OOP HW1

4. Debugger (15%)

請利用 debugger 執行以下 7 個指令，並截圖附在作業檔案中。

GDB(C++) 或 JDB(Java) 擇一使用即可，如果你要使用 LLDB、Visual Studio 或 VSCode 也可以接受。

以下使用 Linux g++ GDB

(0)準備

$g++ -g -o -DDEBUG OOP_HW1_406410114_C++.cpp #使用-g 編譯

```
gyc108u@csie0[3:09pm]~/OOP/HW/HW1>g++ -g -o -DDEBUG Debugger_test.cpp
gyc108u@csie0[3:12pm]~/OOP/HW/HW1>ls
-DDEBUG*  Debugger_test.cpp  Main.class  OOP_HW1_406410114_C++.cpp  OC
```

$ gdb ./-DDEBUG #進入 gdb

```
gyc108u@csie0[2:58pm]~/OOP/HW/HW1>gdb ./-DDEBUG
GNU gdb (GDB) 7.12 [GDB v7.12 for FreeBSD]
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
```

(1)印出程式碼

$list #打印程式碼 (若有中文註解需額外處理)

```
(gdb) list
4       using namespace std;
5
6       typedef struct Char_statis {  //the struct of store char and num
7           char c;
8           int num;
9       } Char_statis;
10
11
12      void boot( Char_statis char_fre_matrix[] ); //boot
13      int is_find( Char_statis char_fre_matrix[],char c ,int ch_f_num); // find appeared ,yes return inde
x,no return -1
(gdb) list
14
15      int main (int argc, char *argv[]) {
16          Char_statis char_fre_matrix[200];
17          int i,j,k,lentemp,find_result ;
18          int ch_f_num = 0;
19
20          //boot initial
21          boot(char_fre_matrix);
22
23          //begain string analysis
```

(2)設定 breakpoint

$ b 18 #設定 breakpoint 在 18 行

```
(gdb) b 18
Breakpoint 1 at 0x400b5f: file OOP_HW1_406410114_C++.cpp, line 18.
(gdb)
```

(3)向下執行程式碼(直到 breakpoint)

$ r #重頭執行到第 18 行 中間的話用 c 也可執行到直到 breakpoint

```
(gdb) r
Starting program: /.amd_mnt/cs1/host/csdata/home/under/u108/gyc108u/OOP/HW/HW1/-DDEBUG

Breakpoint 1, main (argc=1, argv=0x7fffffffea28) at OOP_HW1_406410114_C++.cpp:18
18          int ch_f_num = 0;
```

(4)向下執行一行程式碼

$ s #執行下一行

```
18          int ch_f_num = 0;
(gdb) s
19          boot(char_fre_matrix);//boot initial
```

(5)印出特定變數值(ex: 變數 a 現在的值)

$ p ch_f_num #打印出 ch_f_num 的值

```
(gdb) p ch_f_num
$1 = 0
(gdb)
```

(6)設定持續印出某變數值

$display ch_f_num #每次執行的時候都會打印 ch_f_num 的值

```
(gdb) display ch_f_num
1: ch_f_num = 0
(gdb)
```

```
(gdb) r "abc"
Starting program: /.amd_mnt/cs1/host/csdata/home/und

Breakpoint 1, main (argc=2, argv=0x7fffffffea18) at
18          int ch_f_num = 0;
1: ch_f_num = 0
(gdb) s
19          boot(char_fre_matrix);//boot initial
1: ch_f_num = 0
(gdb) c
Continuing.
a-1
b-1
c-1
[Inferior 1 (process 89492) exited normally]
```

(7)結束 debugger

$Quit #結束 Debug

```
(gdb) Quit
gyc108u@csie0[1:45pm]~/OOP/HW/HW1>
```