Homework Tutorial

Data Structure

Fall 2017

作業規則與注意事項

- 作業、程式碼嚴禁抄襲
  - 抄襲者與被抄襲者都一律0分計算！
- 作業繳交期限為兩個星期，**不接受遲交**
- 沒有屍體分數，遲交零分且不得補交
- **使用NTHU Online Judge**
  - 目的：為了讓同學有訂正的機會
  - 實際作業分數，以繳交期限前最後上傳版本為準
  - 助教會使用上傳版本檢查code，有爭議再送OJ測試為準

作業規則與注意事項

- NTHU OJ: https://acm.cs.nthu.edu.tw/
- 每次作業共有四筆測資，以通過筆數計算分數，如下

	1/4	2/4	3/4	4/4
成績	60	75	90	100

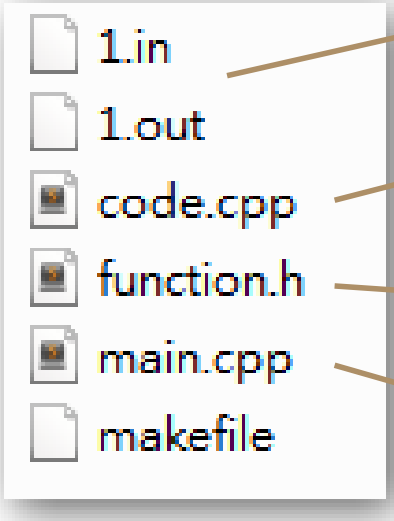
- 不會公布測資，只會公布測試結果。

Requirements

- We use NTHU Online Judge to test all students' homework.
- TA will use g++ 5.4.0 to compile all students' homework again
- Your mission is to inherit the class that we have defined.
- You need to implement your code in the child class "Implement"

File Format

In the zip file, you can see following things:



The sample input and output.

Implement your **class Implement** in the file **code.cpp**

The pre-defined class you are asked to **inherit** in your class Implement is in **function.h**.

This file is the way we **test your code**.

Methods You Need to Implement

function.h | code.cpp | main.cpp | makefile

```
#ifndef _FUNCTION_H_
#define _FUNCTION_H_
class abstractAdd
{
public:
    abstractAdd(){};
    ~abstractAdd(){};
    virtual int myAdd(int a, int b) = 0;

};

class Implement;

#endif
```

Implement the pure virtual function(s) in Implement class (code.cpp)

How to Implement

unction.h | code.cpp | main.cpp | makefile

```
#include "function.h"

class Implement : public abstractAdd
{
private:
    //int mystuff;
public:
    Implement(){
        //mystatus = 123;
    }
    ~Implement(){};
    int myAdd(int a, int b){
```

You have to **#include “function.h”.**

You have to define the class **Implement**, and it need to **inherit** the class given to you.

You have to **override the pure virtual function**

How to Test Your Code?

- We provide a basic testing file, you can use it to evaluate correctness of your code.
- Make sure your code can pass basic testing.

How to Test Your Code?

```
int main(int argc, char const *argv[])
{
    abstractAdd& ra = *new Implement();
    string input;
    while(getline(std::cin, input)){
        istringstream testcase(input);
        int a,b;
        testcase >> a;
        testcase >> b;
        cout << ra.myAdd(a,b) << endl;
    }

    return 0;
}
```

This is main.cpp

Test your implement function here

Test in NTHU OJ

- https://acm.cs.nthu.edu.tw/problem/11569/
- Submit your **code.cpp**
- Status
  - All Accepted
  - Not Accepted
  - Runtime Error
  - Compile Error
  - ……

Test in NTHU OJ

11569 - DS HW_Practice

Status | Limits Submit

Click to submit your code

Description

Please implement the add function in the class implement.

int implement::add(int,int)

which will return the sum of 2 integer variable.

Note that you don't need to consider the overflow of int data type.

