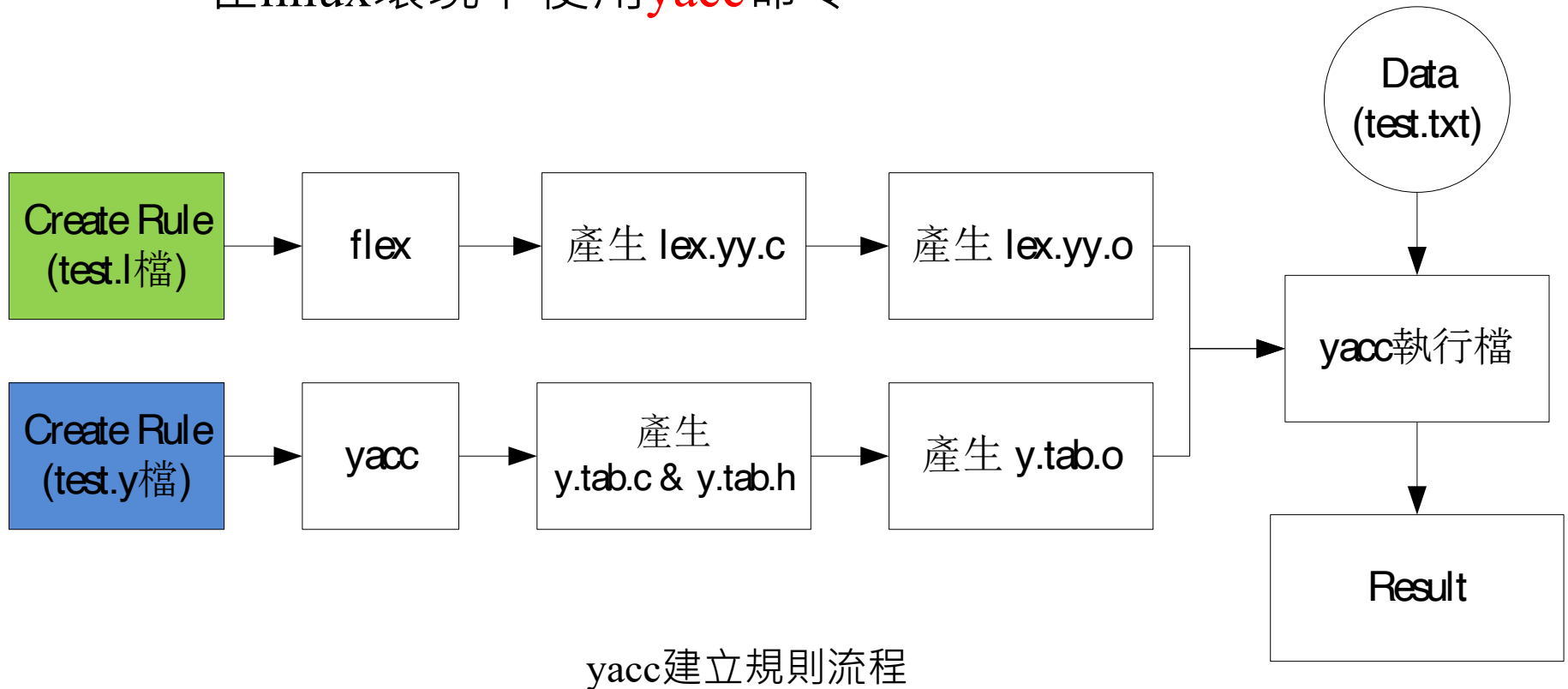


Yacc

-Yet Another Compiler Compiler

What is yacc

- yacc是linux系統中用來生成編譯器的編譯器
 - 在linux環境下使用yacc命令



How to Create rules-*yacc*

- test.y檔：
 - 與lex類似

```
definition //宣告  
%%  
rules      //規則建立  
%%  
user code //主程式
```

definition

- 內容必須用%{ %}包住
 - 主要是用來宣告token，也可以宣告變數、include
 - EX：

```
%token LPAREN RPAREN PLUS REALNUMBER
%{
    #include<stdio.h>
    double global_value;
%}
```

Rules

- 訂定規則：
 - EX：

\$\$ **\$1** **\$2** **\$3**

EXP : REALNUMBER PLUS REALNUMBER

{ printf("%d add %d is %d",\$1,\$3, \$1+ \$3); }

EXP : REALNUMBER MINUS REALNUMBER

{ printf("%d cut %d is %d",\$1,\$3,\$1-\$3); }

User Code

- 利用main()來執行
- EX :

```
int main()
{
    yyparse();
    return 0;
}
int yyerror(char *msg)
{
    printf("Error:%s \n", msg);
}
```

How to Create rules-lex

- test.l檔：
 - 分為3個部分，並以%%隔開

```
definition //標頭檔 宣告變數  
%%  
rules //規則建立  
%%  
user code //主程式(可省略)
```

definition

- 內容必須用%{ %} 包住
 - 主要內容為#include、宣告變數
 - EX：

```
%{  
    #include<stdio.h>  
    #include<stdlib.h>  
    #include "y.tab.h"  
%}
```


Rules

- 訂定規則：
 - EX：

```
[0-9]+ {      return(REALNUMBER);  }  
“(“   {      return(LPAREN);    }  
“)“    {      return(RPAREN);    }  
“+”    {      return(PLUS);      }
```

- [0-9]：代表數字

範例-加減法

```
// test.l
%{
    #include "y.tab.h"
%}
%option noyywrap
%%
[0-9]+          {      yyival = atoi(yytext);
                    return(Realnumber);      }
"+"            {      return(Plus);      }
"-"            {      return(Minus);      }
\n             {      return '\n' ;}
.              {      return yytext[0];}
%%
```

