001

針對以下程式紅色部分的程式碼，寫上詳細的註解

```
01 //server.c
02 #include <stdio.h>
03 #include <stdlib.h>
04 #include <string.h>
05 #include <unistd.h>
06 #include <sys/types.h>
07 #include <sys/socket.h>
08 #include <netinet/in.h>
09
10 double getCard();
11
12 int main(int argc , char *argv[]){
13     char receiveMessage[256] = {};
14     char message[256] = {"Hi, this is server.\n"};
15     double player_score = 0;
16     int sockfd = 0, clientSockfd = 0;
17     sockfd = socket(AF_INET , SOCK_STREAM , 0);
18     if (sockfd == -1){            printf("Fail to create a socket.");       }
19     struct sockaddr_in serverInfo,clientInfo;
20     int addrlen = sizeof(clientInfo);
21
22     bzero(&serverInfo,sizeof(serverInfo));
23     serverInfo.sin_family = AF_INET;
24     serverInfo.sin_addr.s_addr = INADDR_ANY;
25     serverInfo.sin_port = htons(8700);
26     bind(sockfd,(struct sockaddr *)&serverInfo,sizeof(serverInfo));
27     listen(sockfd, 8);
28     clientSockfd = accept(sockfd,(struct sockaddr*) &clientInfo, &addrlen);
29
30     send(clientSockfd, message, sizeof(message),0);
31     while(1){
32
33         recv(clientSockfd, receiveMessage, sizeof(receiveMessage),0);
34         printf("Client said: %s\n",receiveMessage);
35
```

```
36              if(strcmp("more",receiveMessage)==0){
37                  player_score = player_score + getCard();
28                  if(player_score<10.5){
29                      sprintf(message,"%lf",player_score);
40                  }else if (player_score==10.5){
41                      strcpy(message,"you win, 10.5 get");
42                  }else{
43                      strcpy(message,"you lose , greater than 10.5");
44                  }
45                  send(clientSockfd, message, sizeof(message),0);
46
47              }
48              else if(strcmp("end",receiveMessage)==0){
49                  break;
50              }
51          }
52
53          printf("%s, close Socket\n", receiveMessage);
54          close(sockfd);
55          return 0;
56      }
57
58      double getCard() {
59          srand( time(NULL));
60          /* 指定亂數範圍 */
61          int min = 1;
62          int max = 13;
63          /* 產生 [min , max] 的整數亂數 */
64          int card = rand() % (max - min + 1) + min;
65          double point = (double)card;
66          if (point >10.0) point = 0.5;
67          return point;
68      }
69
70      //client.c
71      #include <stdio.h>
72      #include <stdlib.h>
73      #include <string.h>
```

```
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
int main(int argc , char *argv[]) {
    char message[256] = {"Hi Server, this is client.\n"}, receiveMessage[256] = {};
    int sockfd = 0;
    sockfd = socket(AF_INET , SOCK_STREAM , 0);
    if (sockfd == -1){          printf("Fail to create a socket.");     }
    struct sockaddr_in info;
    bzero(&info,sizeof(info));
    info.sin_family = PF_INET;
    info.sin_addr.s_addr = inet_addr("127.0.0.1");
    info.sin_port = htons(8700);
    int err = connect(sockfd,(struct sockaddr *)&info,sizeof(info));
    if(err==-1){          printf("Connection error");     }
    send(sockfd, message, sizeof(message),0);
    while(1){
        recv(sockfd, receiveMessage, sizeof(receiveMessage),0);
        printf("Server said: %s\n", receiveMessage);
        if(strcmp("end",receiveMessage)==0 ||strcmp("you win, 10.5 get",receiveMessage)==0
                                        || strcmp("you lose , greater than 10.5",receiveMessage)==0 ){
            break;
        }
        printf("Input 'more' to get more card, input 'end' to stop the program. \n");
        printf("Please input message send to sever:\n");
        scanf("%s",&message);
        send(sockfd, message, sizeof(message),0);
        if(strcmp("end",message)==0){
            strcpy(receiveMessage,"end");
            break;
        }
    }
    printf("%s, close Socket\n", receiveMessage);
    close(sockfd);
    return 0;
}
```