You must #include "function.h"

Input

2 interger **a**, **b**.

Output

Sum of **a** and **b**.

Sample Input Download

```
1 2
```

Sample Output Download

```
3
```

Test in NTHU OJ

Submit Code 11569 - DS HW_Practice

Problem ID

11569

Language

C++: -O2 -lm -std=c++

Backend:
gcc:
g++:

Write your code here

Code

```
1 Code goes here...
```

We support Sublime key bindings!

Or upload your code.cpp

Browse

Submit

Test in NTHU OJ

Username	Problem	Status
Your Account	11569 - DS HW_Practice	All Accepted (1/1)
Your Account	11569 - DS HW_Practice	All Accepted (1/1)

Don’t try to modified the following file, we may replace them and use stricter testing.

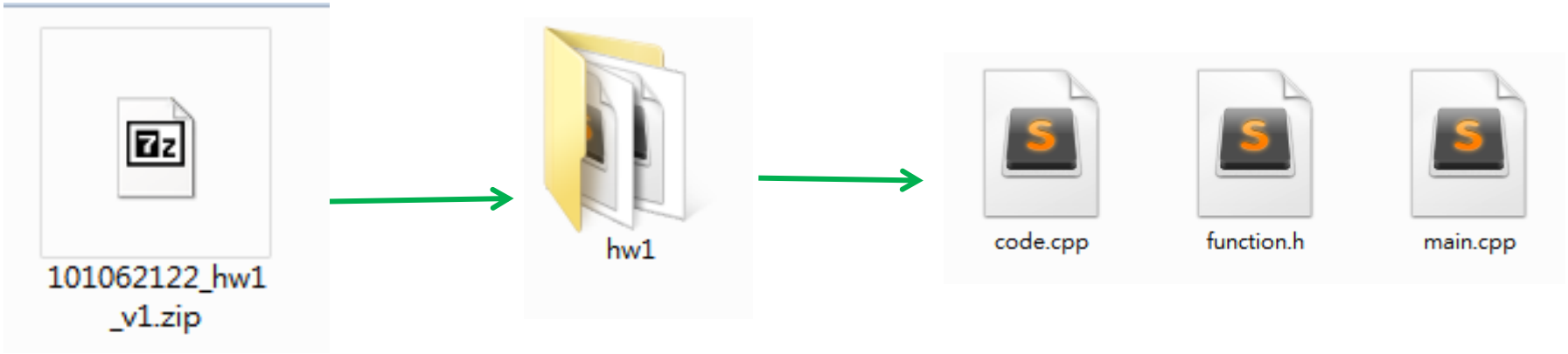
- function.h
- main.cpp
- Name of class “Implement”

作業繳交格式

- 以Homework1為例，請將所有檔案放到名為hw1的資料夾之下，並且將hw1打包成一個.zip檔
- Zip檔的命名格式
  - iLMS帳號_作業.zip
  - 例: 101062122_hw1.zip
  - 所有英文檔案名皆須為小寫
  - 勿使用zip以外的格式，如tar,rar,7z等

作業繳交格式

- hw資料夾底下的預設檔案結構請勿變動



作業繳交格式

- 不要使用額外的檔案，因為makefile不會處理額外的檔案
- 上傳之前可以盡量使用NTHU Online Judge作測試
- 請使用**和quiz相同的帳號**(DSK+學號)到OJ送出