傳回一個整數

```

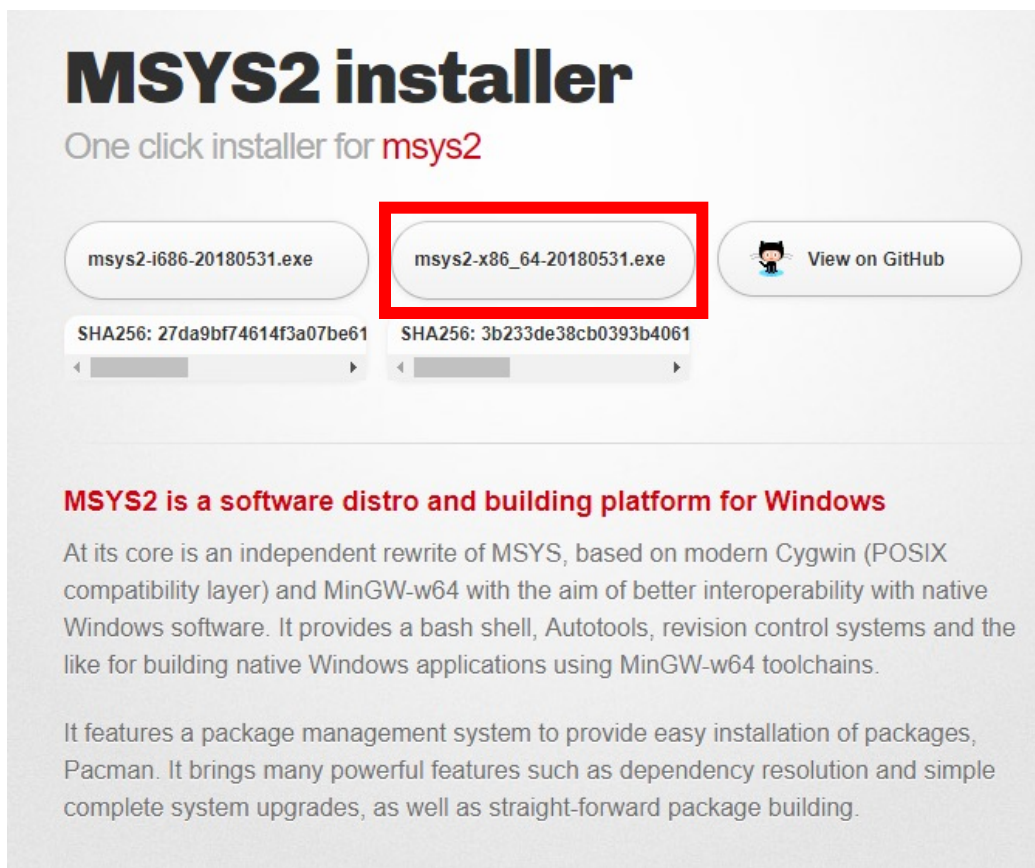
// test.y
%token Realnumber Plus Minus
%{
    #include<stdio.h>
    void yyerror(char *msg);
    extern int yylex();
}%
%%
statement :
    |      statement expression
    |      statement expression '\n'
    ;
expression : Realnumber Plus Realnumber
            { printf("%d add %d is %d\n", $1, $3, $1+$3); }
expression : Realnumber Minus Realnumber
            { printf("%d cut %d is %d\n", $1, $3, $1-$3); }
%%
int main()
{
    yyparse();
    return 0;
}
void yyerror(char *msg)
{
    printf("Error:%s \n", msg);
}

```

可以輸入多行測試

實作方式

- MSYS2
 - <https://www.msys2.org/>



MSYS2 installer

One click installer for **msys2**

[msys2-i686-20180531.exe](#) [msys2-x86_64-20180531.exe](#) [View on GitHub](#)

SHA256: 27da9bf74614f3a07be61 SHA256: 3b233de38cb0393b4061

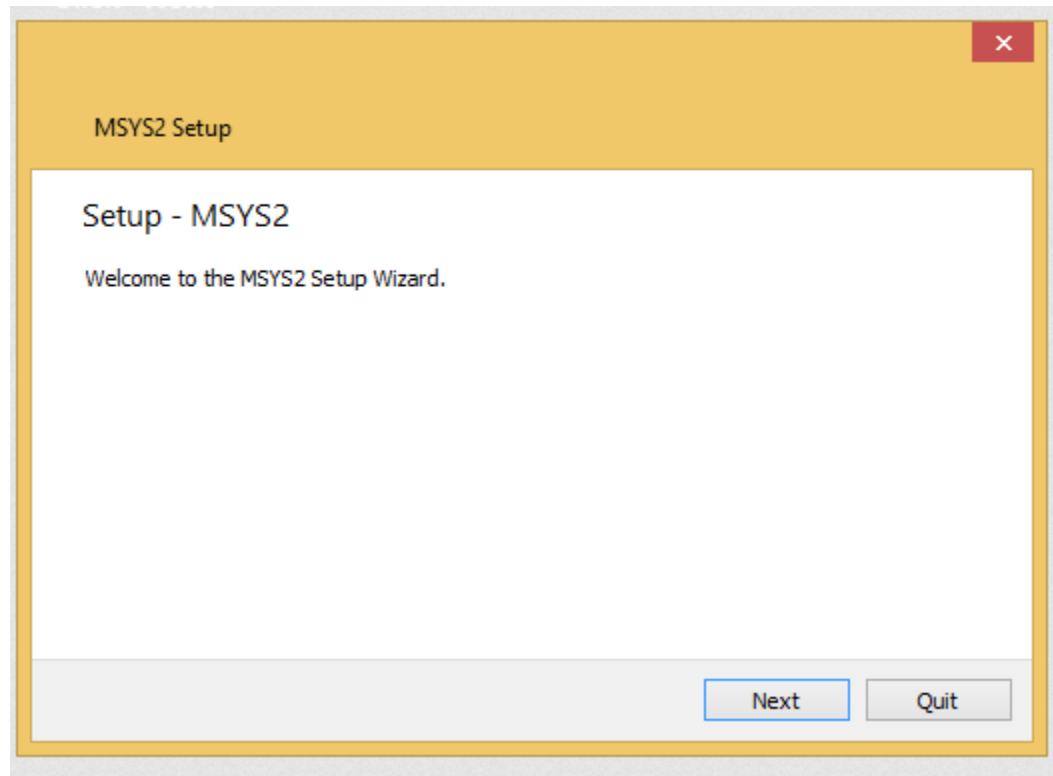
MSYS2 is a software distro and building platform for Windows

At its core is an independent rewrite of MSYS, based on modern Cygwin (POSIX compatibility layer) and MinGW-w64 with the aim of better interoperability with native Windows software. It provides a bash shell, Autotools, revision control systems and the like for building native Windows applications using MinGW-w64 toolchains.

It features a package management system to provide easy installation of packages, Pacman. It brings many powerful features such as dependency resolution and simple complete system upgrades, as well as straight-forward package building.