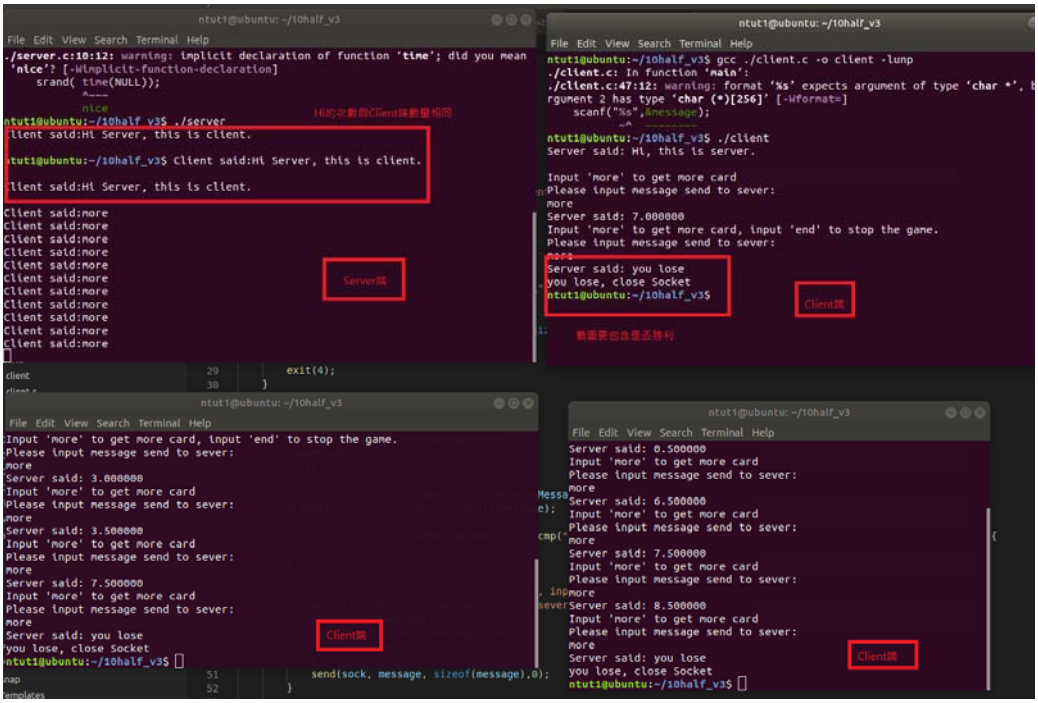
002

請改寫上一題的程式，使其可以讓多個 Client 端，連線到 Server 端進行遊戲，請遵照以下說明進行改寫：

(1) 本題使用一台虛擬機即可。
(2) 刪除”end”指令，本題將不需要使用到該指令，請將 Client 端的使用者提示改為” Input 'more' to get more card”。
(3) 任一個 Client 端與 Sever 端建立連線後，Server 端使用 Fork()來新增 Child process，讓每一個 Clinet 端與 Sever 端的連線透過不同的 process 進行。
(4) 每一個 Client 端所進行的遊戲，都是獨立的，不同 Client 端的牌面大小並不會互相干擾。
(5) 只有兩個條件可以結束遊戲，牌面大小剛好等於 10.5 或者爆牌【即:牌面大小超過 10.5】，前者為遊戲勝利，後者為失敗。
(6) 每一個 Client 端可以輸入”more”指令來得到更多牌，直到爆牌或勝利為止。
(7) 若 Client 端輸入其他指令【即:”Hi Server, this is client.\n” 與 ” more”以外】，Sever 都不會理會。
(8) 截圖並上傳 Server 端與多個 Client 端的視窗畫面。

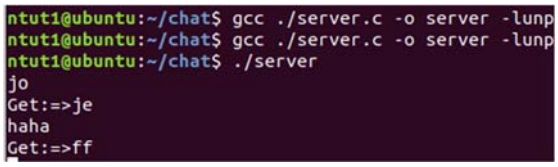
其他注意事項：

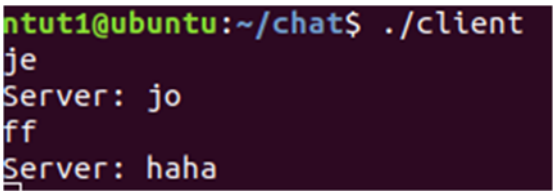
- 本題要需多個 Client 端進行遊戲，請截圖多個 Client 端(至少三個)與 Server 端之畫面，須包含所有 Client 端的牌面大小與是否為贏家。
- 本題只需要兩個檔案，分別為 server.c 與 client.c。
- 本題執行結果參考



