an online Book store.The

website should consist of following pages.

- Home page
- Registration and user Login
- User profile page
- Books catalog
- Shopping cart
- Payment by credit card Order Conformation

Index.html

```
<html>
<head>
   <title>Online Bookstore - Home</title>
</head>
<body>
   <h1>Welcome to Our Online Bookstore</h1>
   <p>Find your favorite books and enjoy reading!</p>
   <nav>
      <a href="login.html">Login</a> |
      <a href="register.html">Register</a> |
      <a href="catalog.html">Browse Catalog</a>
   </nav>
</body>
</html>
```

Login.html

```
<html>
<head>
   <title>Login - Online Bookstore</title>
</head>
<body>
   <h1>Login</h1>
   <form action="login_process.php" method="POST">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username" required><br><br>
      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required><br><br>
      <input type="submit" value="Login">
   </form>

   <p>New user? <a href="register.html">Register here</a></p>
</body>
</html>
```

Register.html

```
<html>
```

```
<head>
    <title>Register - Online Bookstore</title>
</head>
<body>
    <h1>User Registration</h1>
    <form action="registration_process.php" method="POST">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br><br>
        <input type="submit" value="Register">
    </form>

    <p>Already have an account? <a href="login.html">Login</a></p>
</body>
</html>
```

Profile.html

```
<html>
<head>
    <title>User Profile - Online Bookstore</title>
```

```
</head>
<body>
    <h1>User Profile</h1>
    <p>Welcome, [Username]!</p>
    <p>Email: [Email]</p>
    <nav>
        <a href="catalog.html">Browse Catalog</a> |
        <a href="cart.html">Shopping Cart</a> |
        <a href="logout.html">Logout</a>
    </nav>
</body>
</html>
```

Catalog.html

```
<html>
<head>
    <title>Books Catalog - Online Bookstore</title>
</head>
<body>
    <h1>Books Catalog</h1>
    <ul>
        <li>Book 1</li>
        <li>Book 2</li>
        <li>Book 3</li>
```

```
        <!-- Add more books as needed -->
    </ul>
    <nav>
        <a href="cart.html">Shopping Cart</a> |
        <a href="logout.html">Logout</a>
    </nav>
</body>
</html>
```

Cart.html

```
<html>
<head>
    <title>Shopping Cart - Online Bookstore</title>
</head>
<body>
    <h1>Shopping Cart</h1>
    <ul>
        <li>Book 1 - $10</li>
        <li>Book 2 - $15</li>
        <!-- Add more items as needed -->
    </ul>
    <p>Total: $25</p>
    <form action="payment.html">
        <input type="submit" value="Proceed to Checkout">
```

```
    </form>
    <nav>
        <a href="catalog.html">Continue Shopping</a> |
        <a href="logout.html">Logout</a>
    </nav>
</body>
</html>
```

Payment.html

```
<html>
<head>
    <title>Payment - Online Bookstore</title>
</head>
<body>
    <h1>Payment</h1>
    <form action="confirmation.html">
        <label for="card_number">Credit Card Number:</label>
        <input type="text" id="card_number" name="card_number" required><br><br>
        <label for="expiry_date">Expiry Date:</label>
        <input type="text" id="expiry_date" name="expiry_date" placeholder="MM/YY" required><br><br>
        <label for="cvv">CVV:</label>
        <input type="text" id="cvv" name="cvv" required><br><br>
```

```
        <input type="submit" value="Pay Now">
    </form>
</body>
</html>
```

Confirmation.html

```
<html>
<head>
    <title>Order Confirmation - Online Bookstore</title>
</head>
<body>
    <h1>Order Confirmation</h1>
    <p>Your order has been successfully placed!</p>
    <p>Order Number: 123456</p>
    <p>Total Amount: $25</p>
    <p>Estimated Delivery Date: [Date]</p>
    <p>Thank you for shopping with us!</p>
</body>
</html>
```

Welcome to Our Online Bookstore

Find your favorite books and enjoy reading!

Login | Register | Browse Catalog

Login

Username:

Password:

Login

New user? Register here

User Registration

Username:

Email:

Password:

Register

Already have an account? Login

Books Catalog

- Book 1
- Book 2
- Book 3

Shopping Cart | Logout

Shopping Cart

- Book 1 - $10
- Book 2 - $15

Total: $25

Proceed to Checkout

Continue Shopping | Logout

Payment

Credit Card Number:

Expiry Date: MM/YY

CVV:

Pay Now

Order Confirmation

Your order has been successfully placed!

