

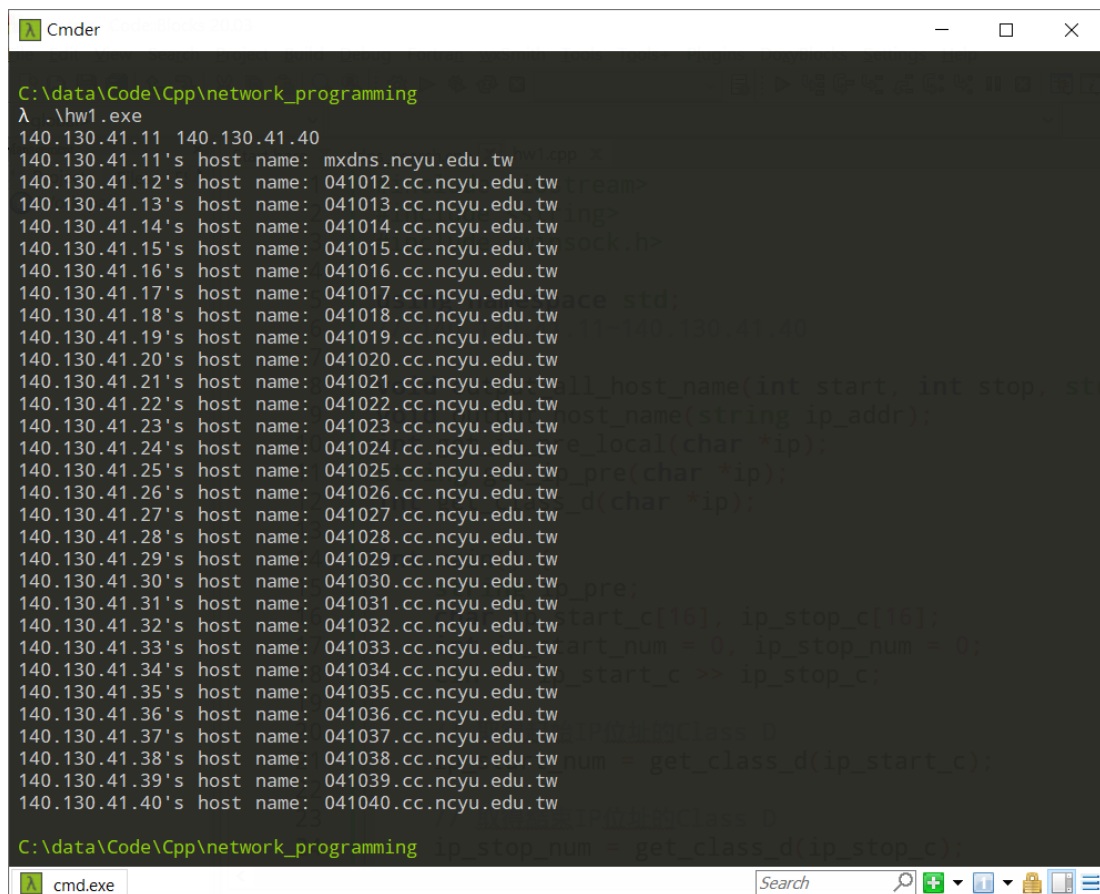
網路程式設計 HW-連續查詢 IP

一、 程式說明

輸入兩個 IP 位址，兩 IP 位址需為 IPv4 格式，起始 IP 位址與結束 IP 位址須為同 Class D 區段，且起始 IP 位址需小於結束 IP 位址，兩 IP 位址中間以空格或換行分隔，此程式會查詢此段位址之間所有 IP 位址所對應的 hostname，例如輸入 140.130.41.11 140.130.41.12 會查詢出兩筆紀錄，分別為 140.130.41.11's host name: mxdns.ncyu.edu.tw 及 140.130.41.12's host name: 041012.cc.ncyu.edu.tw。

二、 執行畫面

執行 hw1.exe，輸入 140.130.41.11 140.130.41.40，以下為執行結果。



```

C:\data\Code\Cpp\network_programming
λ .\hw1.exe
140.130.41.11 140.130.41.40
140.130.41.11's host name: mxdns.ncyu.edu.tw
140.130.41.12's host name: 041012.cc.ncyu.edu.tw
140.130.41.13's host name: 041013.cc.ncyu.edu.tw
140.130.41.14's host name: 041014.cc.ncyu.edu.tw
140.130.41.15's host name: 041015.cc.ncyu.edu.tw
140.130.41.16's host name: 041016.cc.ncyu.edu.tw
140.130.41.17's host name: 041017.cc.ncyu.edu.tw
140.130.41.18's host name: 041018.cc.ncyu.edu.tw
140.130.41.19's host name: 041019.cc.ncyu.edu.tw
140.130.41.20's host name: 041020.cc.ncyu.edu.tw
140.130.41.21's host name: 041021.cc.ncyu.edu.tw
140.130.41.22's host name: 041022.cc.ncyu.edu.tw
140.130.41.23's host name: 041023.cc.ncyu.edu.tw
140.130.41.24's host name: 041024.cc.ncyu.edu.tw
140.130.41.25's host name: 041025.cc.ncyu.edu.tw
140.130.41.26's host name: 041026.cc.ncyu.edu.tw
140.130.41.27's host name: 041027.cc.ncyu.edu.tw
140.130.41.28's host name: 041028.cc.ncyu.edu.tw
140.130.41.29's host name: 041029.cc.ncyu.edu.tw
140.130.41.30's host name: 041030.cc.ncyu.edu.tw
140.130.41.31's host name: 041031.cc.ncyu.edu.tw
140.130.41.32's host name: 041032.cc.ncyu.edu.tw
140.130.41.33's host name: 041033.cc.ncyu.edu.tw
140.130.41.34's host name: 041034.cc.ncyu.edu.tw
140.130.41.35's host name: 041035.cc.ncyu.edu.tw
140.130.41.36's host name: 041036.cc.ncyu.edu.tw
140.130.41.37's host name: 041037.cc.ncyu.edu.tw
140.130.41.38's host name: 041038.cc.ncyu.edu.tw
140.130.41.39's host name: 041039.cc.ncyu.edu.tw
140.130.41.40's host name: 041040.cc.ncyu.edu.tw