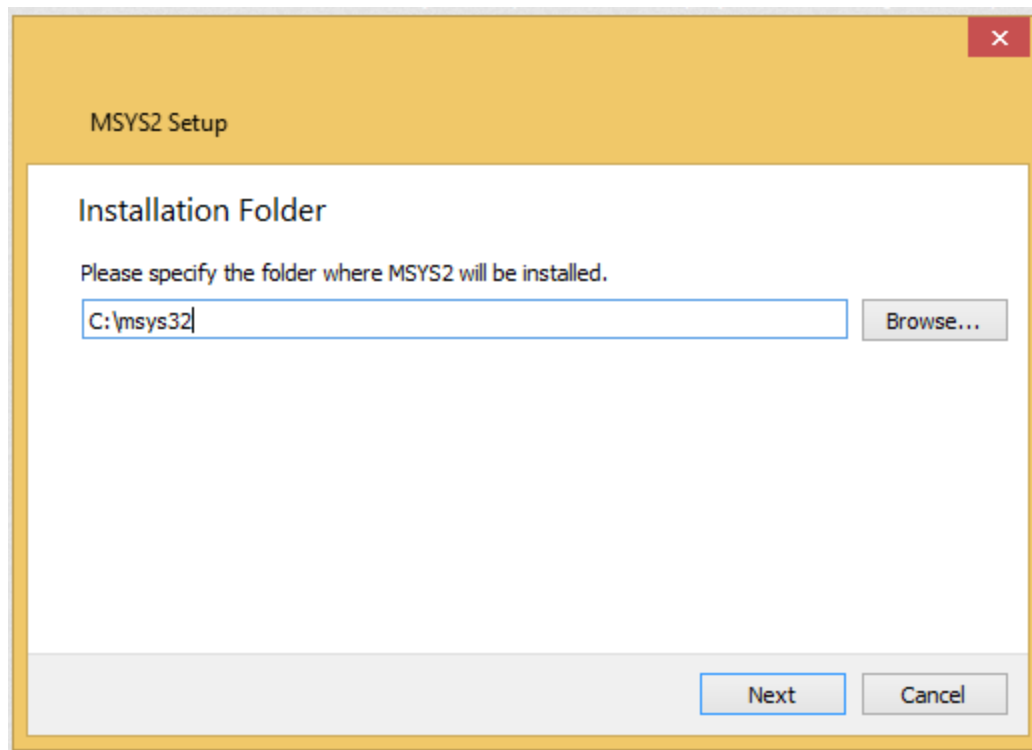
實作方式

- 選擇下一步



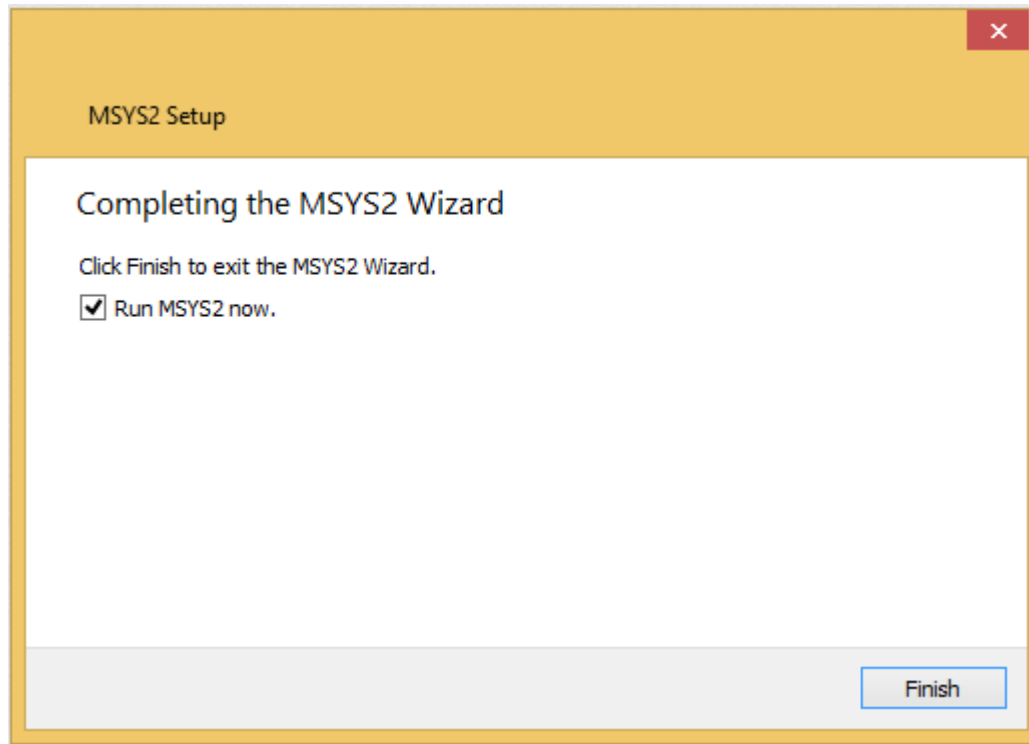
實作方式

- 選擇安裝的路徑



實作方式

- 安裝完成後，便可以開啟MSYS。



實作方式

- 接著需要進行更新，請輸入 `$ pacman -Syu`

```
user@DESKTOP-430QKCT MSYS ~
$ pacman -Syu
:: 正在同步軟體包資料庫...
  mingw32                492.3 KiB   290K/s  00:02 [#####] 100%
  mingw32.sig             119.0 B    0.00B/s  00:00 [#####] 100%
  mingw64                 494.1 KiB   289K/s  00:02 [#####] 100%
  mingw64.sig             119.0 B    0.00B/s  00:00 [#####] 100%
msys 已經是最新版本
:: Starting core system upgrade...
已無事可做
:: 正在進行全系統更新...
正在解決依賴關係...
正在檢查衝突的軟體包...

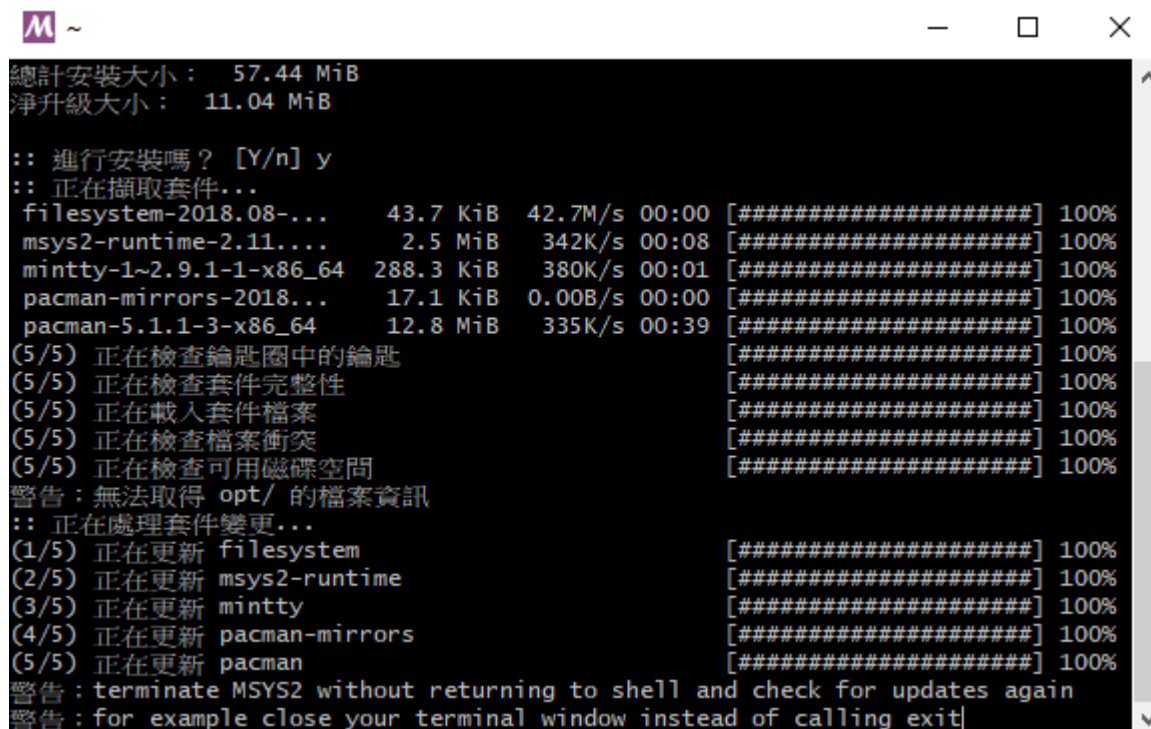
軟體包 (1) mingw-w64-x86_64-gettext-0.19.8.1-6

總計下載大小：   3.15 MiB
總計安裝大小：  18.16 MiB
淨升級大小：    0.00 MiB

:: 進行安裝嗎？ [Y/n] |
```


實作方式

- 看到以下畫面，請重新開啟MSYS



```
M ~
總計安裝大小： 57.44 MiB
淨升級大小： 11.04 MiB

:: 進行安裝嗎？ [Y/n] y
:: 正在擷取套件...
filesystem-2018.08-... 43.7 KiB 42.7M/s 00:00 [#####] 100%
msys2-runtime-2.11.... 2.5 MiB 342K/s 00:08 [#####] 100%
mintty-1~2.9.1-1-x86_64 288.3 KiB 380K/s 00:01 [#####] 100%
pacman-mirrors-2018... 17.1 KiB 0.00B/s 00:00 [#####] 100%
pacman-5.1.1-3-x86_64 12.8 MiB 335K/s 00:39 [#####] 100%
(5/5) 正在檢查鑰匙圈中的鑰匙 [#####] 100%
(5/5) 正在檢查套件完整性 [#####] 100%
(5/5) 正在載入套件檔案 [#####] 100%
(5/5) 正在檢查檔案衝突 [#####] 100%
(5/5) 正在檢查可用磁碟空間 [#####] 100%
警告：無法取得 opt/ 的檔案資訊
:: 正在處理套件變更...
(1/5) 正在更新 filesystem [#####] 100%
(2/5) 正在更新 msys2-runtime [#####] 100%
(3/5) 正在更新 mintty [#####] 100%
(4/5) 正在更新 pacman-mirrors [#####] 100%