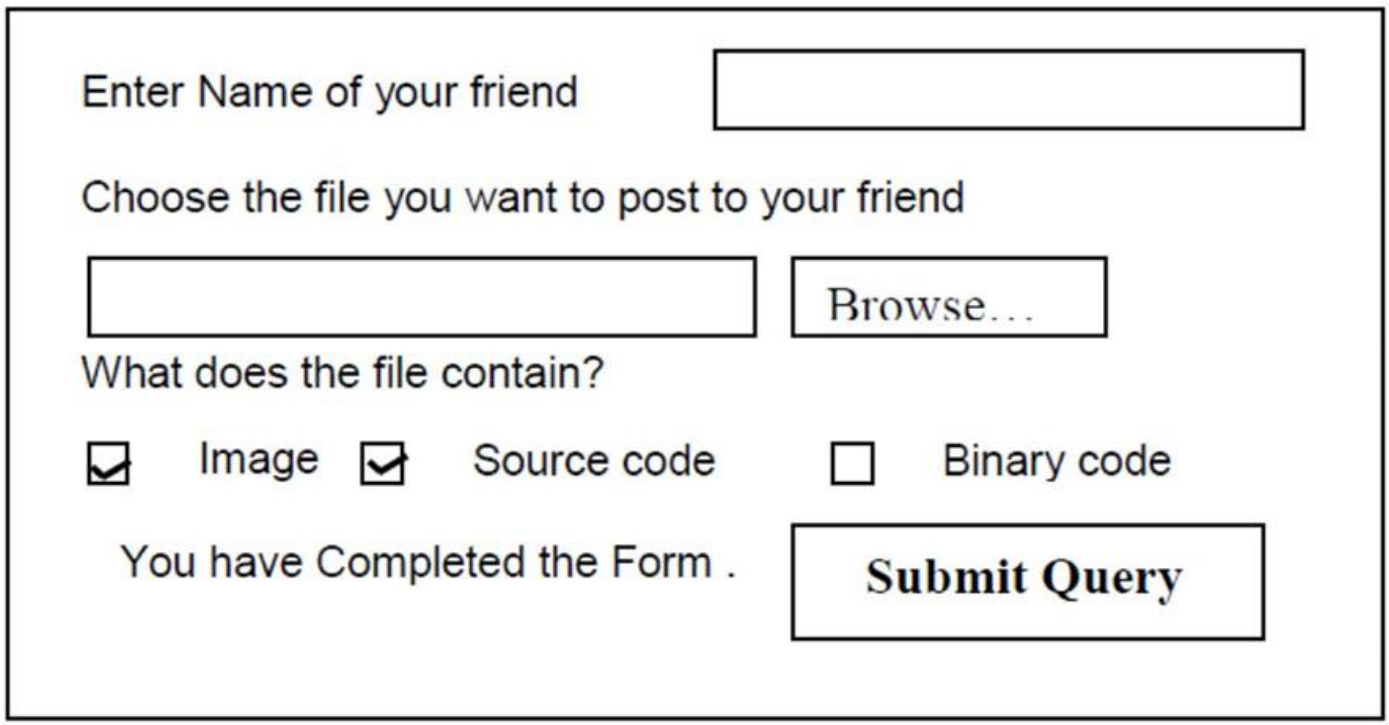
Order Number: 123456

Total Amount: $25

Estimated Delivery Date: [Date]

Thank you for shopping with us!