003
Socket 對話

開啟二個虛擬機，一個當 Server，一個當 Client。
執行順序
1. client 端使用者輸入資料，傳送到 Server 端。
2. Server 端使用者輸入資料，傳送到 Client 端。
3. client 端使用者輸入資料，傳送到 Server 端。
4. Server 端使用者輸入資料，傳送到 Client 端。
...
Client 或 Server 端，使用者輸入 'Bye'結束





請助教檢查執行畫面與 Code 後，將執行結果上傳 client/ server code。

004

透過 Tcp 傳圖片

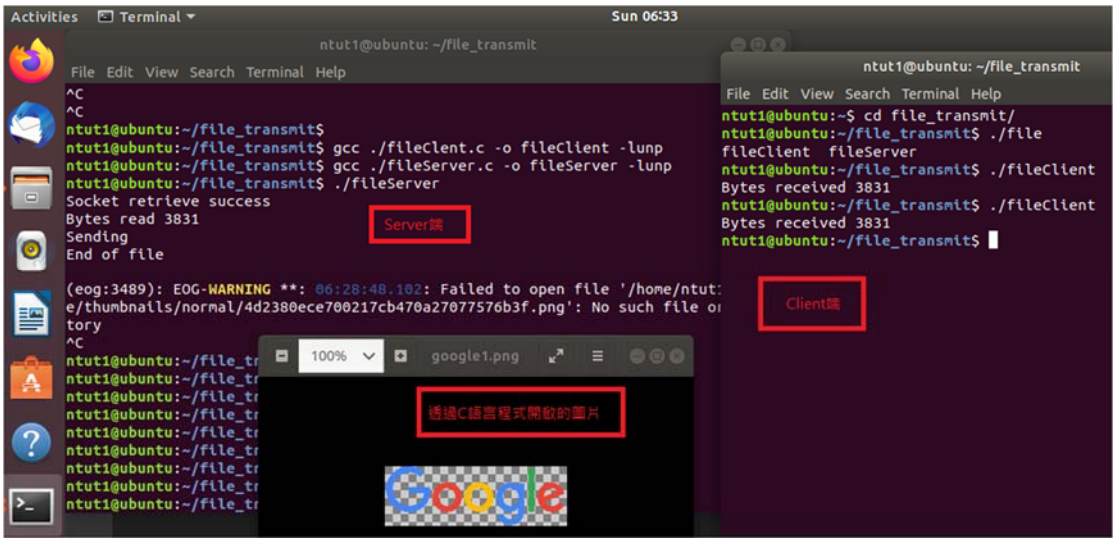
(1) 請參考上課講義檔案傳輸部分。

(2) 請參考上課講義 exec 部分，讓 TCP 程式可以呼叫 Ubuntu 系統命令。

(3) 本題使用一台虛擬機即可。

(4) 實作一個傳送資料的 TCP 程式的 Sever 端與 Client 端，Server 端傳遞一張 png 圖片，Client 端程式收到該檔案後，Client 端內部包含此 c 語言指令以開啟圖片：【execl("/usr/bin/xdg-open", "xdg-open", the_file, (char *)0);】，請替換”the_file”部分為你的圖片，完成本題 TCP 程式撰寫。

(5) 截圖並上傳 Server 端與 Client 端還有開啟照片的視窗。



005

三個虛擬機，兩個 Client 與一個 sever，計算自然對數

(1) 請參考上課講義，計算 PI 部分。

(2) 一台 client，要求計算自然對數到第 N 項。

(3) 二台 Server 計算自然對數，各分 N/2 給兩台 Server 計算。

(4) 公式請見下圖：

$$e = 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!} + \cdots = \sum_{n=0}^{\infty} \frac{1}{n!}.$$