(5/5) 正在更新 pacman [#####] 100%
警告：terminate MSYS2 without returning to shell and check for updates again
警告：for example close your terminal window instead of calling exit
```

實作方式

- 當開啟完成時，請輸入 `$ pacman -Su`

```
hugh@DESKTOP-QE760B3 MSYS ~  
$ pacman -Su  
:: Starting core system upgrade...  
已無事可做  
:: 正在進行全系統更新--  
正在解決依賴關係--  
正在檢查衝突的軟體包--  
  
軟體包 (62) bash-completion-2.8-2 brotli-1.0.6-1 bsdcpio-3.3.3-3  
bsdtar-3.3.3-3 coreutils-8.30-1 curl-7.61.0-2 dash-0.5.10.2-1  
dtc-1.4.7-1 file-5.34-1 gawk-4.2.1-1 gcc-libs-7.3.0-3  
glib2-2.54.3-1 gnupg-2.2.9-1 grep-3.1-1 heimdal-libs-7.5.0-2  
icu-62.1-1 info-6.5-2 less-530-1 libarchive-3.3.3-3  
libargp-20110921-2 libassuan-2.5.1-1 libcrypto-2.1-2  
libcurl-7.61.0-2 libexpat-2.2.6-1 libffi-3.2.1-3  
libgcrypt-1.8.3-2 libgnutls-3.6.3-1 libgpg-error-1.32-1  
libgpgme-1.11.1-1 libhogweed-3.4-1 libidn2-2.0.5-1  
libksba-1.3.5-1 liblz4-1.8.2-1 liblzma-5.2.4-1 liblzo2-2.10-2  
libnettle-3.4-1 libnptl-1.6-1 libopenssl-1.0.2.p-2  
libpcre-8.42-1 libpcre16-8.42-1 libpcre32-8.42-1  
libpcrecpp-8.42-1 libpcreposix-8.42-1 libpsl-0.20.2-1  
libreadline-7.0.005-1 libsqlite-3.21.0-4 libunistring-0.9.10-1  
libutil-linux-2.32.1-1 libxml2-2.9.8-1 m4-1.4.18-2  
ncurses-6.1.20180908-1 nettle-3.4-1 openssl-1.0.2.p-2  
pcre-8.42-1 pinentry-1.1.0-2 rebase-4.4.4-1 sed-4.5-1  
time-1.9-1 ttyrec-1.0.8-2 util-linux-2.32.1-1 wget-1.19.5-1  
xz-5.2.4-1  
  
總計下載大小： 34.08 MiB  
總計安裝大小： 153.95 MiB  
淨升級大小： 12.42 MiB  
  
:: 進行安裝嗎？ [Y/n]
```