6. Write a HTML code to generate following output.



```
<html>
<head>
<title>
Question 6
</title>
</head>
<body>
<form>
<h3>Enter Name of your friend <input></h3>
<h3>Choose the file you want to post to your friend</h3> <h3><input type="text"><input type="file"></h3>
<h3>What does the file contain?</h3>
<h3><input type="checkbox"> Image <input type="checkbox"> Source code <input type="checkbox"> Binary code </h3> <h3>You have Completed the Form <input type= "button" value= "Submit Query"> </h3>
</form>
</body>
</html>
```

NETWORK ALGORITHMS PRACTICAL LIST

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.

```
 #include<iostream>
using namespace std;
string xorfun( string encoded , string crc)
{
   int crclen = crc.length();

   for ( int i = 0 ; i <= (encoded.length() - crclen) ; )
   {
      for( int j=0 ; j < crclen ; j++)
      {
         encoded[i+j] = encoded[i+j] == crc[j] ? '0' : '1' ;
      }
```

```
    for( ; i< encoded.length() && encoded[i] != '1' ; i++) ;

    }

    return encoded;
}

int main()
{
    string data , crc , encoded = "";
    cout<<endl<<"-----------Sender Side --------------"<<endl;
    cout<<"Enter Data bits: "<<endl;
    cin>>data;

    cout<<"Enter Generator: "<<endl;
    cin>>crc;

    encoded += data;

    int datalen = data.length();
    int crclen = crc.length();

    for(int i=1 ; i <= (crclen - 1) ; i++)
        encoded += '0';
```

```
    encoded = xorfun(encoded , crc);

    cout<<"Checksum generated is: ";
    cout<<encoded.substr(encoded.length() - crclen + 1)<<endl<<endl;
    cout<<"Message to be Transmitted over network: ";
    cout<<data + encoded.substr(encoded.length() - crclen + 1);
    cout<<endl<<"----------Reciever Side----------"<<endl;

    cout<<"Enter the message recieved: "<<endl;
    string msg;
    cin>>msg;

    msg = xorfun( msg , crc);
    for( char i : msg.substr(msg.length() - crclen + 1))
       if( i != '0' )
          {
             cout<<"Error in communication "<<endl;
             return 0;
          }

    cout<<"No Error !"<<endl;
    return 0;
}
```

Output:

```
----------Sender Side --------------
Enter Data bits:
11011011
Enter Generator:
1011
Checksum generated is: 110

Message to be Transmitted over network: 11011011110
--------------Reciever Side----------------
Enter the message recieved:
11011011110
No Error !
```

2. Simulate and implement stop and wait protocol for noisy channel.

```
#include <iostream>
#include <time.h>
#include <unistd.h>
using namespace std;

int main() {
   int frames[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
   unsigned long seconds = 5000, to;
   bool delay = false;

   srand(time(NULL));

   cout << "Sender has to send frames : ";
   for (int i = 0; i < 10; i++)
      cout << frames[i] << " ";
   cout << endl << "timeout : 5s";
```

```
cout << endl << endl << "Sender\t\t\t\t\tReceiver" << endl;
int count = 0;
do {
    bool timeout = false;
    cout << "Sending Frame : " << frames[count]<< "       ";
    cout.flush();
    cout << "\t\t";

    to = rand() % 5500;
    usleep(to * 1000);

    if (to <= seconds)
    {
        cout << "Received Frame : " << frames[count] << "       ";
        if (delay)
        {
            cout << "Duplicate";
            delay = false;
        }
        cout << endl;
        count++;
    }
    else
    {
```

```
            cout << "---" << endl;
            cout << "Timeout" << endl;
            timeout = true;
        }

        to = rand() % 5500;
        usleep(to * 1000);
        if (to > seconds)
        {
            cout << "Delayed Ack" << endl;
            count--;
            delay = true;
        }
        else if (!timeout)
            cout << "Acknowledgement : " << frames[count-1] << endl;
    } while (count != 10);
    return 0;
}
```

```
Sender has to send frames : 1 2 3 4 5 6 7 8 9 10
timeout : 5s

Sender                                  Receiver
Sending Frame : 1                       Received Frame : 1
Acknowledgement : 1
Sending Frame : 2                       Received Frame : 2
Acknowledgement : 2
Sending Frame : 3                       Received Frame : 3
Acknowledgement : 3
Sending Frame : 4                       Received Frame : 4
Acknowledgement : 4
Sending Frame : 5                       Received Frame : 5
Acknowledgement : 5
Sending Frame : 6                       Received Frame : 6
Acknowledgement : 6
Sending Frame : 7                       Received Frame : 7
Acknowledgement : 7
Sending Frame : 8                       Received Frame : 8
Acknowledgement : 8
Sending Frame : 9                       Received Frame : 9
Acknowledgement : 9
Sending Frame : 10                      Received Frame : 10
Acknowledgement : 10
```