(5) 截圖並上傳 Server 端與 Client 端的視窗。

006

開啟三台 VM，一台 client，要求計算 PI 到 2N 項，二台 Server 分工計算 PI。

1.client 需使用 fork 對二台 server 進行連線。

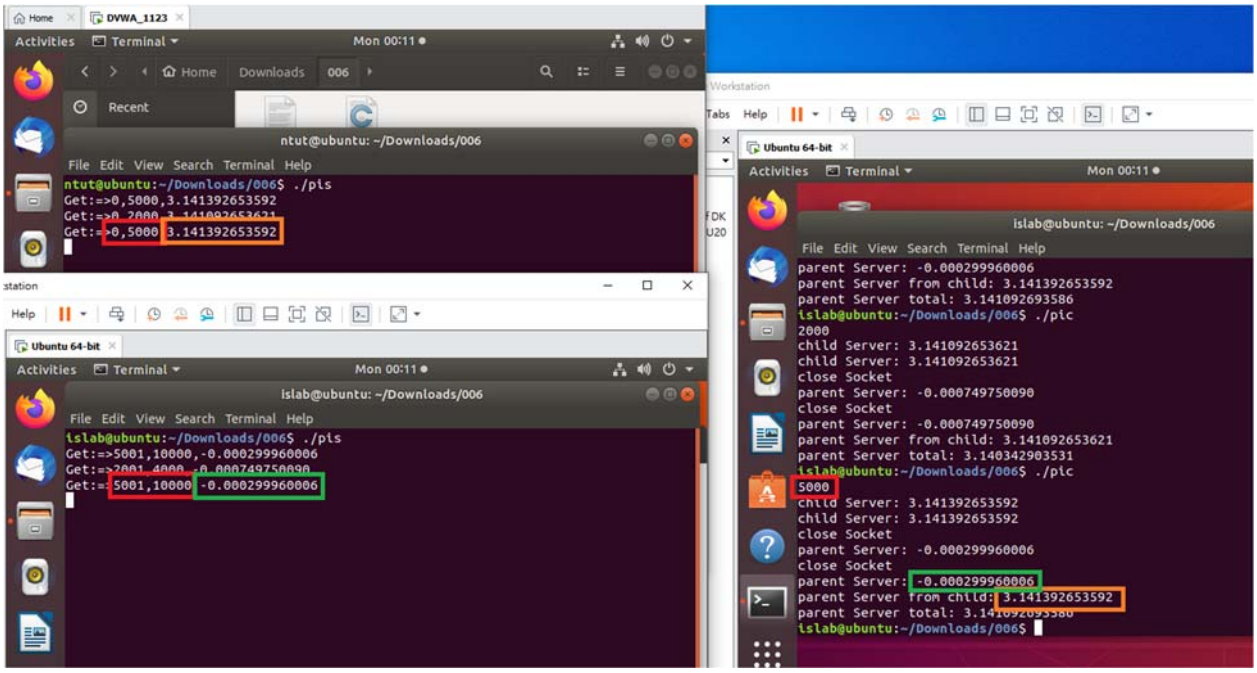
2.第一台 server 計算 0 ~ N 項。

3.第二台 server 計算 N+1 ~ 2N 項。

4.二台 server 在收到 client 的連線請求後，各自需在終端機顯示負責計算的項數範圍(當 client 輸入 N=5000，server1 終端機顯示 0, 5000，server2 終端機顯示 5001, 10000) 。