- 這裡請輸入y

實作方式

- 安裝GCC 輸入
- `$ pacman -S git mingw-w64-x86_64-gcc base-devel`

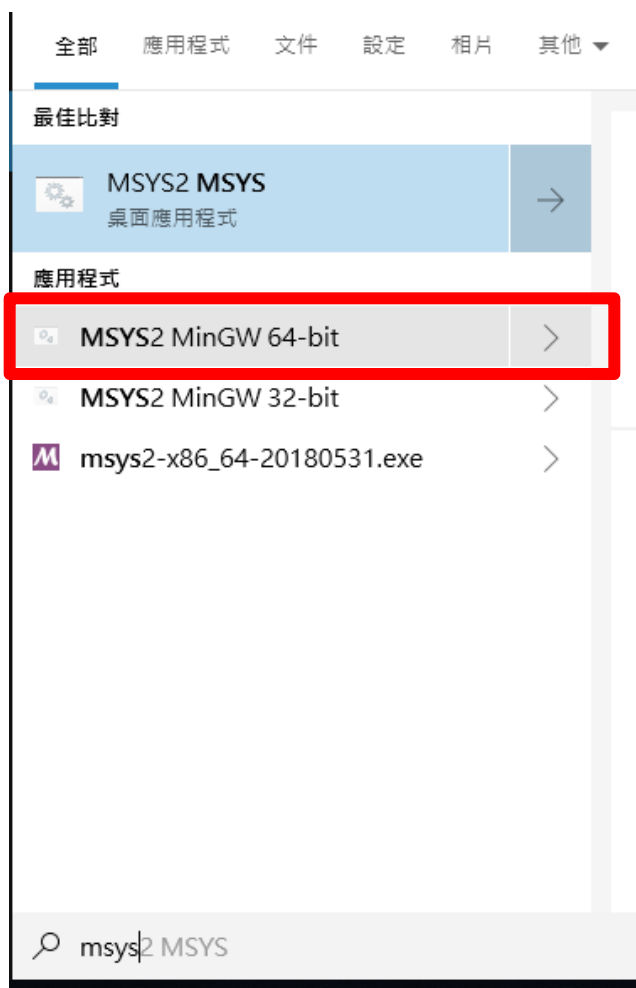
```
$ pacman -S git mingw-w64-x86_64-gcc base-devel
:: 在群組 base-devel 中有 56 個成員：
:: 軟體庫 msys
 1) asciidoc  2) autoconf  3) autoconf2.13  4) autogen  5) automake-wrapper  6
15) automake1.8 16) automake1.9 17) bison 18) diffstat 19) diffutils 20)
33) libtool 34) libunrar 35) libunrar-devel 36) m4 37) make 38) man-db
52) texinfo-tex 53) ttyrec 54) unrar 55) wget 56) xmlto

輸入某個選擇（預設=全部選定）：
```

