CIS657 Programming Assignment 4

In this assignment you will properly use semaphores to ensure the output of the program is as expected. If you do not place the semaphores correctly, you will most likely end up in deadlock, and you will need to exit your program with CTRL-C

Task 1. (50 points)

The source code with some missing pieces for the program `sum` is below (`sum.c`). There is 50 points earned for a working, correct program. Partial credit will be assigned as 5 points for each of 8 correct answers. (A-F, X and Y)

```
 1 #include <stdio.h>
 2 #include <pthread.h>
 3 #include <semaphore.h>
 4
 5 int total = 0, done = 0;
 6 sem_t s1, s2;
 7
 8 void sum(int n){
 9     for (int i=1; i<=n; i++){
10         sem_wait(A);
11         total += i;
12         printf("worker: processed i = %d\n", i);
13     sem_post(B); 14
       }
15 }
16
17 void finish(){
18     sem_wait(C);
19     done = 1;
20     sem_post(D);
21     printf("worker done\n");
22 }
23
24 void *worker(void *arg){
25     int num = (int) arg;
26     printf("worker: received n = %d\n", num);
27     sum(num);
28     finish();
29     return NULL;
30 }
31
```

```
32 void thr_wait(){
33     while (1){
34         sem_wait(E);
35         if (done==1){
36             break;
37         }
38         printf("main: running total = %d\n", total);
39     sem_post(F); 40
       }
41 }
42
43 int main(int argc, char *argv[]){
44     printf("main: begin\n");
45     sem_init(&s1, 0, X);
46     sem_init(&s2, 0, Y);
47     pthread_t t;
48     pthread_create(&t, NULL, worker, (void *)3);
49     thr_wait();
50     printf("main: outcome = %d\n", total);
51     printf("main: end\n");
52 }
```

The missing pieces are:

- A in line 10
- B in line 13
- C in line 18
- D in line 20
- E in line 34
- F in line 39
- X in line 45
- Y in line 46

State what A, B, C, D, E, F, X, and Y should be for the program to output the following when run.

```
main: begin
worker: received n = 3
worker: processed i = 1
main: running total = 1
worker: processed i = 2
main: running total = 3
worker: processed i = 3
main: running total = 6
worker: done
main: outcome = 6
main: end
```