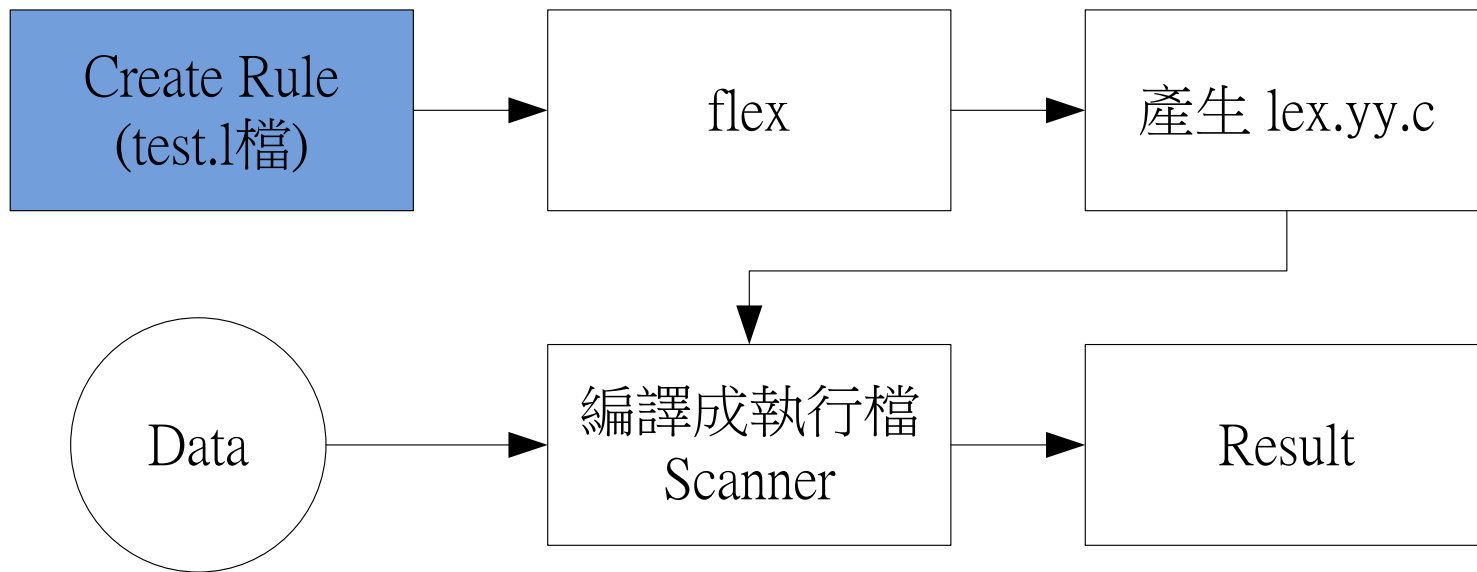


# lex

-A scanner generator

# What is lex

- lex是linux系統中標準的詞法分析器產生程式
  - 我們使用flex（fast lexical analyzer,快速的詞法分析器）



lex建立規則流程

# How to Create rules

- 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
```

# 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>  
    int Total_Num = 0 ;  
% }
```

# Rules

- 訂定規則：
  - EX：

```
[0-9]  {  
        printf(“integer : %d\n”,yytext);  
        TotalNum++;  
    }
```

- [0-9]：代表數字

# Rules

型式	規則描述	範例
ABC...abc...123..	構成了部份的字串以及數字	printf
.	代表任意字串or數字，(\n除外)	.
-	用來指定範圍	A-Z
[]	一個字串的集合，只要符合[]中任一個字便成立	[abc]
*	符合0個或多個	[0-9]*A
+	符合1個或多個	[0-9]+A
?	符合0個或1個	[0-9]?A
{}	用來表示某個字可能出現的次數	A{1,3}
^	否定符號	^[A-Z]
	或	Abc   123

# Rules

- 訂定規則例子：
  - $[0-9]^+[A-Z]^?$  : 09998A、123C、22.....
  - $A\{1,3\}[abc]$  : Aa,Ab,Ac,AAAa,AAAb,AAAc
  - $\text{Printf}[abc]$  : Printf a、Printf b、Printf c
  - $\backslash(+ :$  (、((、(((、((((、.....
  - $[A-Z0-9]^+$  : N95、RS232、Z454E.....