5.client 端查看執行結果。

請助教檢查執行畫面與 Code 後，上傳 client / server code。



007

開啟三台 VM，一台 client，要求計算 PI 到 2N 項，二台 Server 分工計算 PI。

1.client 需使用 fork 對二台 server 進行連線。

2.在 client 執行 Wireshark。

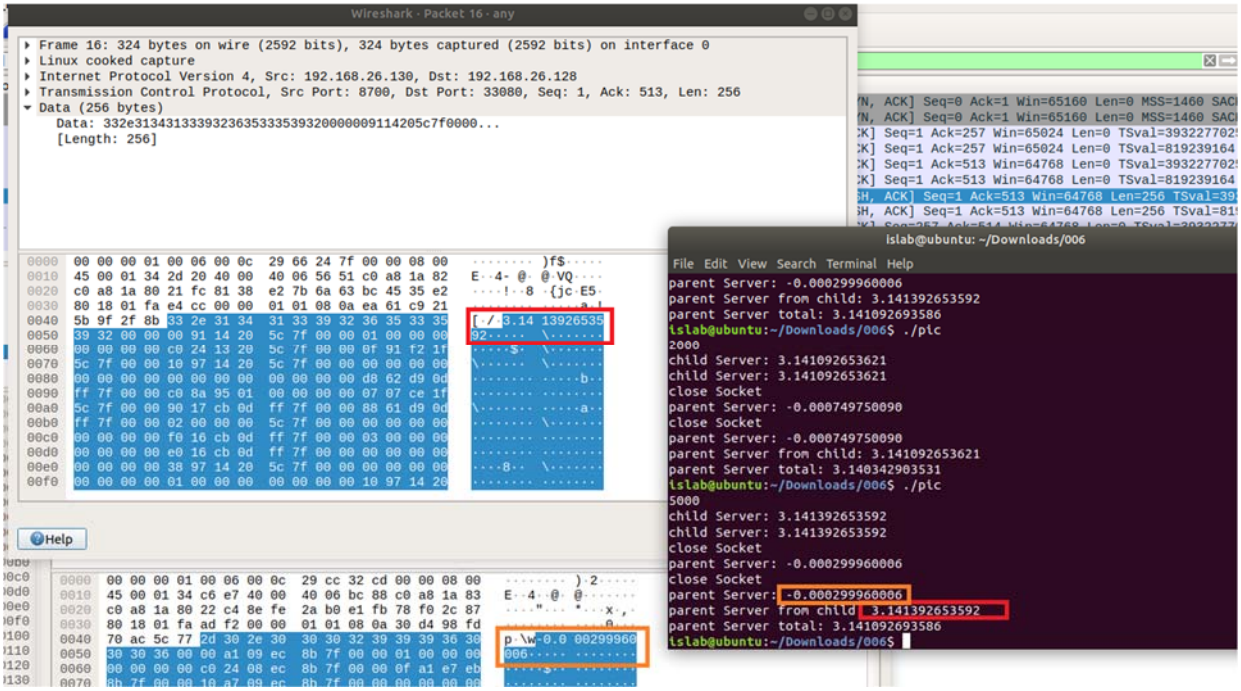
3.第一台 server 計算 0～N 項。

4.第二台 server 計算 N+1～2N 項。

5.client 端查看執行結果。

6.在 client 中使用 Wireshark 查看二台 server 分別回傳的封包資訊，資訊須含有 server 計算出的數值。

請助教檢查執行畫面，上傳 client 終端機執行結果以及在 wireshark 查看到 server 回傳結果的封包內容截圖。



008

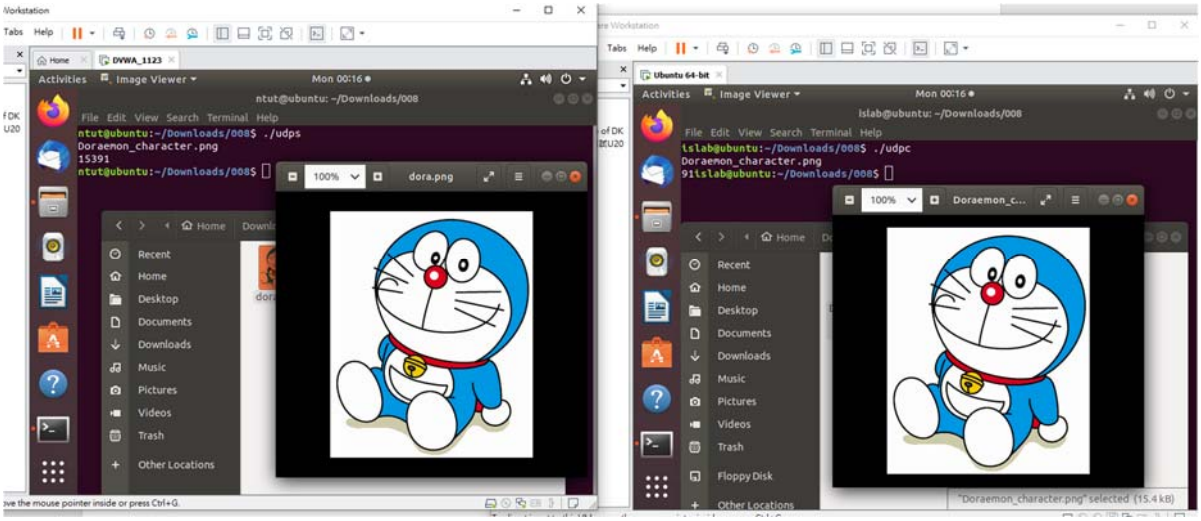
開啟兩個虛擬機，一個當 Server，一個當 Client。

1.下載圖片【https://upload.wikimedia.org/wikipedia/en/b/bd/Doraemon_character.png】，圖片檔名存為 dora.png，置於 client 端虛擬機。