3. Simulate and implement go back n sliding window protocol.

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

#include<ctime>

#define ll long long int
using namespace std;

void transmission(ll & i, ll & N, ll & tf, ll & tt) {
```

```
  while (i <= tf) {
    int z = 0;
    for (int k = i; k < i + N && k <= tf; k++) {
      cout << "Sending Frame " << k << "..." << endl;
      tt++;
    }
    for (int k = i; k < i + N && k <= tf; k++) {
      int f = rand() % 2;
      if (!f) {
        cout << "Acknowledgment for Frame " << k << "..." << endl;
        z++;
      } else {
        cout << "Timeout!! Frame Number : " << k << " Not Received" <<
endl;
        cout << "Retransmitting Window..." << endl;
        break;
      }
    }
    cout << "\n";
    i = i + z;
  }
}

int main() {
```

```
  ll tf, N, tt = 0;
  srand(time(NULL));
  cout << "Enter the Total number of frames : ";
  cin >> tf;
  cout << "Enter the Window Size : ";
  cin >> N;
  ll i = 1;
  transmission(i, N, tf, tt);
  cout << "Total number of frames which were sent and resent are : " <<
tt <<
    endl;
  return 0;
}
```

Output

```
Enter the Total number of frames : 8
Enter the Window Size : 3
Sending Frame 1...
Sending Frame 2...
Sending Frame 3...
Timeout!! Frame Number : 1 Not Received
Retransmitting Window...

Sending Frame 1...
Sending Frame 2...
Sending Frame 3...
Timeout!! Frame Number : 1 Not Received
Retransmitting Window...

Sending Frame 1...
Sending Frame 2...
Sending Frame 3...
Acknowledgment for Frame 1...
Timeout!! Frame Number : 2 Not Received
Retransmitting Window...
```

```
Sending Frame 4...
Acknowledgment for Frame 2...
Acknowledgment for Frame 3...
Timeout!! Frame Number : 4 Not Received
Retransmitting Window...

Sending Frame 4...
Sending Frame 5...
Sending Frame 6...
Timeout!! Frame Number : 4 Not Received
Retransmitting Window...

Sending Frame 4...
Sending Frame 5...
Sending Frame 6...
Acknowledgment for Frame 4...
Timeout!! Frame Number : 5 Not Received
Retransmitting Window...

Sending Frame 5...
Sending Frame 6...
Sending Frame 7...
Acknowledgment for Frame 5...
```

```
Acknowledgment for Frame 6...
Acknowledgment for Frame 7...

Sending Frame 8...
Timeout!! Frame Number : 8 Not Received
Retransmitting Window...

Sending Frame 8...
Acknowledgment for Frame 8...

Total number of frames which were sent and resent are : 23
```

4. Simulate and implement selective repeat sliding window protocol.

/*We are assuming the window size =4. both at sender and receiver's side.

There might be different order of transmission but the no of retransmissions are same.

The order of transmission will depend upon the acknowledgement timer and Timeout timer*/

#include<iostream>

```
 int tmp1, tmp2, tmp3, tmp4, tmp5, i, windowsize = 4, noofPacket,
morePacket;

using namespace std;
int main()
{

   char c;
   int receiver(int);
   int simulate(int);
   int negack(int);

   for(int i = 0;i < 10;i++)
      rand();
      noofPacket = rand()%10;
   cout<<"Number of frames is: "<<noofPacket;
   morePacket = noofPacket;

   while(morePacket >= 0)
   {
      tmp1 = simulate(windowsize);
      windowsize -= tmp1;
      tmp4 += tmp1;
```

```
if(tmp4 > noofPacket)
    tmp4 = noofPacket;

for(i = noofPacket - morePacket; i <= tmp4; i++)
cout<<"\nSending Frame "<<i;

tmp2 = receiver(tmp1);
tmp3 += tmp2;

if(tmp3 > noofPacket)
   tmp3 = noofPacket;

tmp2 = negack(tmp1);
tmp5 += tmp2;

if(tmp5 != 0)
{
   cout<<"\nNo acknowledgement for the frame "<<tmp5;
   cout<<"\nRetransmitting frame "<<tmp5;
}
morePacket -= tmp1;

if(windowsize <= 0)
```

```
        windowsize = 4;
    }
    cout<<"\n Selective Repeat Protocol Ends. All packets are successfully transmitted.";
}

int receiver(int tmp1)
{
    int i;
    for(i = 0;i < 5;i++)
        rand();
        i = rand() % tmp1;
        return i;
}

int negack(int tmp1)
{
    int i;
    for(i = 0;i < 5;i++)
        rand();
        i = rand() % tmp1;
        return i;
}
```

