金門大學 資訊工程系 101 學年度下學期 網路程式設計 期中考 出題者：陳鍾誠
考生姓名：　　學號：　　得分：

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 : Internet protocol** (網路協定)，核心為由數字所組成的網址，IPv4 用 32bit 指定網址，通常分成 4 組 8bit ，例如：210.37.135.24 就是一個 IPv4 的網址。IP 協定層確保每台電腦都有一個唯一不衝突的網址。
1. **TCP : Transmission Control Protocol** (傳送控制協定)，Internet 上用來傳輸資料的協定，此協定可確保訊息封包按順序傳送，如果有封包遺失的情況，則會重新傳送封包，然後在交給上層程式時，封包看來就像按照順序傳送一般。TCP 通常用在必須要高可靠度的資料傳輸，例如 Web 的 HTTP 協定，用來傳輸網頁等應用上。
2. **UDP**：**User Datagram Protocol** (使用者資料協定)，Internet 上用來傳輸資料的協定，此協定並不確保封包的抵達與順序，封包可能會遺失、或者先送的封包可能會後到。UDP 通常用在需要快速傳送，但不需要按照順序的情況，例如多媒體影片或聲音的傳遞上。
3. **HTTP : HyperText Transfer Protocol** (超文字傳輸協定), Internet/Web 上用來傳輸網頁 (含影像與檔案) 的協定， HyperText 原本指的是 HTML，後來 HTTP 已經不限於傳 HTML 了。HTTP 的表頭是 ASCII 文字模式，採用 one touch 的「一次性」傳輸模式，瀏覽器連上後收取一個網頁或檔案，然後就立刻斷線。等到下次要另一個網頁時，再進行另一次連線。對於那些想從瀏覽器中傳給 Server 的資料而言，可以用 get 或 post 等兩種傳輸模式，get 會放在網址內進行傳輸，而 post 則會放在表頭尾部進行傳輸。
4. **HTML : HyperText Markup Language** (超文字標記語言)，也就是網頁的格式，採用 <tag> … </tag> 的形式標註文字的語言，例如 <html><body>Yahoo! </body></html> 就是一個 HTML 的範例，其中的 <a>... 標記代表了超連結，這也是為何稱為超文字的原因。
5. **CSS : Cascading Style Sheets** (層疊樣式表)，是用來對 HTML 進行呈現方式描述的語言，例如我們可以用 a { color:red } 來將所有的超連結都設定為紅色字體，您可以用 CSS 對 HTML 中的任何標記進行呈現方式描述，以便讓網頁更加好看，而且透過套用相同的 CSS 也可以讓網站具有統一的顯示風格。
6. **Socket : Socket** 一詞原指插座，但在網路程式中用來指稱一套由插座所啟發設計出來的網路函式庫，原先由柏克萊大學發展出來，將網路存取包裝成類似檔案串流的讀寫方式，讓網路的存取變得更容易且一致。C/C++/C#/Java 等各語言當中也都發展出了各自的 Socket 函式庫。
7. **DNS : Domain Name System** (網域名稱系統)，用來將網域 (像是 www.yahoo.com.tw) 轉譯為 IP 網址 (像是 203.188.197.200) 的系統，而其運作的核心 Domain Name Server 也通常被簡稱為 DNS。
8. **Internet :** 網際網路 (互聯網)，原先由美國國防部 DoD 所發展出來，用來傳遞訊息的網路，後來進一步成為全球性的網路，採用 TCP/IP 的架構，這種方式以封包 (Packet) 為訊息傳輸單位，透過分散式的傳遞方式，讓訊息傳遞到指定 IP 的目標電腦中，是目前使用最廣的網路系統。
9. **Web : Web** 一詞原指蜘蛛網，在電腦領域是 World Wide Web 的簡稱，指的是在 Internet 之上透過 HTTP/HTML/URL/WebServer/Browser 所架構出來的一套系統，這套系統讓大家可以透過瀏覽器上網看到各式各樣的網站，這套系統原本於 1990 年由 Tim Burner Lee 開始設計，後來迅速被全世界所接受並強化，現在已經是一個全球所有電腦使用者都經常使用的系統了。
10. **Crawler :** 網路爬蟲，又稱 Spider，是一種抓取網頁的程式，其原理是透過「抓取網頁、取出超連結得到更多網址，然後再抓取這些網址所對應的網頁」的方式。這種方式可以抓取指定網站或全世界的網頁，這也是 Google 等搜尋引擎背後的重要技術之一。
11. **Web Server :** 網站伺服器，是 Web 構成的伺服端元件，用來提供「HTML、影像、檔案」等訊息給瀏覽器閱讀，當瀏覽器發出連線請求時，WebServer 就會取出對應的網頁或檔案傳回給瀏覽器。
12. **Browser :** 瀏覽器，是 Web 的客戶端元件，透過傳送網址與表頭資訊給 WebServer，以取得網頁或檔案，並呈現在電腦螢幕上的一種程式。
13. **Client :** 客戶端，指網路連線上發出請求的一端，可以與使用者互動後，將使用者的請求傳送給 Server 並取得回傳訊息後呈現給使用者看。
14. **Server**：伺服端，指網路連線上接收請求的一端，可以接收客戶端傳來的訊息，然後加以解釋處理後，回傳

客戶端所需的訊息給 **Client**。

15. **ipconfig :** 是 **Windows** 上的一個命令列指令，可以用來查看本機的 **IP**、**MAC Address** 等資訊。
16. **URL : Uniform Resource Locator (**全球資源定位器**)**，簡單來說就是網址，也就是您在瀏覽器上方所看到的網頁位址，例如：**http://en.wikipedia.org/wiki/User_Datagram_Protocol** 就是一個網址。
17. **Thread :** 線程 **(**執行緒**)**，是一種程式的次單位，一個主程式可以讓許多個副程式同時執行，這些同時執行的副程式就稱為 **Thread**，這種方式可以讓程式執行時具有某種程度的獨立性，卻又可以讓線程之間互相共享資料。
18. **Deadlock :** 死結，當很多線程同時執行時，可能會因為鎖定 **lock** 動作而造成死結，其原因是某現成鎖定特定資源後，又要求另一資源，而對方也做類似的鎖定動作，因而造成我等你，你等我的循環情況，導致雙方互相鎖死，動彈不得的窘境。

考生姓名：　　學號：

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));
     }
   }
}
```

考生姓名：　　學號：

程式解釋：(共 31 題，取 10 題，請先寫出你應做的題號)

```
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();
      }
   }
}
```