2.程式執行，在 client 端讀取圖片 dora.png，運用 UDP 通訊協定，要分段傳輸，每次傳 256 byte，將檔案傳輸到 server 端。

3.server 端接收檔案，查看圖片是否正確。

請助教檢查執行畫面與 Code 後，上傳 client / server code。



009

開啟兩個虛擬機，一個當 Server，一個當 Client

1. 下載圖片【https://upload.wikimedia.org/wikipedia/en/b/bd/Doraemon_character.png】，圖片檔名存為 dora.png，置於 client 端虛擬機。

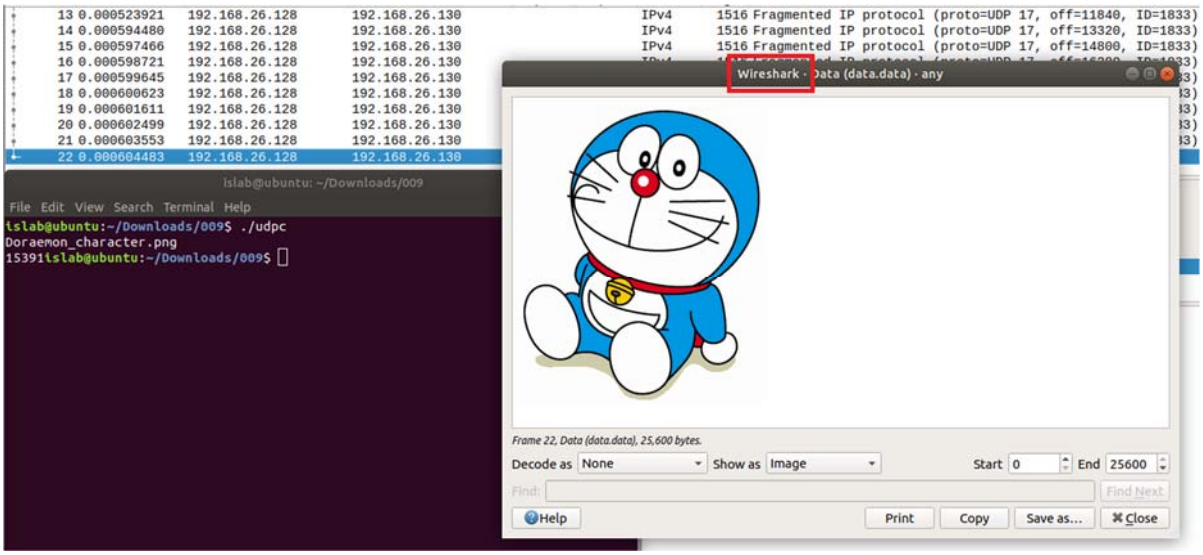
2.在 client 執行 Wireshark。

3.程式執行，在 client 端讀取圖片 dora.png，運用 UDP 通訊協定，不分段一次將檔案傳輸到 server 端。

4.server 端接收檔案，查看圖片是否正確。

5.在 client 中使用 Wireshark 捕捉傳送到 server 的封包，並使用查看圖片的方式查看圖片 dora。

請助教檢查執行畫面，上傳使用 Wireshark 查看圖片 dora 的截圖。



010
使用 C 或 python，實做一個 Class A～C 的子網域計算機，
使用者輸入：ip 地址(Class A～C)及子網路遮罩，形式為(xxx.xxx.xxx.xxx/xx)
程式需輸出：
1.這個 IP 為哪一類(Class)
2.這個子網路遮罩下所切分出的網段數量
3.這個 IP 所在的 Host 中可用的 IP 範圍

範例輸入 1：
140.124.123.123/16
範例輸出 1：
B
1
140.124.0.1~140.124.255.254

範例輸入 2：
191.254.254.254/20
範例輸出 2：
B
16
191.254.240.1~191.254.255.254

範例輸入 3：
192.0.25.54/20
範例輸出 3：
C
16
192.0.16.1~192.0.31.254