

Problem 1

Below is the screenshot of the program and the command-line of my solution for Homework5 Problem1.

The screenshot shows a code editor with the following C code for `lab5_problem1.c`:

```
0  int main(int argc, char* argv[]) {
7      int seqNum = atoi(argv[1]);
8      int diff = atoi(argv[2]);
9
10     if (seqNum < 0) {
11         printf("Invalid length of sequence!\n");
12         exit(1);
13     }
14
15     int pipeLine[2];
16     pipe(pipeLine);
17
18     pid_t childPid;
19
20     childPid = fork();
21     if (childPid < 0) {
22         printf("Fork Failed!\n");
23         exit(1);
24     }
25
26     if (childPid == 0) {
27         //In child process, we close the read side of the pipe
28         close(pipeLine[0]);
29         for (int i = 0; i < seqNum; i++) {
30             int seqVal = 1 * diff;
31             char valueText[20];
32             sprintf(valueText, "%d", seqVal);
33             printf("In child process, the next number is: %d\n", seqVal);
34             write(pipeLine[1], valueText, strlen(valueText) + 1);
35         }
36     }
```

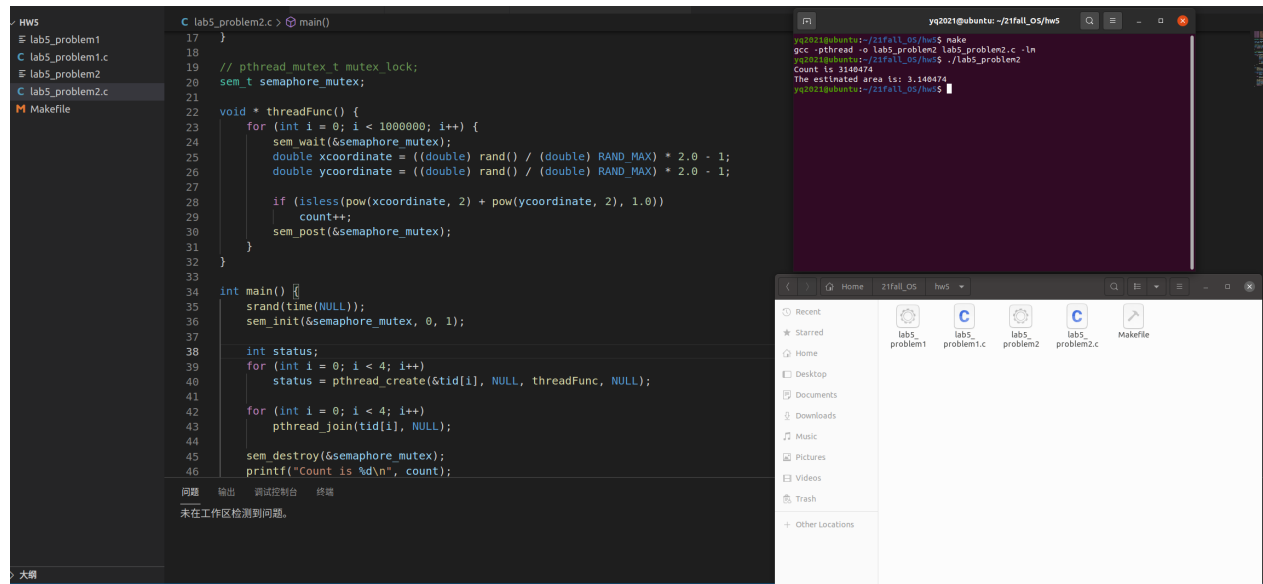
The terminal output shows the execution of the program with arguments 7 and 7:

```
yq2021@ubuntu:~/21fall_05/hw5$ make
gcc -o lab5_problem1 lab5_problems.c
yq2021@ubuntu:~/21fall_05/hw5$ ./lab5_problem1 7 7
In child process, the next number is: 0
In child process, the next number is: 7
In child process, the next number is: 14
In child process, the next number is: 21
In child process, the next number is: 28
In child process, the next number is: 35
In child process, the next number is: 42
In parent process, read value is: 0
yq2021@ubuntu:~/21fall_05/hw5$
```

Figure 1: Screenshot of program and command-line

Problem 2

Below is the screenshot of the program and the command-line of my solution for Homework5 Problem2.



```
HW5
├─ lab5_problem1
├─ lab5_problem1.c
├─ lab5_problem2
├─ lab5_problem2.c
└─ Makefile

C lab5_problem2.c > main()
17 }
18
19 // pthread mutex t_mutex_lock;
20 sem_t semaphore_mutex;
21
22 void * threadFunc() {
23     for (int i = 0; i < 1000000; i++) {
24         sem_wait(&semaphore_mutex);
25         double xcoordinate = ((double) rand() / (double) RAND_MAX) * 2.0 - 1;
26         double ycoordinate = ((double) rand() / (double) RAND_MAX) * 2.0 - 1;
27
28         if (isless(pow(xcoordinate, 2) + pow(ycoordinate, 2), 1.0))
29             count++;
30         sem_post(&semaphore_mutex);
31     }
32 }
33
34 int main() {
35     srand(time(NULL));
36     sem_init(&semaphore_mutex, 0, 1);
37
38     int status;
39     for (int i = 0; i < 4; i++)
40         status = pthread_create(&tid[i], NULL, threadFunc, NULL);
41
42     for (int i = 0; i < 4; i++)
43         pthread_join(tid[i], NULL);
44
45     sem_destroy(&semaphore_mutex);
46     printf("Count is %d\n", count);
47 }

问题 输出 调试控制台 终端
未在工作区检测到问题。
```

```
yq2021@ubuntu:~/21fall_OS/hw5$ make
gcc -pthread -o lab5_problem2 lab5_problem2.c -ln
yq2021@ubuntu:~/21fall_OS/hw5$ ./lab5_problem2
Count is 3140474
The estimated area is: 3.140474
yq2021@ubuntu:~/21fall_OS/hw5$
```

Figure 2: Screenshot of program and command-line