In Windows

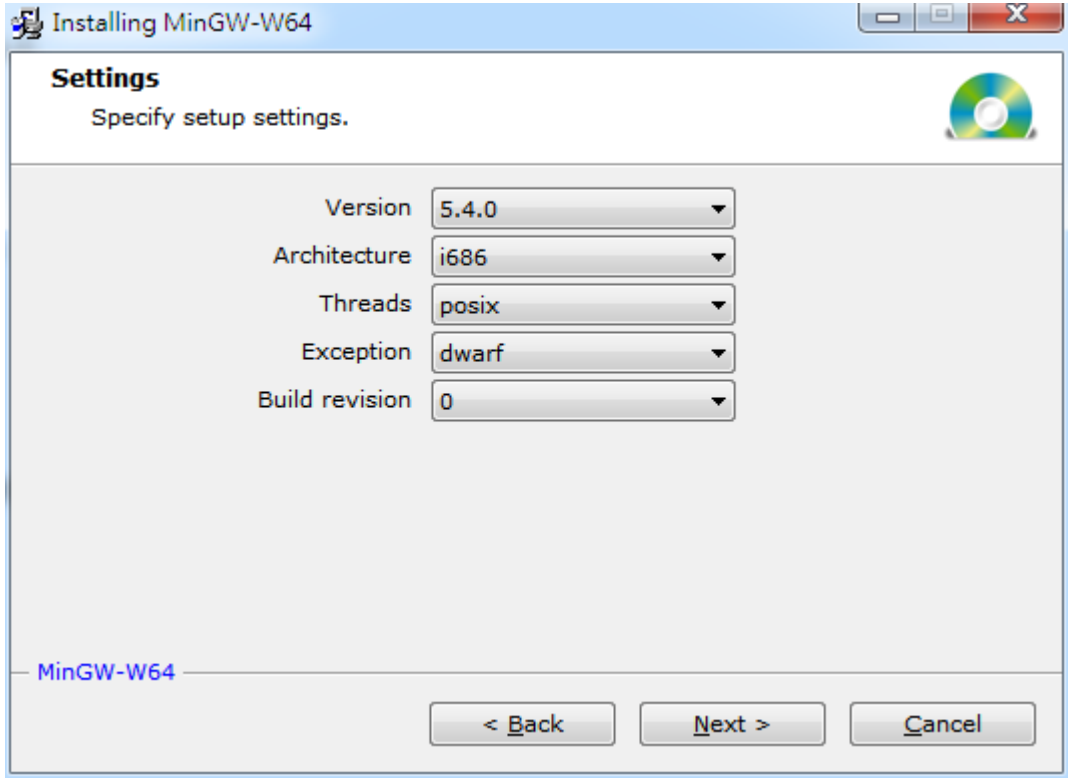
- You must make sure that your homework can be compiled successfully by our makefile.
- If you only compile your source code on your IDE, we won't recognize.

In Windows (Step 1)

- Download the compiler installer(both x86 and x86_64)
  - http://m101.nthu.edu.tw/~s101062122/DS/mingw-w64-install.exe
  - https://sourceforge.net/projects/mingw-w64/