- 這裡請按Enter

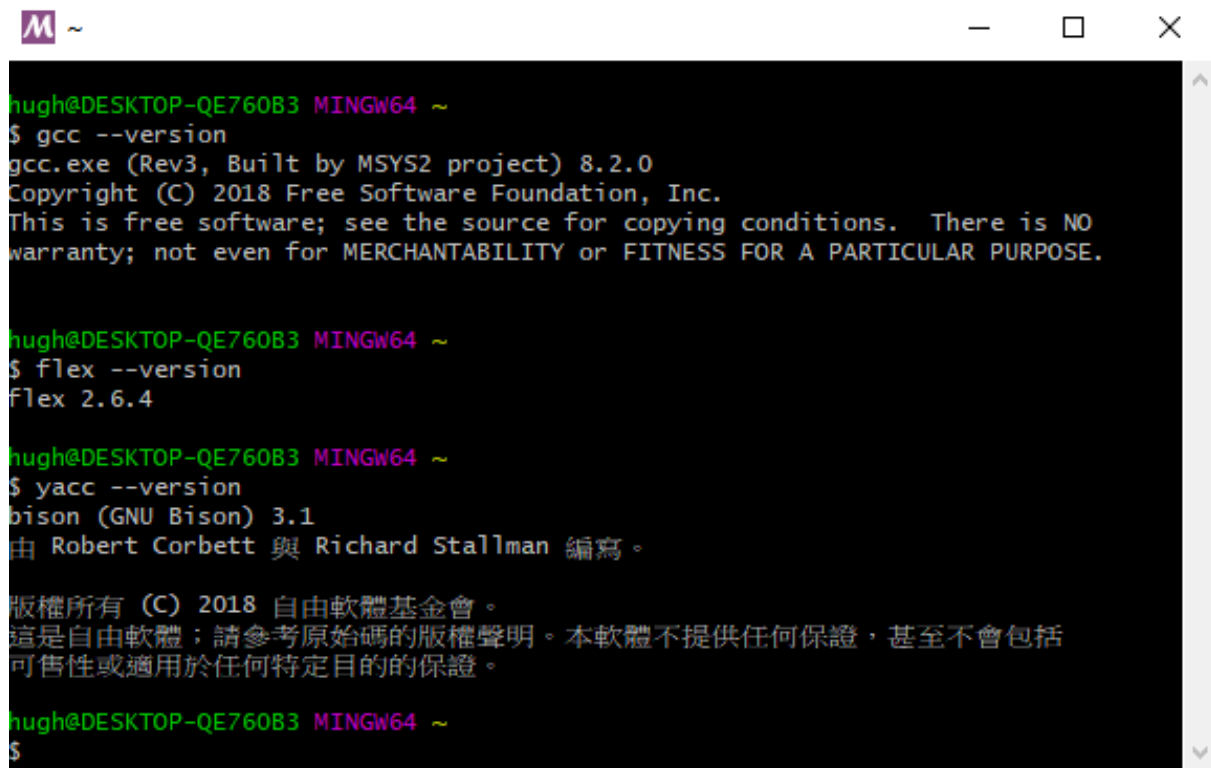
實作方式

關掉原本的MSYS，使用搜尋，開啟MSYS2 MinGW 64-bit



實作方式

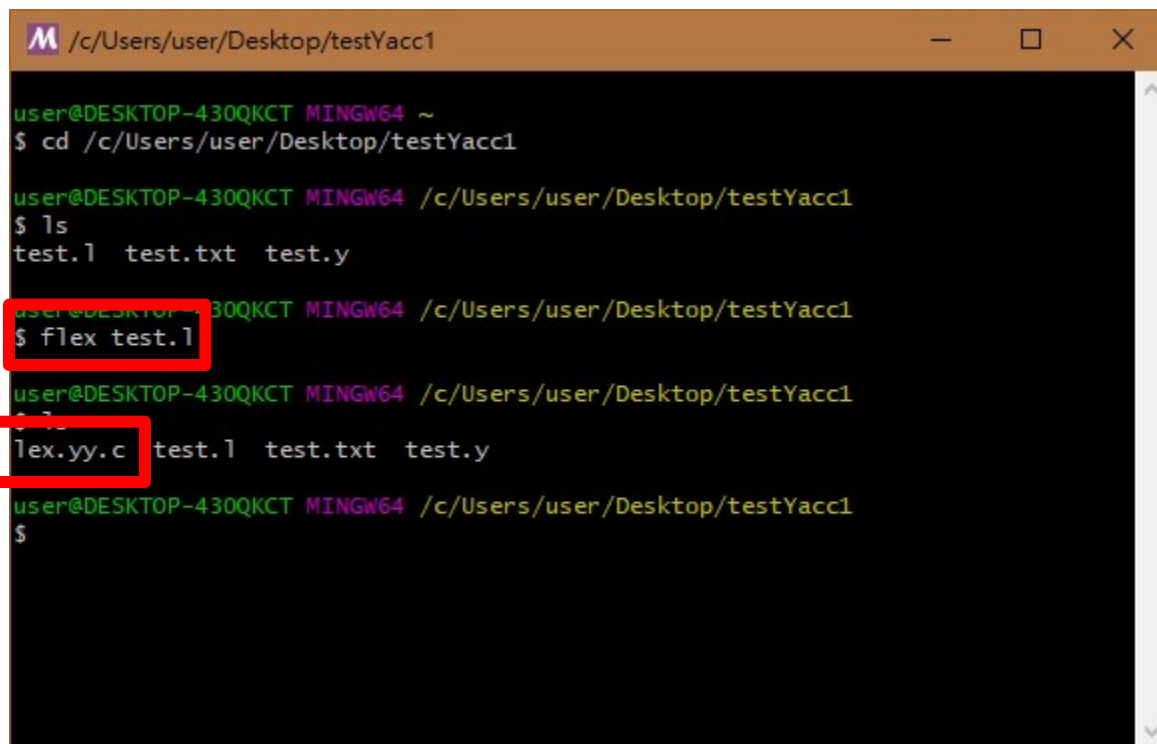
- 可以透過輸入以下指令檢查是否安裝成功
- `$ gcc --version`
- `$ flex --version`
- `$ yacc --version`



```
hugh@DESKTOP-QE760B3 MINGW64 ~  
$ gcc --version  
gcc.exe (Rev3, Built by MSYS2 project) 8.2.0  
Copyright (C) 2018 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
  
hugh@DESKTOP-QE760B3 MINGW64 ~  
$ flex --version  
flex 2.6.4  
  
hugh@DESKTOP-QE760B3 MINGW64 ~  
$ yacc --version  
bison (GNU Bison) 3.1  
由 Robert Corbett 與 Richard Stallman 編寫。  
  
版權所有 (C) 2018 自由軟體基金會。  
這是自由軟體；請參考原始碼的版權聲明。本軟體不提供任何保證，甚至不會包括  
可售性或適用於任何特定目的的保證。  
  
hugh@DESKTOP-QE760B3 MINGW64 ~  
$
```

實作方式

- 在我們撰寫好test.1檔的規則後，我們可以利用flex命令使得test.1檔轉換為lex.yy.c檔。
 - flex test.1



```

M /c/Users/user/Desktop/testYacc1
user@DESKTOP-430QKCT MINGW64 ~
$ cd /c/Users/user/Desktop/testYacc1

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$ ls
test.1 test.txt test.y

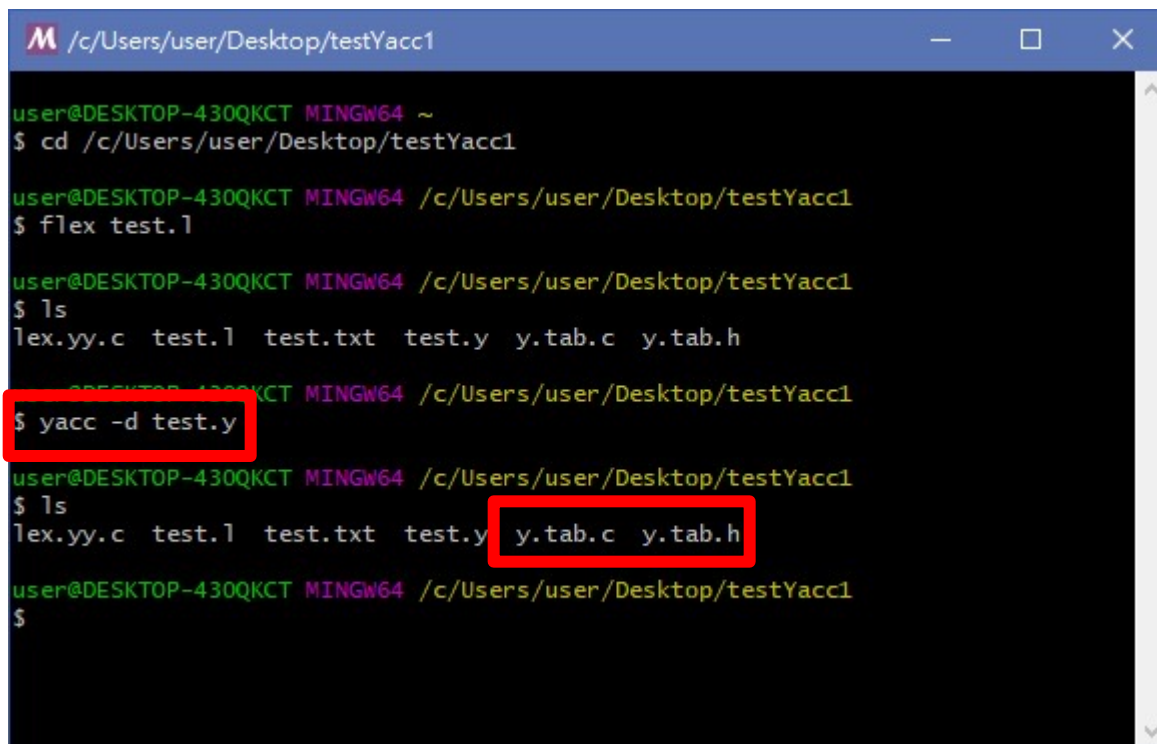
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$ flex test.1

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$ ls
lex.yy.c test.1 test.txt test.y

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$
```

