ITCS222 Computer Organization and Architecture Report

C program

test.c

```c
#include <stdio.h>
int main()
{
    int a = 0;
    int b = 0;
    int c = 0;
    printf("Enter number1:\n");
    scanf("%d", &a);
    printf("Enter number2:\n");
    scanf("%d", &b);
    printf("Enter number3:\n");
    scanf("%d", &c);
    int d = (a + b + c) / 3;
    printf("Answer = %d\n", d);
    return 0;
}
```

The output of C Program

```
Last login: Tue Oct 30 19:11:55 on ttys001
Thanakorns-MacBook-Pro:~ thanakornpasangthien$ /Users/thanakornpasangthien/Desktop/web_learning/comarch_project/c_p
rogram/avg ; exit;
Enter number1:
11
Enter number2:
12
Enter number3:
13
Answer = 12
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

Assembly Program

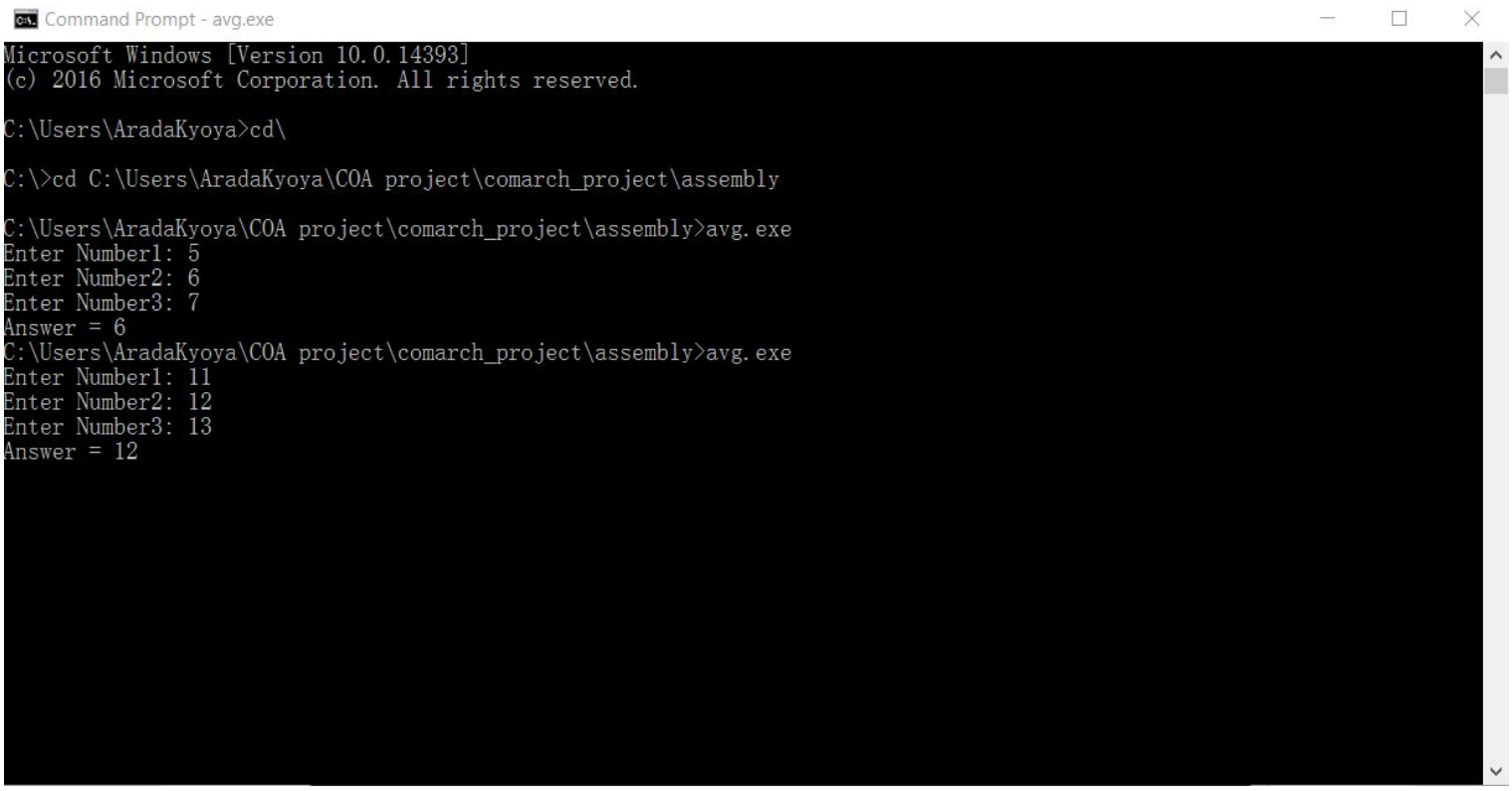
avg.asm

```
;Thanakorn Pasangthien 6088109
%include "asm_io.inc"

segment _DATA public align=4 class=DATA use32
enter1          db      "Enter Number1: ", 0     ;first string
enter2          db      "Enter Number2: ", 0     ;second string
enter3          db      "Enter Number3: ", 0     ;third string
answer          db      "Answer = ",0            ;fourth string

segment _BSS public align=4 class=BSS use32
num1            resd    16          ;reserve for first number
num2            resd    16          ;reserve for second number
num3            resd    16          ;reserve for third number

group DGROUP _BSS _DATA

segment _TEXT public align=1 class=CODE use32
        global  _asm_main
_asm_main:
        enter   0,0
        pusha

        mov     eax,  enter1
        call    print_string    ;print the first string

        call    read_int
        mov     [num1], eax     ;input num1

        mov     eax,  enter2    ;print the second string
        call    print_string

        call    read_int
        mov     [num2], eax     ;input num2
```

avg.asm

```
34
35          mov     eax,  enter3    ;print the third string
36          call    print_string
37
38          call    read_int
39          mov     [num3], eax     ;input num3
40
41          mov     eax,answer      ;print the fourth string
42          call    print_string
43
44          mov eax,[num1]
45          add eax,[num2]
46          add eax,[num3]  ;sum num1 num2 and num3
47
48          mov edx,0
49          mov ecx,3
50          div ecx
51          push eax
52          call print_int ;print the average
53
54          ret
```

The output of Assembly Program



*I used Arada's computer to execute the assembly program that I wrote because I use mac book which can't execute the .exe file

*For the C program, I have used my own mac to execute so, there I include both the Mac version and window version in the CD.