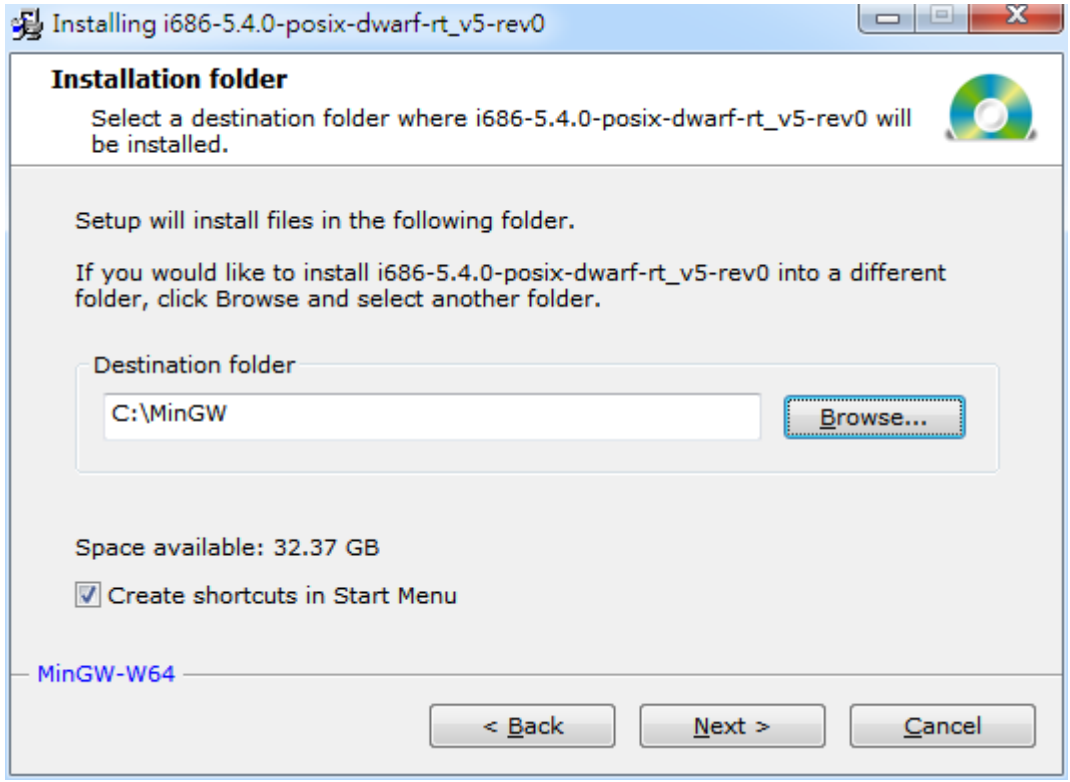
In Windows (Step 2)

- Select gcc version 5.4.0 and i686 architecture



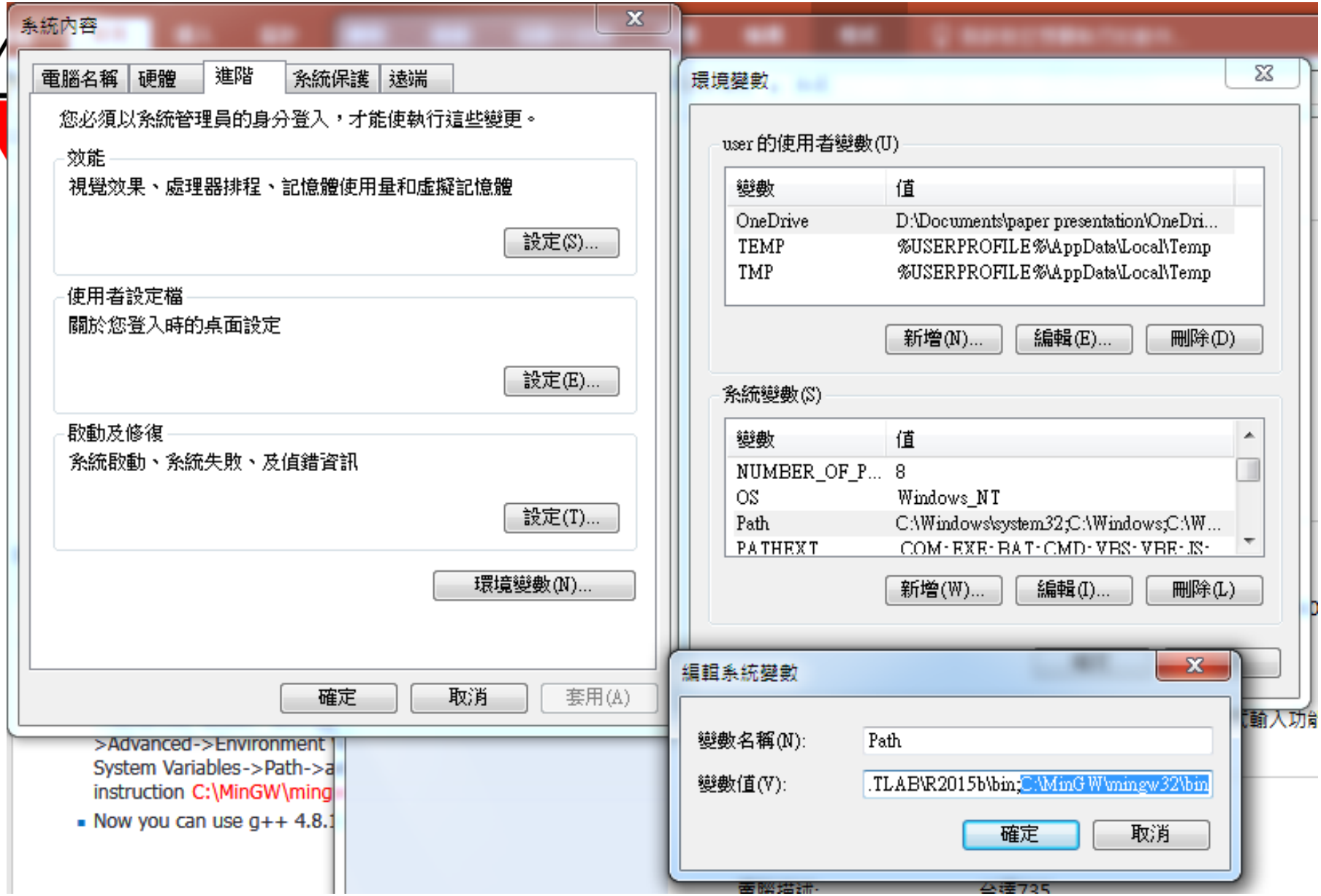
In Windows (Step 3)