# User Code

- 利用 main() 來執行
- EX :

```
int main()
{
    yylex();
    return 0;
}
```



# Global Variable

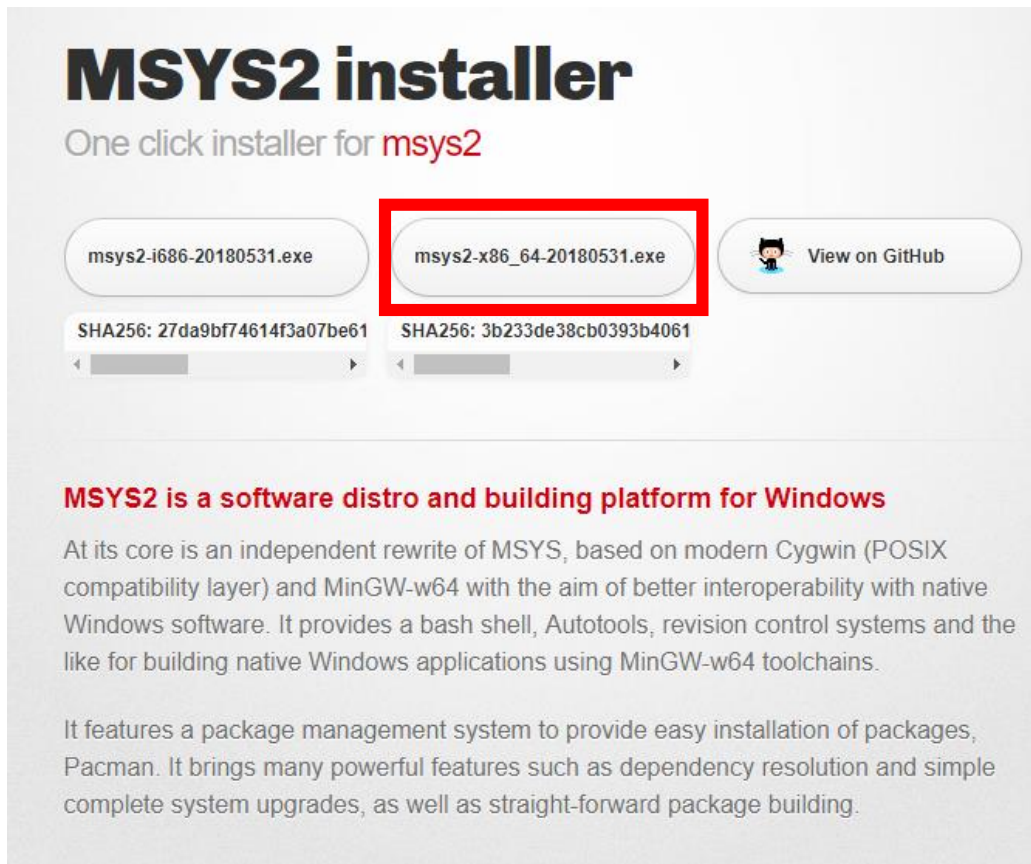
- 下列是lex預設的變數，在寫xxx.l檔時，我們可以直接使用這些變數：
  - yytext：當Rule中有符合的字串時，符合的字串就會存放在yytext中，型態為char\*。
  - yyleng：記錄著yytext的長度。
  - yylineno：記錄目前的yyin讀到第幾行。
  - yyin：是lex的input來源，型態為FILE，預設為stdin。

# 範例-數字統計

```
%{  
    #include<stdio.h>  
    int Total_Num=0;  
%}  
%option noyywrap ← 分析單一文件  
%%  
[0-9]+ {  
    printf("Find a Number: %s\n",yytext);  
    Total_Num++;  
}  
.  
\n {  
%%  
int main()  
{  
    yylex();  
    printf("Total_Num = %d\n",Total_Num);  
    return 0;  
}
```

# 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

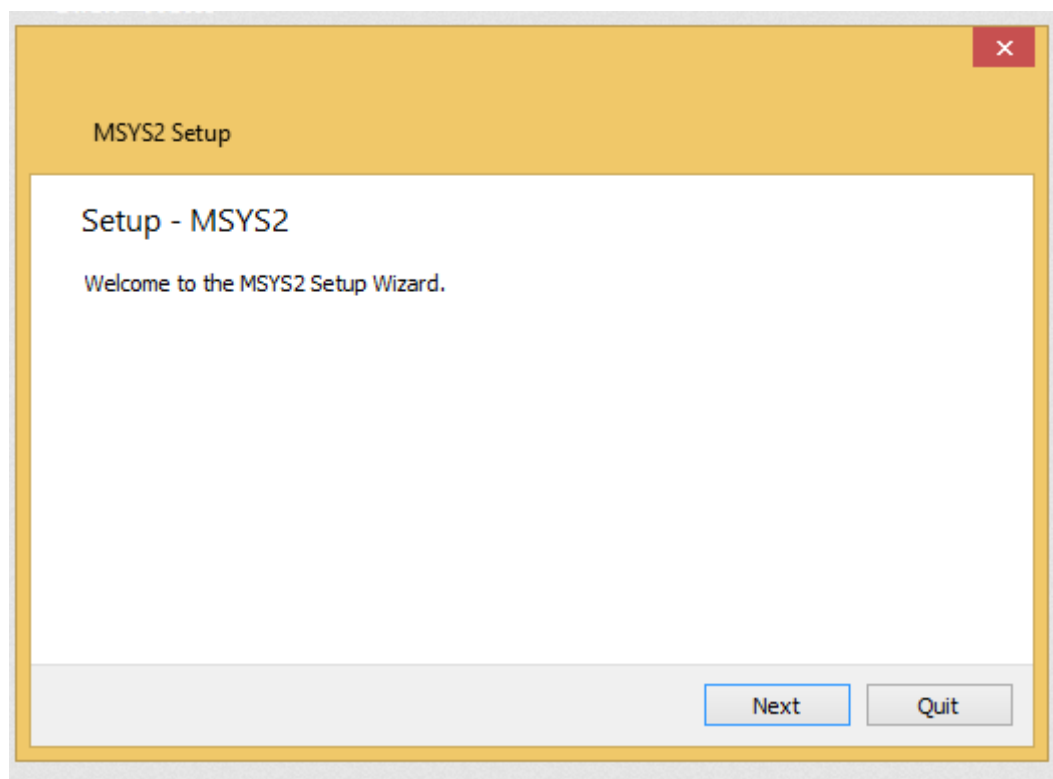
**MSYS2 is a software distro and building platform for Windows**

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.