```

三、 程式碼

```

#include <iostream>
#include <string>
#include <winsock.h>

```

```

using namespace std;

void output_all_host_name(int start, int stop, string pre);
void output_host_name(string ip_addr);
int get_ip_pre_local(char *ip);
string get_ip_pre(char *ip);
int get_class_d(char *ip);

int main() {
    string ip_pre;
    char ip_start_c[16], ip_stop_c[16];
    int ip_start_num = 0, ip_stop_num = 0;
    cin >> ip_start_c >> ip_stop_c;

    // 取得起始 IP 位址的 Class D
    ip_start_num = get_class_d(ip_start_c);

    // 取得結束 IP 位址的 Class D
    ip_stop_num = get_class_d(ip_stop_c);

    // 取得 IP 位址 Class D 前面的部分
    ip_pre = get_ip_pre(ip_start_c);

    // 輸出所有查詢結果
    output_all_host_name(ip_start_num, ip_stop_num, ip_pre);
    return 0;
}

/*
 * 輸出所有查詢結果
 * int start    => 起始 IP 位址的 Class D
 * int stop     => 起始 IP 位址的 Class D
 * string pre   => IP 位址 Class D 前面的部分
 */
void output_all_host_name(int start, int stop, string pre) {
    string ip_addr;
    for(int i=start; i<=stop; i++) {
        ip_addr = pre + to_string(i);
    }
}

```

```

        output_host_name(ip_addr);
    }
}

/*
 * 輸出單一查詢結果
 * string ip_addr => 要查詢的 IP 位址
 */
void output_host_name(string ip_addr) {
    WSADATA wsadata;
    int n_result = WSAStartup(0x101, (LPWSADATA) &wsadata);
    struct in_addr sAddr;

    LPHOSTENT hp;

    if(n_result !=0) {
        // 初始化失敗: 輸出錯誤代碼
        cout << "WSA Initialization failed: " << n_result;
        WSACleanup;
        return;
    }

    // 帶入 IP 位址 (char array 型態)
    sAddr.s_addr = inet_addr(ip_addr.c_str());

    // 由 IP 位址取得 hostname
    hp = gethostbyaddr((LPSTR) &sAddr, sizeof(sAddr), AF_INET);

    if(hp == NULL) {
        // 未取得到結果
        cout << "WSA Error: " << WSAGetLastError() << endl;
    } else {
        cout << ip_addr << "'s host name: " << hp->h_name << endl;
    }
}

/*
 * 取得 IP 位址 Class D 的起始位置

```

```

    * char *ip      => IP 位址
    */
int get_ip_pre_local(char *ip) {
    int i = 0, count_dot = 0, num = 0;
    while(true) {
        if(ip[i] == '.') {
            count_dot++;
            if(count_dot == 3) {
                return i;
            }
        }
        i++;
    }
}

/*
 * 取得 IP 位址 Class D 前的部分
 * char *ip      => IP 位址
 */
string get_ip_pre(char *ip) {
    string ip_str = "";
    for(int i = 0; i <= get_ip_pre_local(ip); i++) {
        ip_str += ip[i];
    }
    return ip_str;
}

/*
 * 取得 IP 位址 Class D 的部分
 * char *ip      => IP 位址
 */
int get_class_d(char *ip) {
    bool do_loop = true;
    int i = 0, count_dot = 0, num = 0;
    while(do_loop) {
        if(ip[i] == '.') {
            count_dot++;
            if(count_dot == 3) {

```

```

while(true) {
    i++;
    if(ip[i] != '\0') {
        num = num*10 + (ip[i] - 48);
    } else {
        do_loop = false;
        break;
    }
}
}
}
i++;
}
return num;
}
// 140.130.41.11 140.130.41.13

```

四、心得

今天是第二次上這堂網路程式設計的課程，老師介紹了 socket，也教我們實作了如何使用 WinSock 函式庫來撰寫程式，我使用 C++ 語言撰寫出了一個可查詢 IP 位址所對應的 hostname 的程式，在程式環境設定時，我們遇到了一些困難，老師告訴我們須連結動態函式庫 -lws2_32 後才解決此問題，另外 Code::Blocks IDE 需另外連結 -lws2_32 函式庫才可順利編譯，若單純使用 command line 利用 g++ 來編譯則不用連結 -lws2_32 函式庫。

老師這次出的作業，我認為能讓我熟練文字的處理，畢竟有一段時間沒有寫 C++ 語言了，讓我喚回我的手感，今天的程式不算是太困難，但是需花費一些時間，我將所有程式碼包成 function，方便功能的實作與使用，此外，也順便加入部分註解，以利之後回來看程式碼時能快速地了解。