- Install the compiler to C:\MinGW



In Windows (Step 4)

- Computer [right button]->Properties->Advanced system settings->Advanced->Environment Variables-> System Variables->Path->add the instruction C:\MinGW\mingw32\bin;
- Now you can use g++ in cmd


系統內容
電腦名稱
硬體
進階
系統保護
遠端
您必須以系統管理員的身分登入，才能使執行這些變更。
效能
視覺效果、處理器排程、記憶體使用量和虛擬記憶體
設定(S)...
使用者設定檔
關於您登入時的桌面設定
設定(E)...
啟動及修復
系統啟動、系統失敗、及偵錯資訊
設定(T)...
環境變數(N)...
確定
取消
套用(A)
環境變數
user 的使用者變數(U)
變數
值
OneDrive
D:\Documents\paper presentation\OneDri...
TEMP
%USERPROFILE%\AppData\Local\Temp
TMP
%USERPROFILE%\AppData\Local\Temp
新增(N)...
編輯(E)...
刪除(D)
系統變數(S)
變數
值
NUMBER_OF_P...
8
OS
Windows_NT
Path
C:\Windows\system32;C:\Windows;C:\W...
新增(W)...
編輯(I)...
刪除(L)
編輯系統變數
變數名稱(N):
Path
變數值(V):
.TLAB\R2015b\bin;C:\MinGW\mingw32\bin
確定
取消
>Advanced->Environment
System Variables->Path->a
instruction C:\MinGW\ming
Now you can use g++ 4.8.1