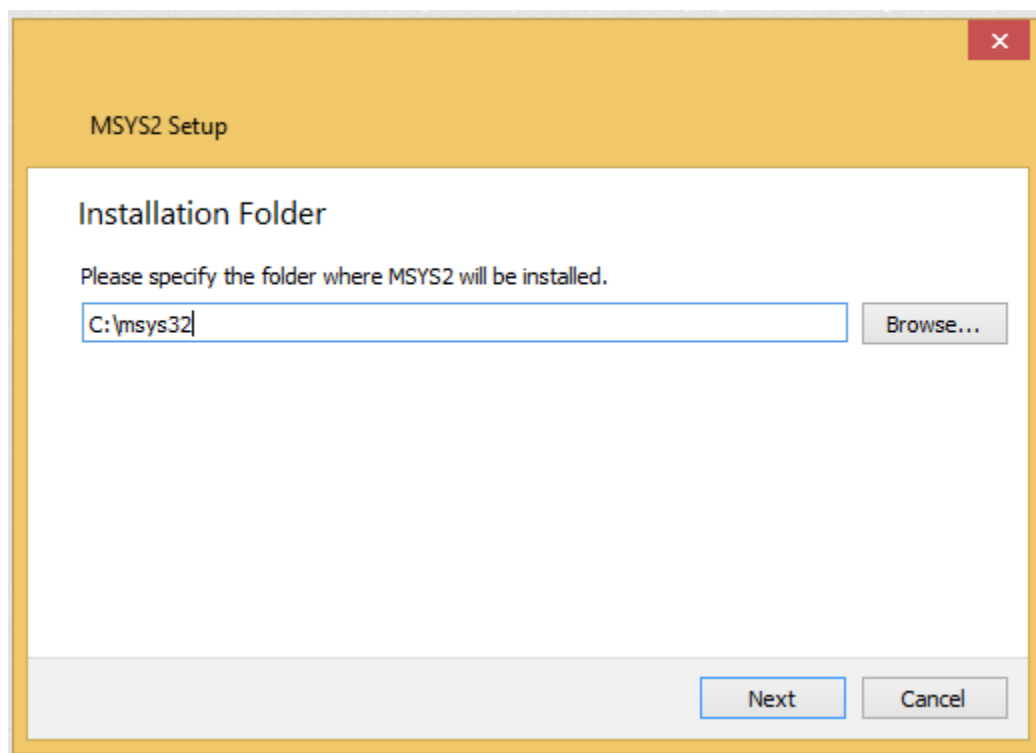
# 實作方式

- 選擇下一步



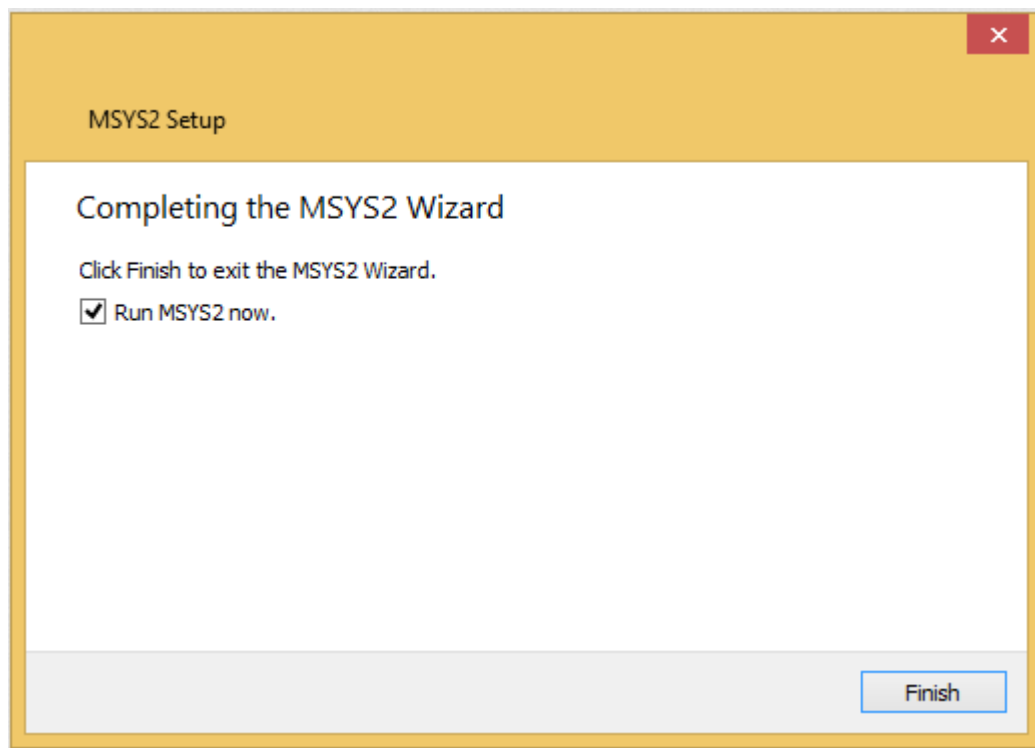
# 實作方式

- 選擇安裝的路徑



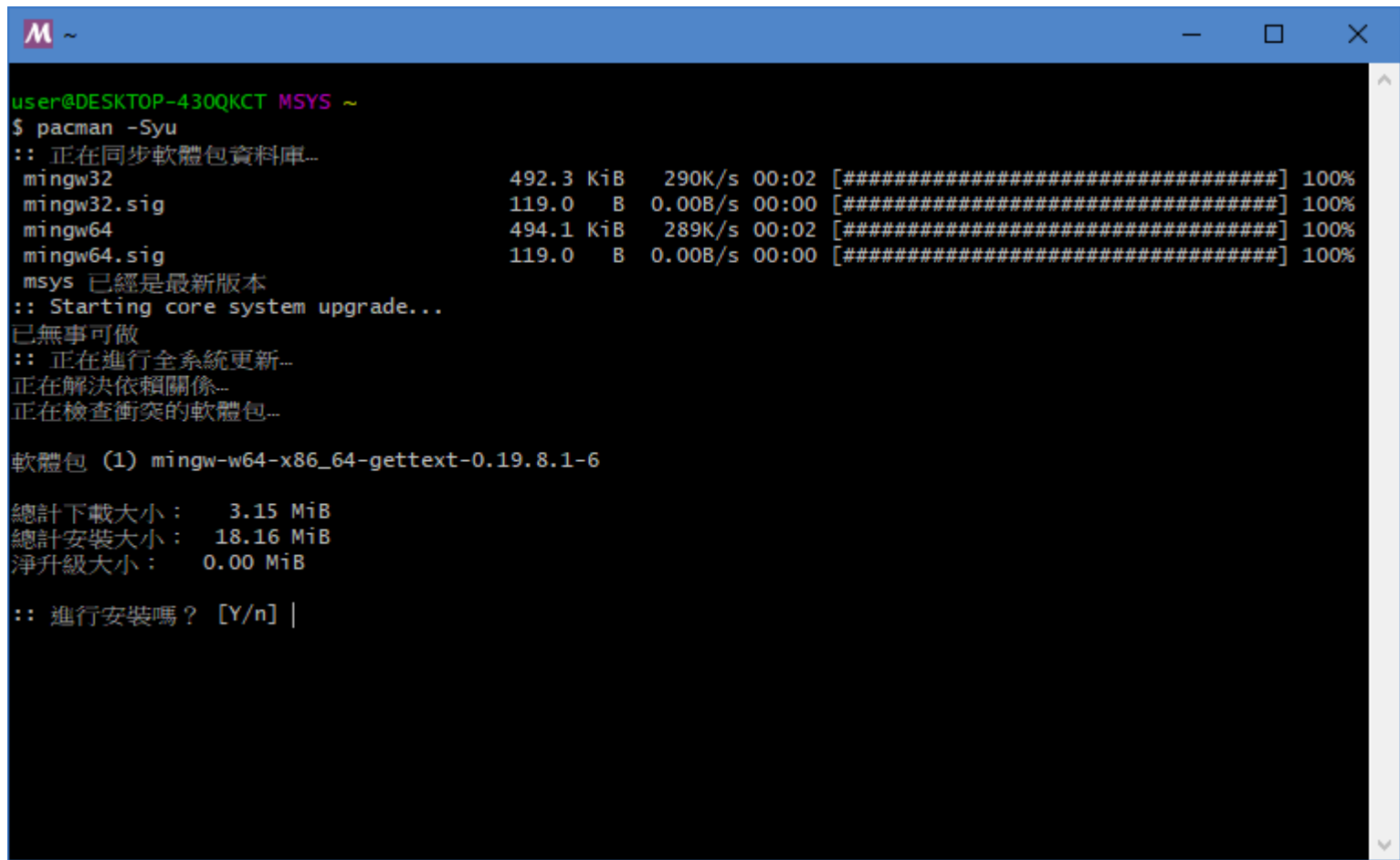
# 實作方式

- 安裝完成後，便可以開啟MSYS。