```
int simulate(int windowsize)
{
   int tmp1, i;
   for(i = 0;i < 5;i++)
      tmp1 = rand();
   if(tmp1 == 0)
      tmp1 = simulate(windowsize);
      i = tmp1 % windowsize;
   if(i == 0)
      return windowsize;
   else
      return tmp1 % windowsize;
}
```

Output:

```
Number of frames is: 5
Sending Frame 0
Sending Frame 1
Sending Frame 2
Sending Frame 3
No acknowledgement for the frame 2
Retransmitting frame 2
Sending Frame 3
Sending Frame 4
No acknowledgement for the frame 2
Retransmitting frame 2
Sending Frame 4
Sending Frame 5
No acknowledgement for the frame 5
Retransmitting frame 5
 Selective Repeat Protocol Ends. All packets are successfully transmitted.
```

5. Shortest Path algorithm.

```
#include <iostream>
#include <limits.h>
using namespace std;

#define V 9

int minDistance(int dist[], bool sptSet[]) {
   int min = INT_MAX, min_index;
   for (int v = 0; v < V; v++) {
      if (sptSet[v] == false && dist[v] <= min) {
         min = dist[v];
         min_index = v;
      }
   }
   return min_index;
}

void printPath(int parent[], int j) {
   if (parent[j] == -1) {
      cout << j;
      return;
   }
   printPath(parent, parent[j]);
```

```
    cout << " -> " << j;
}

void printSolution(int dist[], int parent[], int src) {
    cout << "Vertex \t Distance \t Path" << endl;
    for (int i = 0; i < V; i++) {
        cout << i << " \t\t " << dist[i] << " \t\t ";
        printPath(parent, i);
        cout << endl;
    }
}

void dijkstra(int graph[V][V], int src) {
    int dist[V];
    int parent[V];
    bool sptSet[V];
    for (int i = 0; i < V; i++) {
        dist[i] = INT_MAX;
        sptSet[i] = false;
        parent[i] = -1;
    }
    dist[src] = 0;
    for (int count = 0; count < V - 1; count++) {
        int u = minDistance(dist, sptSet);
```

```
        sptSet[u] = true;
        for (int v = 0; v < V; v++) {
            if (!sptSet[v] && graph[u][v] && dist[u] != INT_MAX &&
                    dist[u] + graph[u][v] < dist[v]) {
                dist[v] = dist[u] + graph[u][v];
                parent[v] = u;
            }
        }
    }
    printSolution(dist, parent, src);
}

int main() {
    int graph[V][V] = {
        { 0, 4, 0, 0, 0, 0, 0, 8, 0 },
        { 4, 0, 8, 0, 0, 0, 0, 11, 0 },
        { 0, 8, 0, 7, 0, 4, 0, 0, 2 },
        { 0, 0, 7, 0, 9, 14, 0, 0, 0 },
        { 0, 0, 0, 9, 0, 10, 0, 0, 0 },
        { 0, 0, 4, 14, 10, 0, 2, 0, 0 },
        { 0, 0, 0, 0, 0, 2, 0, 1, 6 },
        { 8, 11, 0, 0, 0, 0, 1, 0, 7 },
        { 0, 0, 2, 0, 0, 0, 6, 7, 0 }
    };
```

```
    int source = 0;
    dijkstra(graph, source);
    return 0;
}
```

Output:

```
Vertex   Distance        Path
0               0               0
1               4               0 -> 1
2               12              0 -> 1 -> 2
3               19              0 -> 1 -> 2 -> 3
4               21              0 -> 7 -> 6 -> 5 -> 4
5               11              0 -> 7 -> 6 -> 5
6               9               0 -> 7 -> 6
7               8               0 -> 7
8               14              0 -> 1 -> 2 -> 8
```