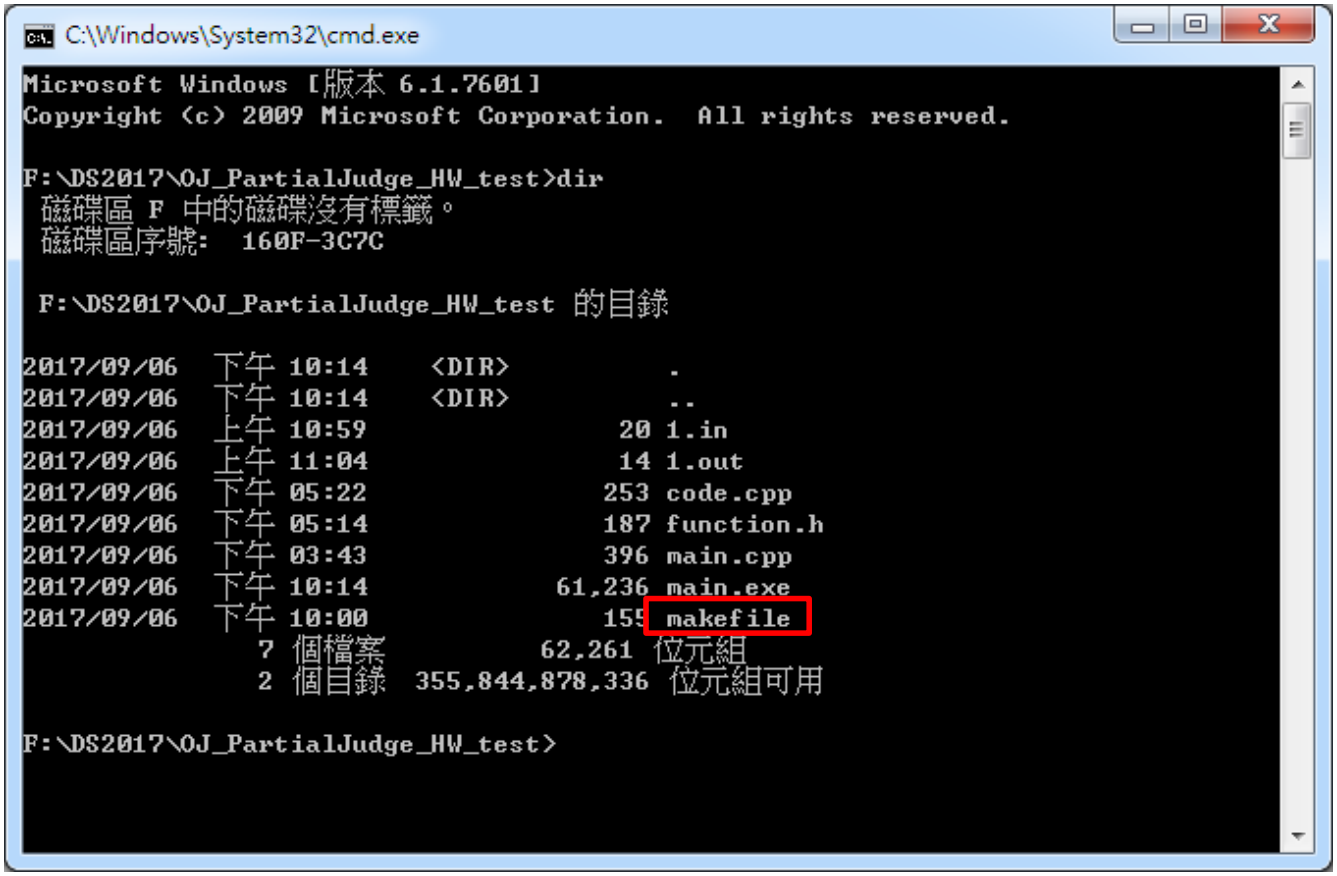
For example

- V sful, t

```
C:\Users\user>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=C:/MinGW/mingw32/bin/../libexec/gcc/i686-w64-mingw32/5.4.0/l
to-wrapper.exe
Target: i686-w64-mingw32
Configured with: ../../../src/gcc-5.4.0/configure --host=i686-w64-mingw32 --buil
d=i686-w64-mingw32 --target=i686-w64-mingw32 --prefix=/mingw32 --with-sysroot=/c
/mingw540/i686-540-posix-dwarf-rt_v5-rev0/mingw32 --with-gxx-include-dir=/mingw3
2/i686-w64-mingw32/include/c++ --enable-shared --enable-static --disable-multili
b --enable-languages=c,c++,fortran,lto --enable-libstdcxx-time=yes --enable-thre
ads=posix --enable-libgomp --enable-libatomic --enable-lto --enable-graphite --e
nable-checking=release --enable-fully-dynamic-string --enable-version-specific-r
untime-libs --disable-sjlj-exceptions --with-dwarf2 --disable-isl-version-check
--disable-libstdcxx-pch --disable-libstdcxx-debug --enable-bootstrap --disable-r
path --disable-win32-registry --disable-nls --disable-werror --disable-symvers -
-with-gnu-as --with-gnu-ld --with-arch=i686 --with-tune=generic --with-libiconv