# 實作方式

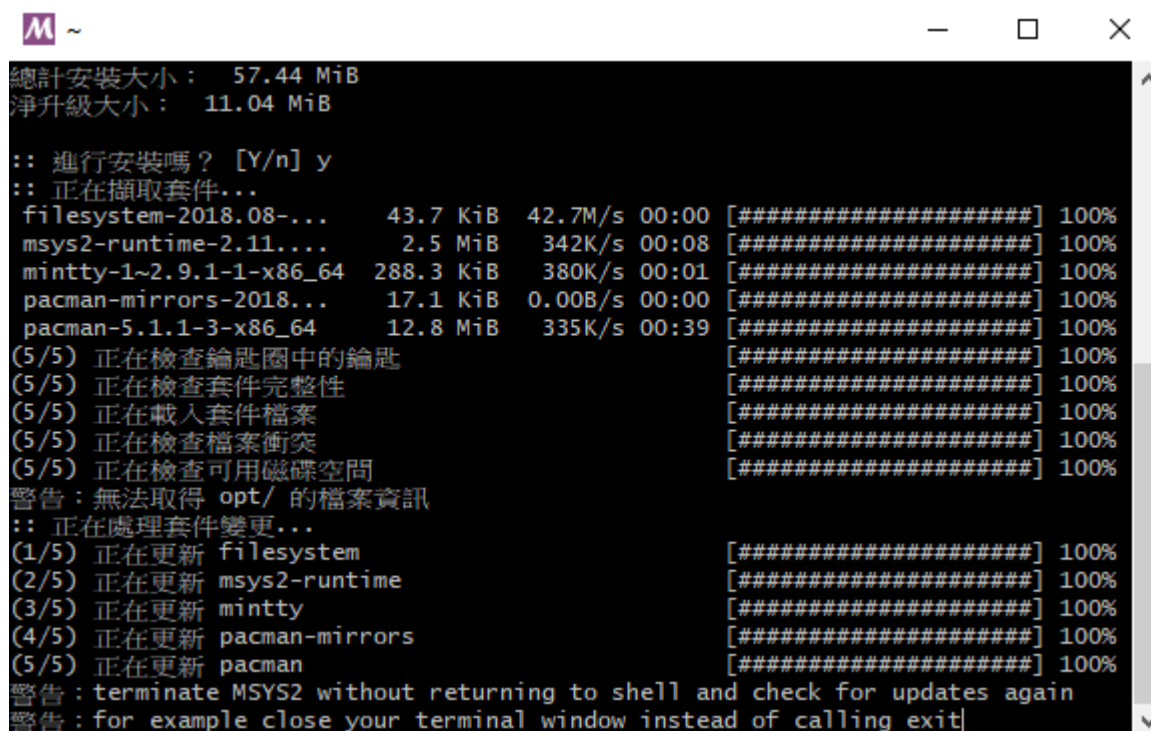
- 接著需要進行更新，請輸入 `$ pacman -Syu`



```
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 libnpth-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

- 更新git 輸入 `$ pacman -S git global`

