Lab Exercise 7

In this exercise, you are expected to understand the concept of runtime stack as implemented in assembly code. You have to use your knowledge on runtime stack to convert the assembly code provided in this file into its equivalent c code.

The assembly code that you have to convert into c code is as follow.

```
00000000004006ba:
        push    rbp
        movq    rsp, rbp
        movl    edi, -0x4(rbp)
        movl    esi, -0x8(rbp)
        movl    edx, -0xc(rbp)
        movl    -0x4(rbp), edx
        movl    -0x8(rbp), eax
        add     eax, edx
        movl    -0xc(rbp), eax
        add     edx, eax
        pop     rbp
        ret
0000000000400984:
        push    rbp
        movq    rsp, rbp
        sub     $24, rsp
        movl    edi, -0x4(rbp)
        movl    esi, -0x8(rbp)
        movl    edx, -0xc(rbp)
        movl    ecx, -0xf(rbp)
        movl    r8d, -0x14(rbp)
        movl    -0xf(rbp), edx
        movl    -0x14(rbp), eax
        add     eax, edx
        movl    -0x8(rbp), eax
        imul    -0xc(rbp), eax
        movl    eax, ecx
        movl    -0x4(rbp), eax
        movl    ecx, esi
        movl    eax, edi
        call    0x4006ba
        movq    rbp, rsp
        pop     rbp
        ret
0000000000400886:
        push    rbp
        movq    rsp, rbp
        sub     $32, rsp
        movl    1, -0x4(rbp)
        movl    2, -0x8(rbp)
        movl    3, -0xc(rbp)
        movl    4, -0xf(rbp)
        movl    5, -0x14(rbp)
```

```
45          movl    -0x14(rbp), edi
46          movl    -0xf(rbp), ecx
47          movl    -0xc(rbp), edx
48          movl    -0x8(rbp), esi
49          movl    -0x4(rbp), eax
50          movl    edi, r8d
51          movl    eax, edi
52          call    0x400984
53          movq    rbp, rsp
54          pop     rbp
55          ret
56
```

This code is actually a part of a larger code. In the larger code, originally there are five functions. These functions and the addresses of their first instructions are listed as follows.

- Function: main, address of its first instruction: 0x400a86
- Function: foo, address of its first instruction: 0x400886
- Function: foo1, address of its first instruction: 0x400984
- Function: foo2, address of its first instruction: 0x4006ba
- Function: foo3, address of its first instruction: 0x4005b6

Based on the assembly code provided above and the addresses of the first instructions of all functions in the original code, please write down the c code version of the assembly code shown above, which shows only three out of the original five functions. The names of the functions in your c code must be selected accurately from the functions in the list. In your c code, you are free to choose the names of the variables used in each function as long as they behave consistently like the memory addresses and/or registers that they represent in the assembly code.

After finishing your task, write your name as a comment in your c code, and submit it to the provided link in Blackboard.