--with-system-zlib --with-gmp=/c/mingw540/prerequisites/i686-w64-mingw32-static
--with-mpfr=/c/mingw540/prerequisites/i686-w64-mingw32-static --with-mpc=/c/ming
w540/prerequisites/i686-w64-mingw32-static --with-isl=/c/mingw540/prerequisites/
i686-w64-mingw32-static --with-pkgversion='i686-posix-dwarf-rev0, Built by MinGW
-W64 project' --with-bugurl=http://sourceforge.net/projects/mingw-w64 CFLAGS='-O
2 -pipe -I/c/mingw540/i686-540-posix-dwarf-rt_v5-rev0/mingw32/opt/include -I/c/m
ingw540/prerequisites/i686-zlib-static/include -I/c/mingw540/prerequisites/i686-
w64-mingw32-static/include' CXXFLAGS='-O2 -pipe -I/c/mingw540/i686-540-posix-dwa
rf-rt_v5-rev0/mingw32/opt/include -I/c/mingw540/prerequisites/i686-zlib-static/i
nclude -I/c/mingw540/prerequisites/i686-w64-mingw32-static/include' CPPFLAGS= LD
FLAGS='-pipe -L/c/mingw540/i686-540-posix-dwarf-rt_v5-rev0/mingw32/opt/lib -L/c/
mingw540/prerequisites/i686-zlib-static/lib -L/c/mingw540/prerequisites/i686-w64
-mingw32-static/lib -Wl,--large-address-aware'
Thread model: posix
gcc version 5.4.0 (i686-posix-dwarf-rev0, Built by MinGW-W64 project)
```

In Windows (makefile)