```
$ pacman -S git global
警告：git-2.19.0-1 已經為最新一重新安裝
正在解決依賴關係...
正在檢查衝突的軟體包...

軟體包 (2) git-2.19.0-1  global-6.6.2-1

總計下載大小：    0.45 MiB
總計安裝大小：   30.01 MiB
淨升級大小：     1.67 MiB

:: 進行安裝嗎？ [Y/n] |
```

- 這裡請輸入y

# 實作方式

- \$ pacman -S git

```
$ pacman -S git
警告：git-2.19.0-1 已經為最新一重新安裝
正在解決依賴關係--
正在檢查衝突的軟體包--

軟體包 (1) git-2.19.0-1

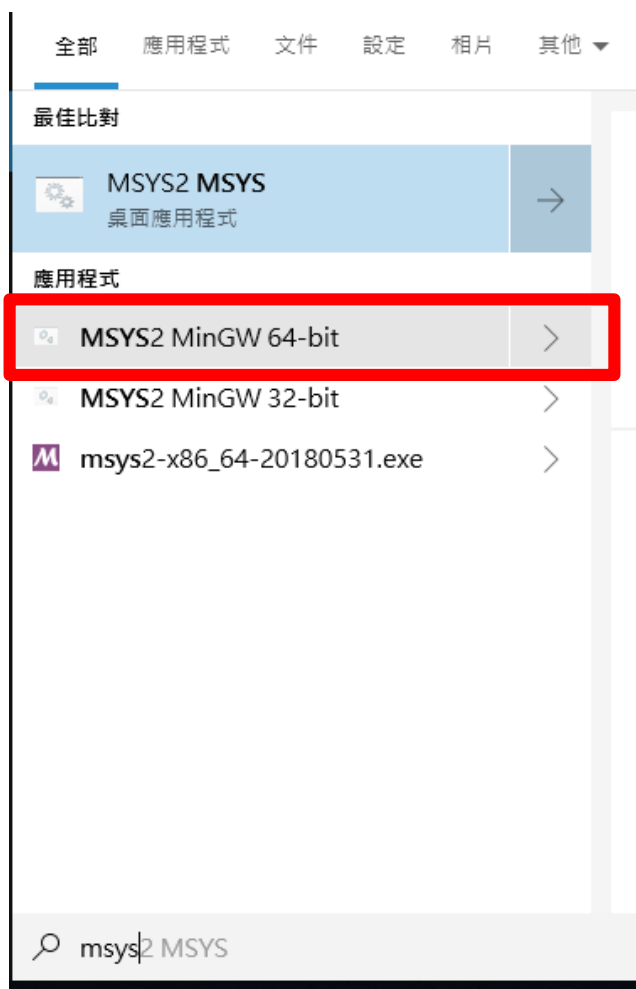
總計安裝大小： 28.34 MiB
淨升級大小： 0.00 MiB

:: 進行安裝嗎？ [Y/n]
```

- 這裡請輸入y

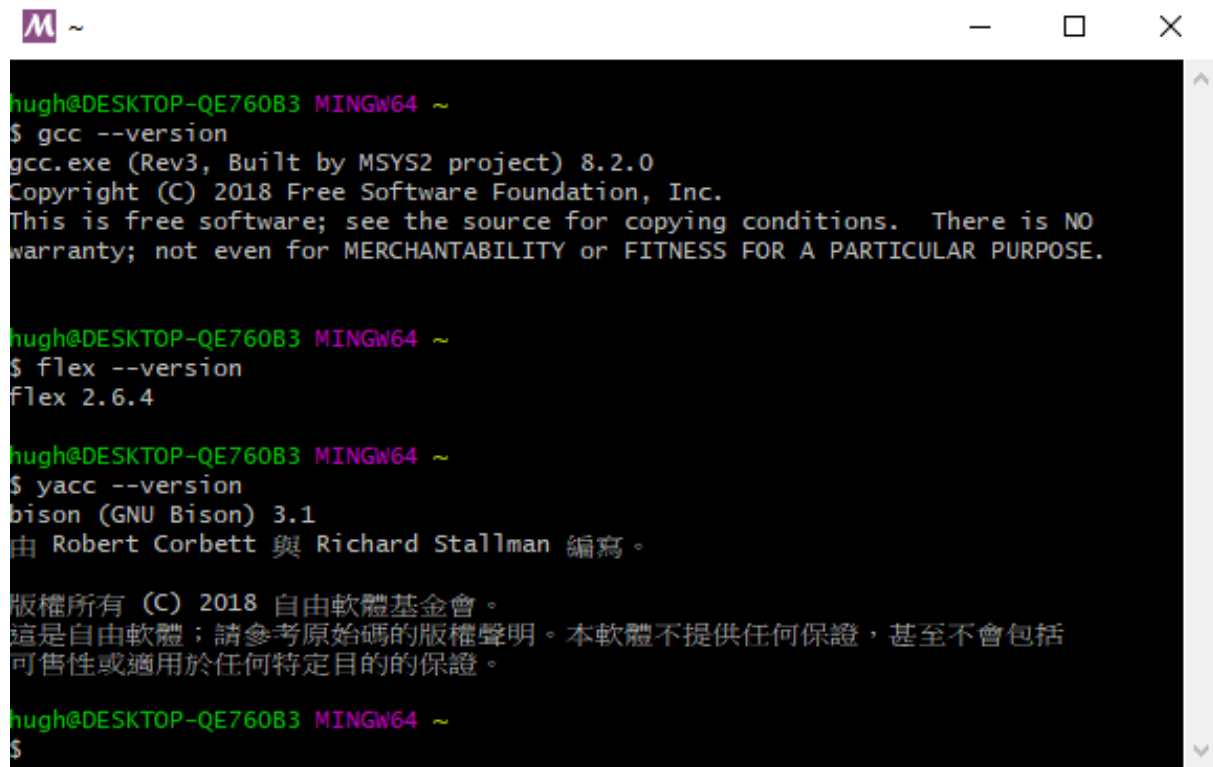
# 實作方式

關掉原本的MSYS，使用搜尋，開啟MSYS2 MinGW 64-bit



# 實作方式

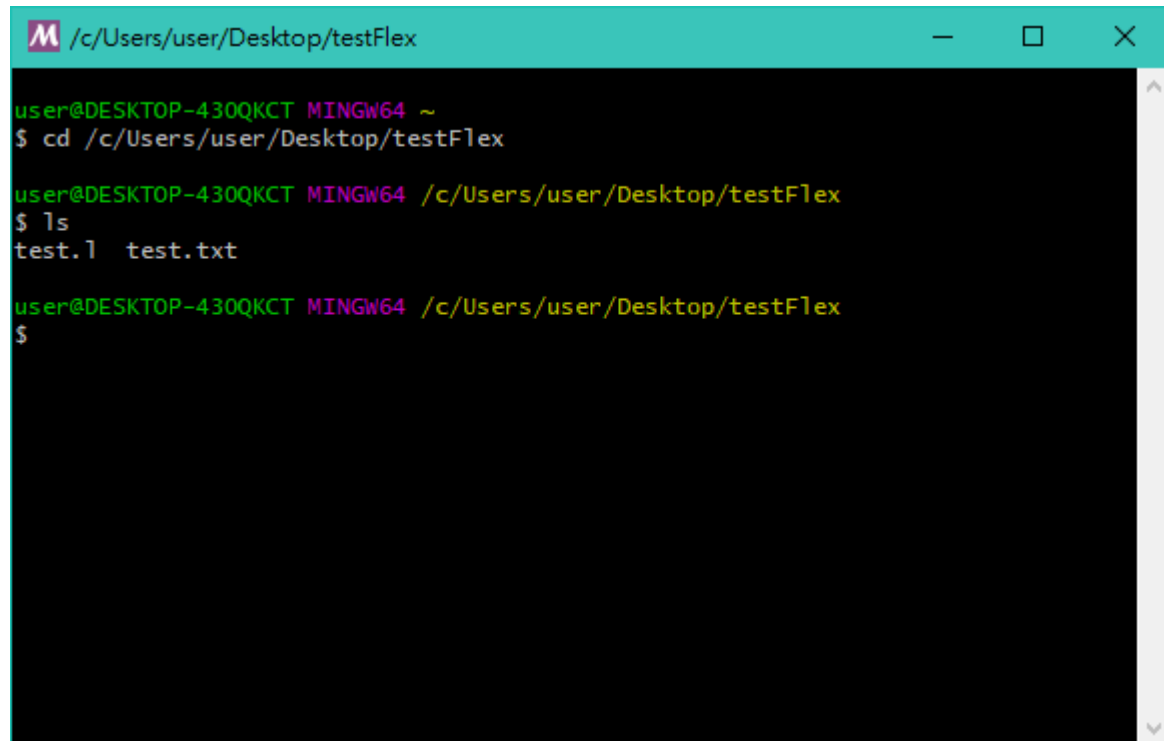
- 可以透過輸入以下指令檢查是否安裝成功
- `$ gcc --version`
- `$ flex --version`