實作方式

- 我們新增yacc檔(test.y) , 並且利用yacc指令產生y.tab.c和y.tab.h 。
 - yacc -d test.y



```
M /c/Users/user/Desktop/testYacc1

user@DESKTOP-430QKCT MINGW64 ~
$ cd /c/Users/user/Desktop/testYacc1

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$ flex test.1

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$ ls
lex.yy.c test.1 test.txt test.y y.tab.c y.tab.h

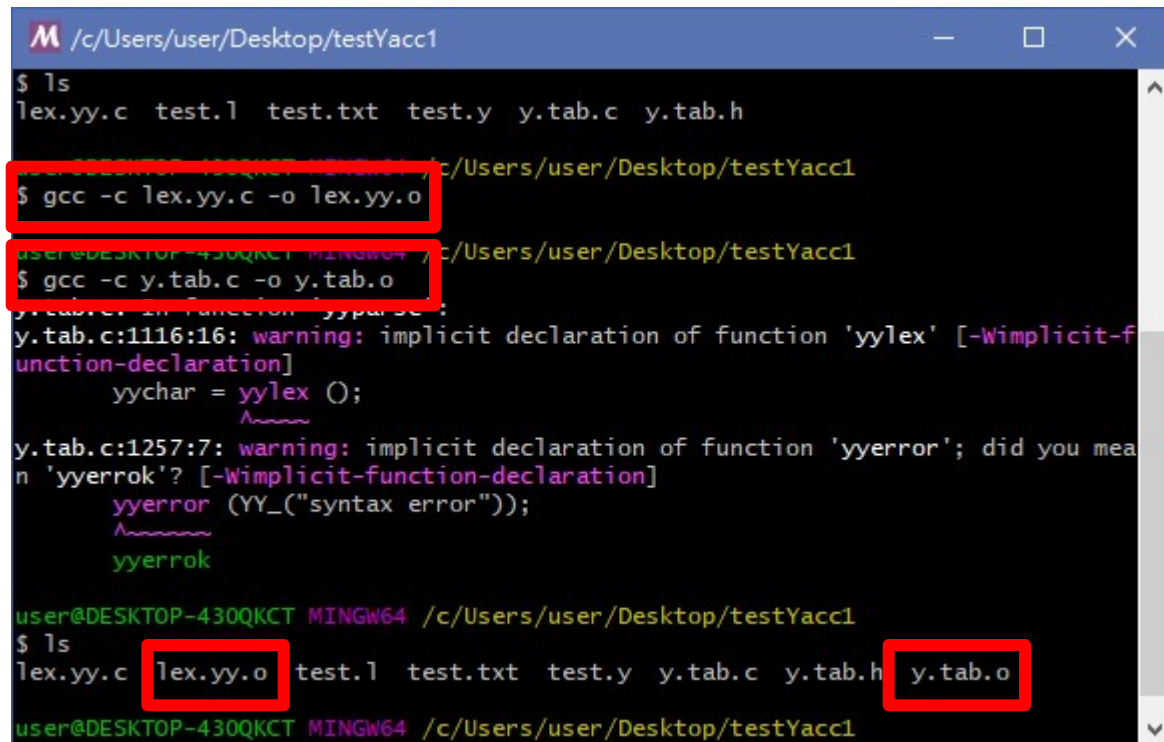
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$ yacc -d test.y

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$ ls
lex.yy.c test.1 test.txt test.y y.tab.c y.tab.h

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$
```

實作方式

- 有了y.tab.c和lex.yy.c，我們可以產生obj檔。
 - gcc -c lex.yy.c -o lex.yy.o
 - gcc -c y.tab.c -o y.tab.o

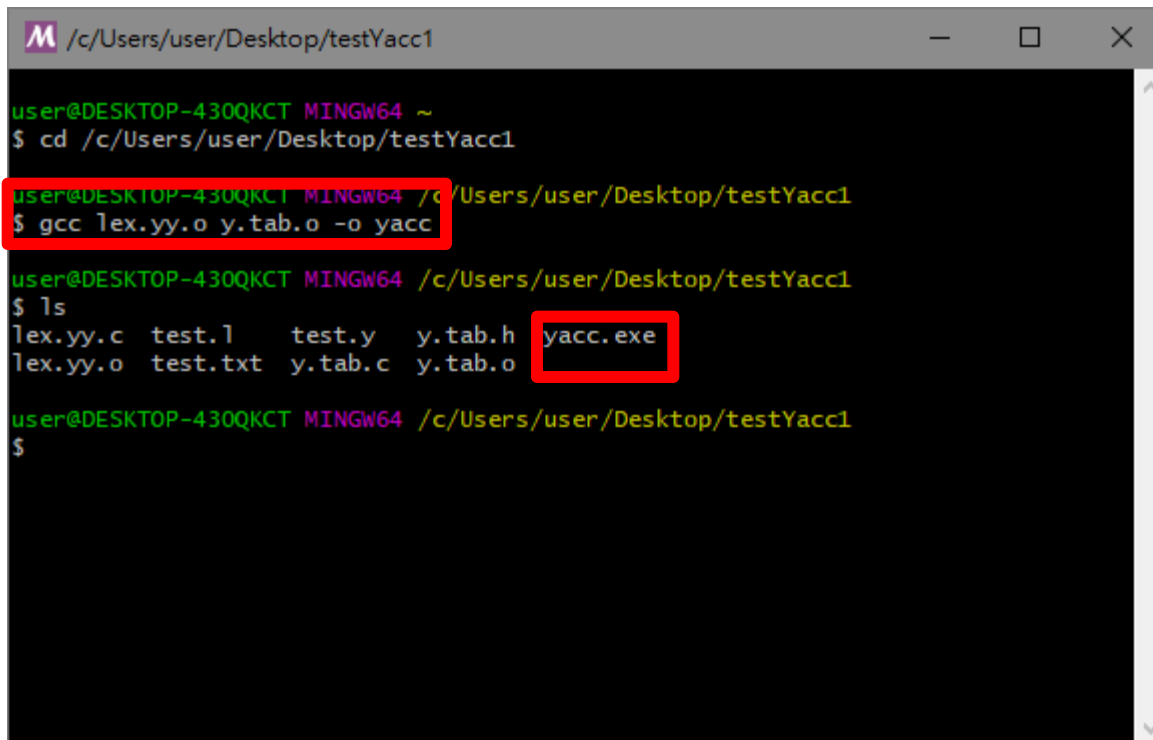


```
M /c/Users/user/Desktop/testYacc1
$ ls
lex.yy.c test.l test.txt test.y y.tab.c y.tab.h
$ gcc -c lex.yy.c -o lex.yy.o
$ gcc -c y.tab.c -o y.tab.o
y.tab.c:1116:16: warning: implicit declaration of function 'yylex' [-Wimplicit-f
function-declaration]
    yychar = yylex ();
               ^~~~~
y.tab.c:1257:7: warning: implicit declaration of function 'yyerror'; did you mea
n 'yyerrok'? [-Wimplicit-function-declaration]
    yyerror (YY_("syntax error"));
    ^~~~~~
    yyerrok

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1
$ ls
lex.yy.c lex.yy.o test.l test.txt test.y y.tab.c y.tab.h y.tab.o
```