- Makefile will help you compile the program

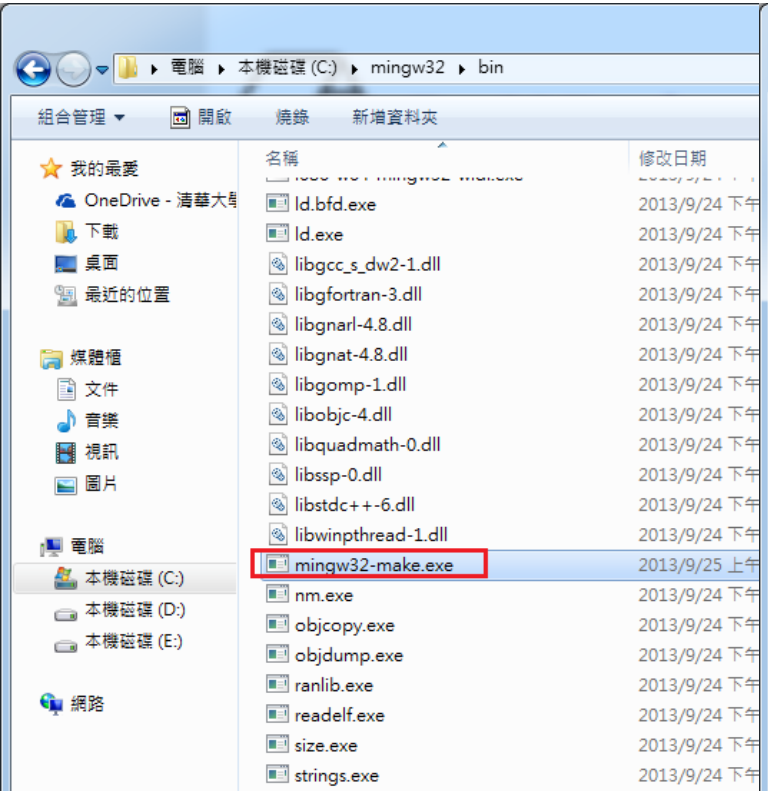


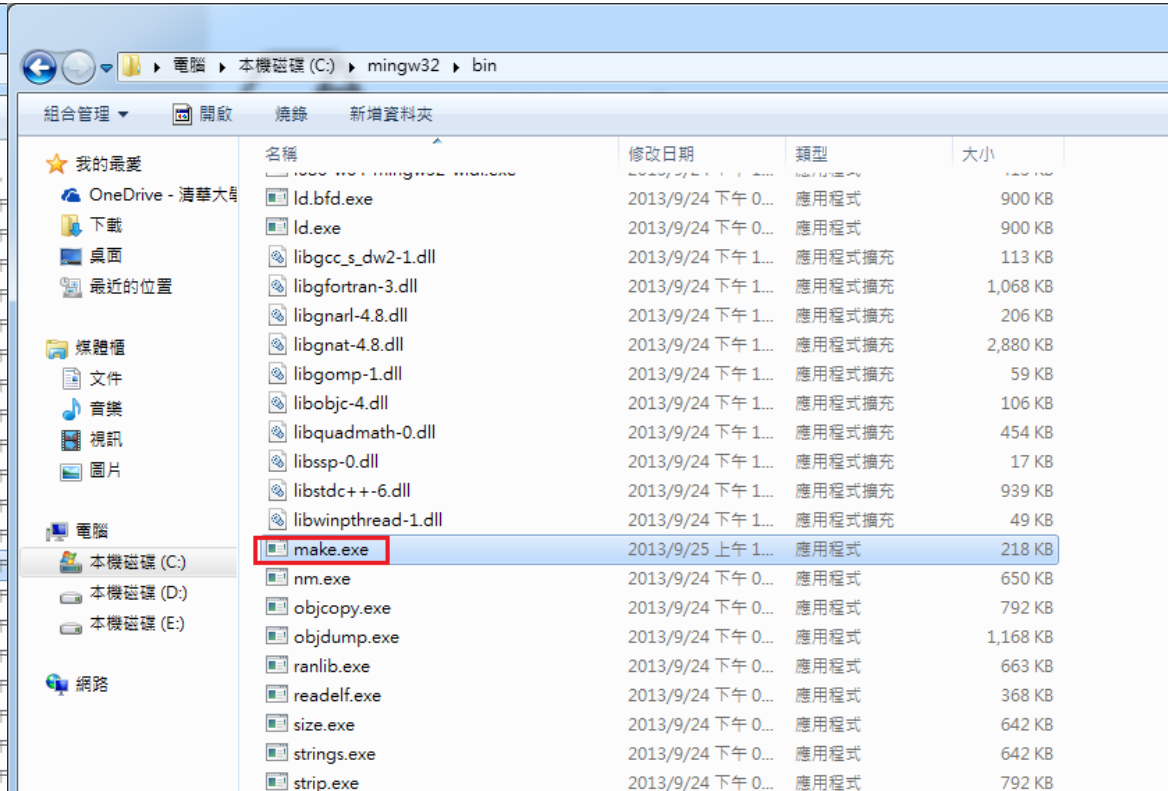
In Windows

- Enter "mingw32-make (-f makefile)" and it will create a executable file of main.cpp.

In Windows

- You can change the filename "mingw32-make.exe" into "make.exe" and use shorter "make"





In Windows

- Finally, run the executable file (main) and use input redirection operator to read the .in file, then main will show the result

```
F:\DS2017\OJ_PartialJudge_HW_test>main.exe < 1.in
3
7
11
15
```