```
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

- 我們可以利用vi指令撰寫xxx.l檔，也可以用Windows上的記事本或VS code撰寫後，再放入自己創建資料夾中。



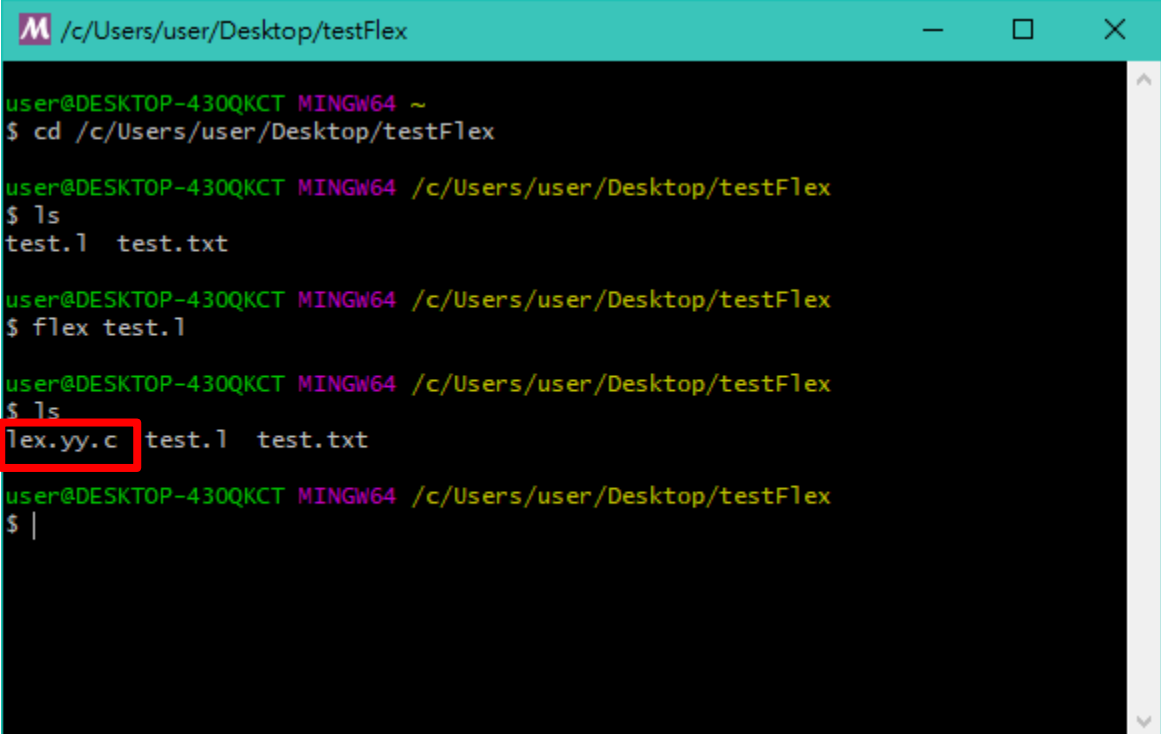
```
M /c/Users/user/Desktop/testFlex
user@DESKTOP-430QKCT MINGW64 ~
$ cd /c/Users/user/Desktop/testFlex

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ ls
test.l  test.txt

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

# 實作方式

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



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

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ ls
test.l  test.txt

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ flex test.l

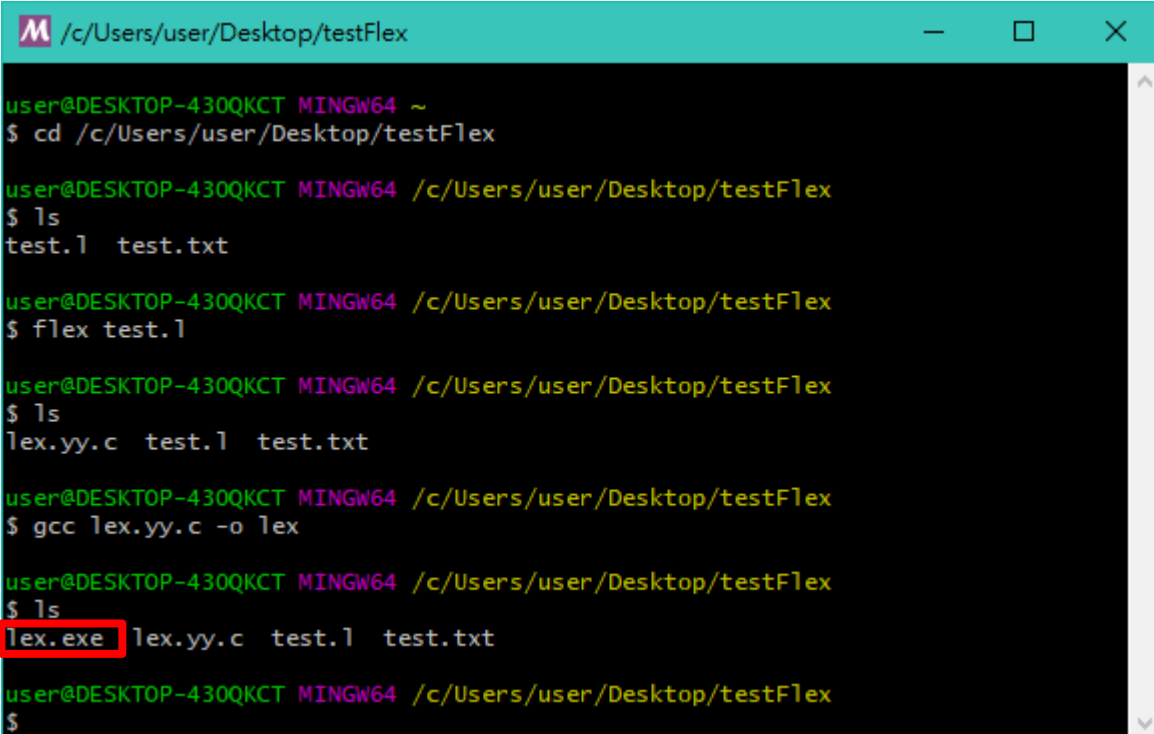
