以下題目均按學號規則推算出指定的題數來做，每題 **5** 分，只寫出中文得 **2** 分，說明正確得 **4** 分，詳細完整得 **5** 分。
題號計算錯誤的話，答案非常完美也只有 **3** 分。

名詞解釋：**(**共 **19** 題，取 **6** 題，請先寫出你應做的題號**)**

- 0. **IP**
- 1. **TCP**
- 2. **UDP**
- 3. **HTTP**
- 4. **HTML**
- 5. **CSS**
- 6. **Socket**
- 7. **DNS**
- 8. **Internet**
- 9. **Web**
- 10. **Crawler**
- 11. **Web Server**
- 12. **Browser**
- 13. **Client**
- 14. **Server**
- 15. **ipconfig**
- 16. **URL**
- 17. **Thread**
- 18. **Deadlock**

程式解釋：**(共 13 題，取 4 題，請先寫出你應做的題號)**

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;

 public class UdpClient {
   public static void Main(string[] args) {
0.    IPEndPoint ipep = new IPEndPoint(IPAddress.Parse(args[0]), 5555);
1.    Socket server = new Socket(AddressFamily.InterNetwork,
                              SocketType.Dgram, ProtocolType.Udp);
2.    while(true) {
3.      string input = Console.ReadLine();
4.      if (input == "exit") break;
5.      server.SendTo(Encoding.UTF8.GetBytes(input), ipep);
      }
6.    server.Close();
  }
}

public class UdpServer {
  public static void Main() {
    IPEndPoint ipep = new IPEndPoint(IPAddress.Any, 5555);
    Socket newsock = new Socket(AddressFamily.InterNetwork,
                  SocketType.Dgram, ProtocolType.Udp);
7.    newsock.Bind(ipep);
8.    IPEndPoint sender = new IPEndPoint(IPAddress.Any, 0);
9.    EndPoint Remote = (EndPoint)(sender);
    while(true) {
10.     byte[] data = new byte[1024];
11.     int recv = newsock.ReceiveFrom(data, ref Remote);
12.     Console.WriteLine(Encoding.UTF8.GetString(data, 0, recv));
      }
    }
}
```

程式解釋：**(共 31 題，取 10 題，請先寫出你應做的題號)**

```csharp
using System;
using System.Collections.Generic;
using System.Text;
using System.Net;
using System.Net.Sockets;
using System.IO;
using System.Threading;

class ChatBox {
  int port = 20;

  public static void Main(String[] args) {
0.   ChatBox chatBox = new ChatBox();
     if (args.Length == 0)
1.     chatBox.ServerMain();
     else
2.     chatBox.ClientMain(args[0]);
  }

3. public void ServerMain() {
4.  IPEndPoint ipep = new IPEndPoint(IPAddress.Any, port);
5.  Socket newsock = new Socket(AddressFamily.InterNetwork,
       SocketType.Stream, ProtocolType.Tcp);
6.  newsock.Bind(ipep);
     newsock.Listen(10);
7.  Socket client = newsock.Accept();
8.  new TcpListener(client);
9.  newsock.Close();
  }

10. public void ClientMain(String ip) {
     IPEndPoint ipep = new IPEndPoint(IPAddress.Parse(ip), port);
11.    Socket server = new Socket(AddressFamily.InterNetwork,
                  SocketType.Stream, ProtocolType.Tcp);
12.    server.Connect(ipep);
13.    new TcpListener(server);
14.    server.Shutdown(SocketShutdown.Both);
     server.Close();
  }
}

15. public class TcpListener {
   Socket socket;
   Thread inThread, outThread;
   NetworkStream stream;
   StreamReader reader;
   StreamWriter writer;
```

```
16. public TcpListener(Socket s) {
      socket = s;
      stream = new NetworkStream(s);
17.   reader = new StreamReader(stream);
18.   writer = new StreamWriter(stream);
19.   inThread = new Thread(new ThreadStart(inLoop));
20.   inThread.Start();
21.   outThread = new Thread(new ThreadStart(outLoop));
22.   outThread.Start();
23.   inThread.Join();
    }

24. public void inLoop() {
25.    while (true) {
26.      String line = reader.ReadLine();
27.      Console.WriteLine("收到：" + line);
      }
    }

    public void outLoop() {
28.    while (true) {
29.      String line = Console.ReadLine();
30.      writer.WriteLine(line);
       writer.Flush();
      }
    }
}
```