實作方式

- 得到兩個obj檔後，我們便可以做出最後的yacc執行檔。
 - `gcc lex.yy.o y.tab.o -o yacc`



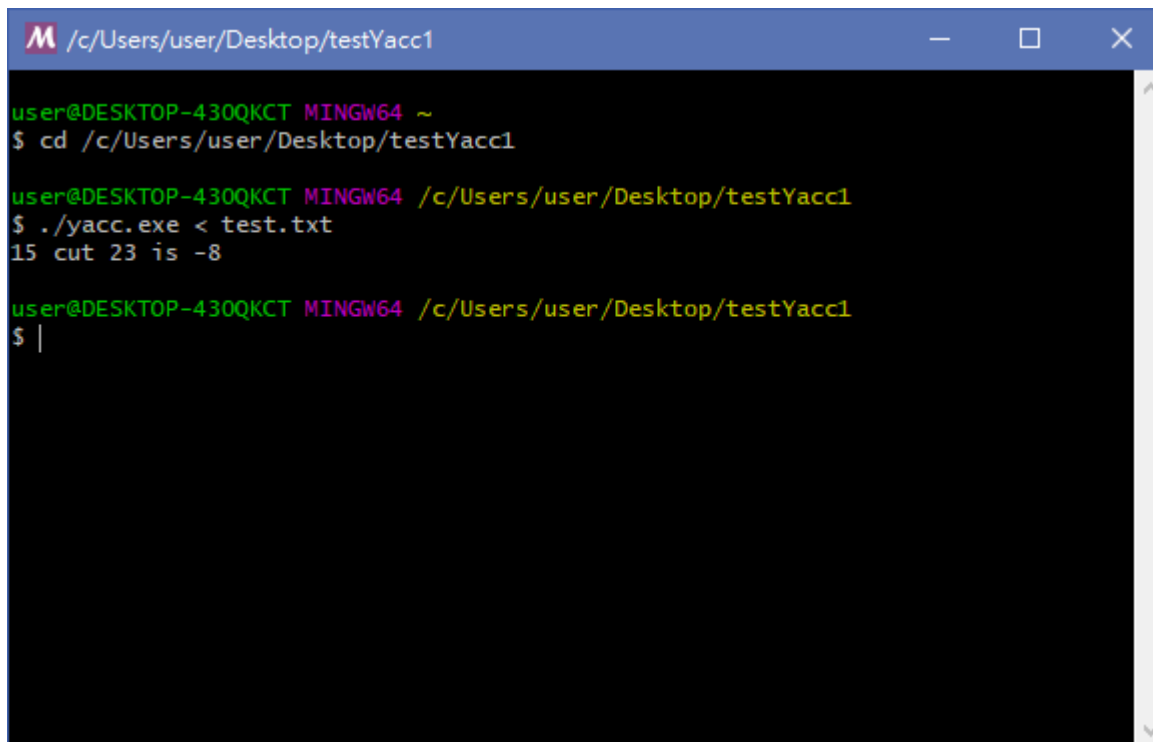
```
user@DESKTOP-430QKCT MINGW64 ~  
$ cd /c/Users/user/Desktop/testYacc1  
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1  
$ gcc lex.yy.o y.tab.o -o yacc  
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1  
$ ls  
lex.yy.c  test.l    test.y    y.tab.h  yacc.exe  
lex.yy.o  test.txt  y.tab.c  y.tab.o  
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1  
$
```

實作方式

- 建立一測試檔test.txt：

```
15-23  
3+3  
5-16
```

- 輸入./yacc.exe < test.txt
分析：



```
M /c/Users/user/Desktop/testYacc1  
user@DESKTOP-430QKCT MINGW64 ~  
$ cd /c/Users/user/Desktop/testYacc1  
  
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1  
$ ./yacc.exe < test.txt  
15 cut 23 is -8  
  
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testYacc1  
$ |
```

作業

- 實作“+”、“-”、“*”、“/”、“(“、“)”、“-”(負號or運算子)等符號，如下：

```
12+32  
56-2*3  
13+6/2  
(5+2)*3  
-5+37
```

- 輸出結果為印出各個計算式的計算過程及答案，必須要遵守先乘除後加減的原則。
- 『-』可能代表負號或運算子

作業顯示結果

測試檔案

$12+32+35+21-20$

$65+35$

$32-2$

$112-(2+32)*3$

$36/(30+3*2)$

$12+(12-(12+12))|$

$89+77/(8-1)$

```
12 plus 32 equals 44
44 plus 35 equals 79
79 plus 21 equals 100
100 minus 20 equals 80
ANS is 80
```

```
65 plus 35 equals 100
ANS is 100
```

```
32 minus 2 equals 30
ANS is 30
```

```
2 plus 32 equals 34
34 multiply 3 equals 102
112 minus 102 equals 10
ANS is 10
```

```
3 multiply 2 equals 6
30 plus 6 equals 36
36 divide 36 equals 1
ANS is 1
```

```
12 plus 12 equals 24
12 minus 24 equals -12
12 plus -12 equals 0
ANS is 0
```

```
8 minus 1 equals 7
77 divide 7 equals 11
89 plus 11 equals 100
ANS is 100
```

作業

- 繳交檔案
 - lex檔案 (.l)
 - YACC檔案 (.y)
 - 報告檔案 (.pdf)
- 繳交方式：上傳至ilearn
- 期限：12/15(日) 晚上 11:55之前
- 報告格式
 - 1.封面：作業題目、班級、學號、姓名
 - 2.作法：規則建立說明
 - 3.程式碼
 - 4.執行結果：測試資料及結果
 - 5.討論：遭遇困難及解決過程
 - 6.心得：自由發揮

延伸作業

- 試著用C/C++或其他語言撰寫parser程式，
 - 輸入及輸出如作業相同。
- 將程式檔與作業一起繳交，並於報告中註記有寫加分題
- 額外加分(3分)。

- 有任何問題請用E-mail或ilearn聯絡或至Lab詢問：
 - Lab：資電230（低功率系統結構實驗室）
 - 莊惠淇：M1310081@o365.fcu.edu.tw
 - 劉秉霖：M1305865@o365.fcu.edu.tw

參考資料

- <http://www.csie.ntu.edu.tw/~compiler/yacc/>
- <http://zh.wikipedia.org/zh-hk/Yacc>
- <http://www.ibm.com/developerworks/cn/linux/sdk/lex/index.html>
- <http://dinosaur.compilertools.net/#yacc>
- <http://0rz.tw/7bDqQ>