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ ls
lex.yy.c  test.l  test.txt

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



# 實作方式

- 產生了lex.yy.c檔後，我們便可以將它編譯成執行檔。
  - gcc lex.yy.c -o lex



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

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

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

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

user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$ gcc lex.yy.c -o lex

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

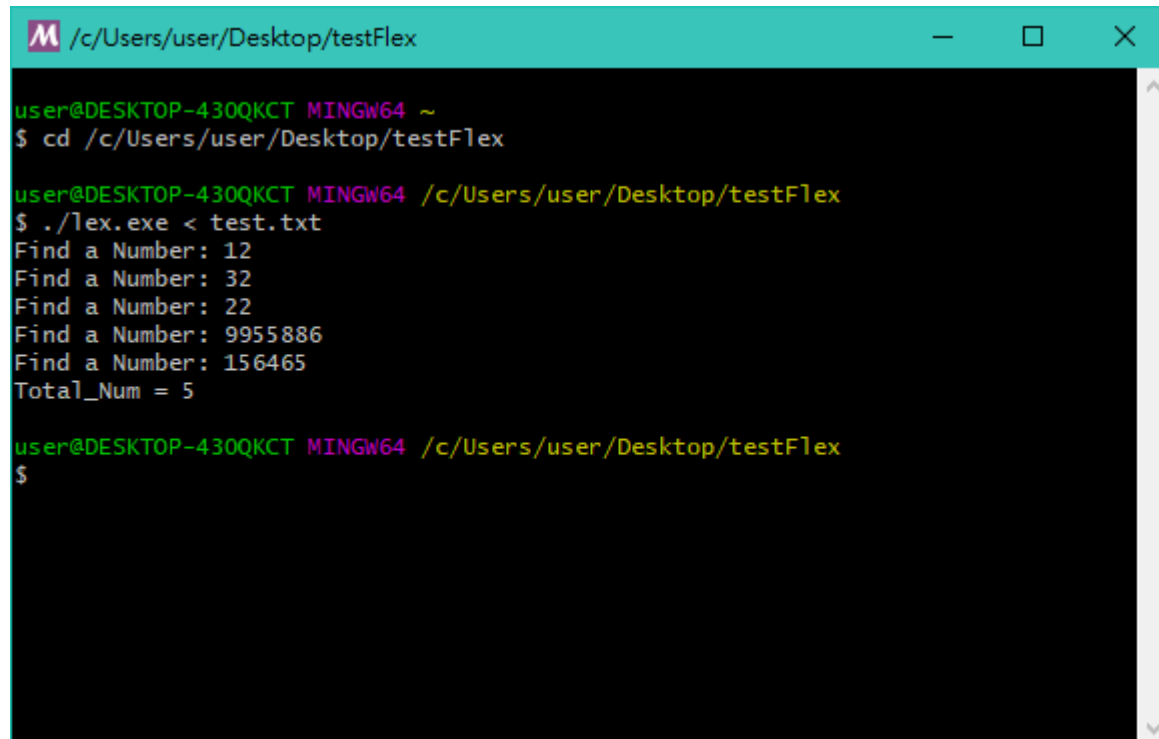
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex
$
```

# 實作方式

- 建立一測試檔test.txt：

```
12+32+22  
9955886  
156465
```

- 輸入./lex < test.txt分析：



```
M /c/Users/user/Desktop/testFlex  
user@DESKTOP-430QKCT MINGW64 ~  
$ cd /c/Users/user/Desktop/testFlex  
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex  
$ ./lex.exe < test.txt  
Find a Number: 12  
Find a Number: 32  
Find a Number: 22  
Find a Number: 9955886  
Find a Number: 156465  
Total_Num = 5  
user@DESKTOP-430QKCT MINGW64 /c/Users/user/Desktop/testFlex  
$
```

# 作業

- 撰寫一個lex檔，並完成下列功能：
  - 輸入為test.txt檔，內有浮點數(float)、整數(int)、變數(id)，並用分號及換行隔開，例如：

```
123;  
458.345;  
Total_Number123;
```

※Demo時會使用較難的範例，請自己完整測試

- 變數規則：第一個字必須為字母，後面可以接字母、底線、數字，變數長度不限。
- 利用lex辨別出test.txt中浮點數、整數、變數，並在每次辨識完後輸出float、int、id或error，並在最後印出各種Token的出現次數。

# 作業

- 作業格式

- 1.封面：作業題目、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
- 繳交報告：上傳至ilearn，繳交word檔  
10/25(五) 晚上 11:30之前

- DEMO時間：

- 10/24(四)地點：資電234
- 時間：晚上6:30-8:30
- 有問題可寄E-mail至 M0705638@o365.fcu.edu.tw,  
M0721067@o365.fcu.edu.tw

# 延伸作業

- 試著用C/C++撰寫scanner程式，
  - 輸入及輸出如作業相同，輸入檔案後，透過scanner可以判斷是數字、浮點數、變數或者錯誤，並印出各Token有幾個。
- 不用額外再寫一份報告，只要DEMO時與作業一起Demo便可額外加分(3分)。