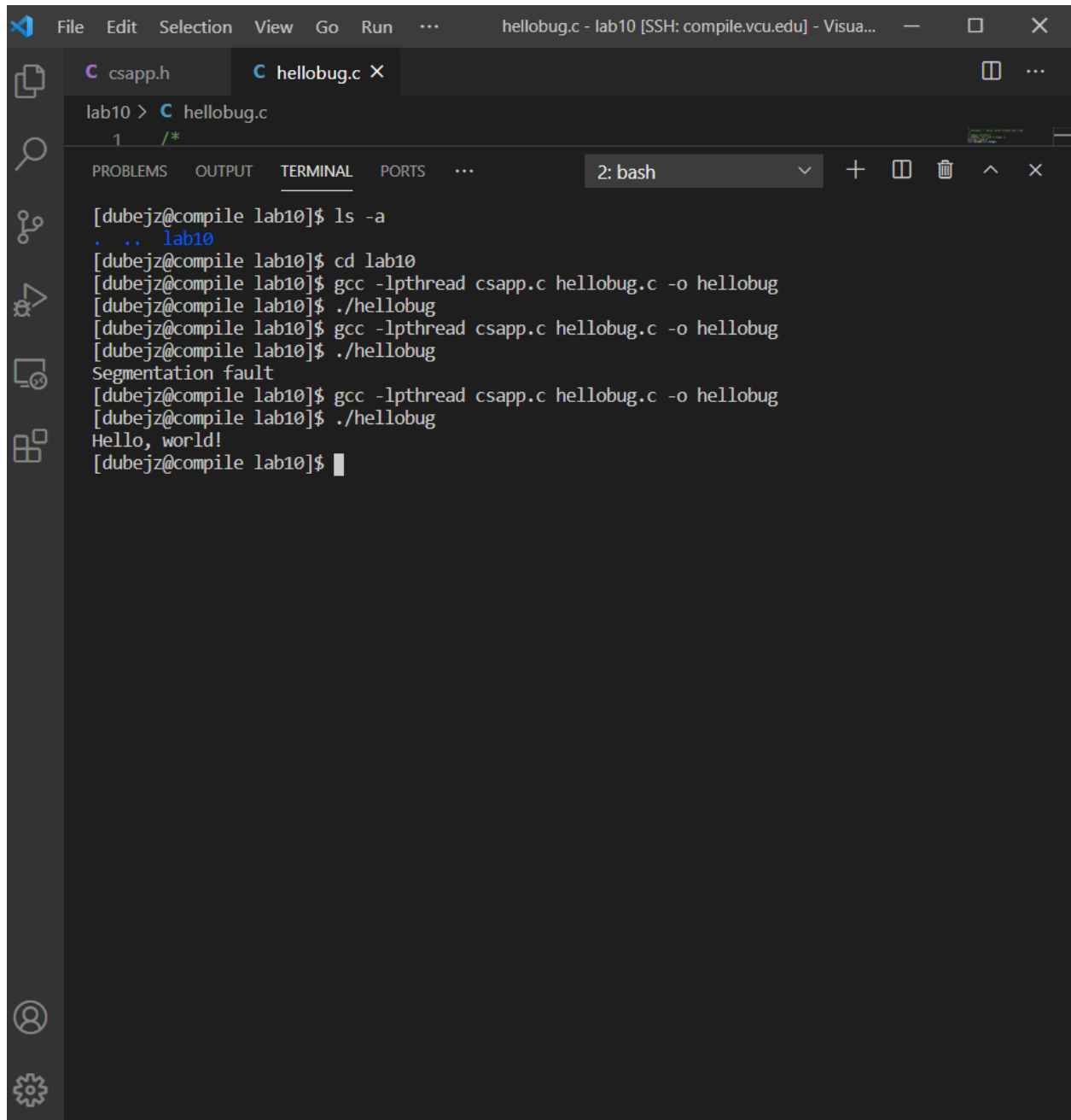


Jordan Dube
CMSC257
Lab 10
Screenshots

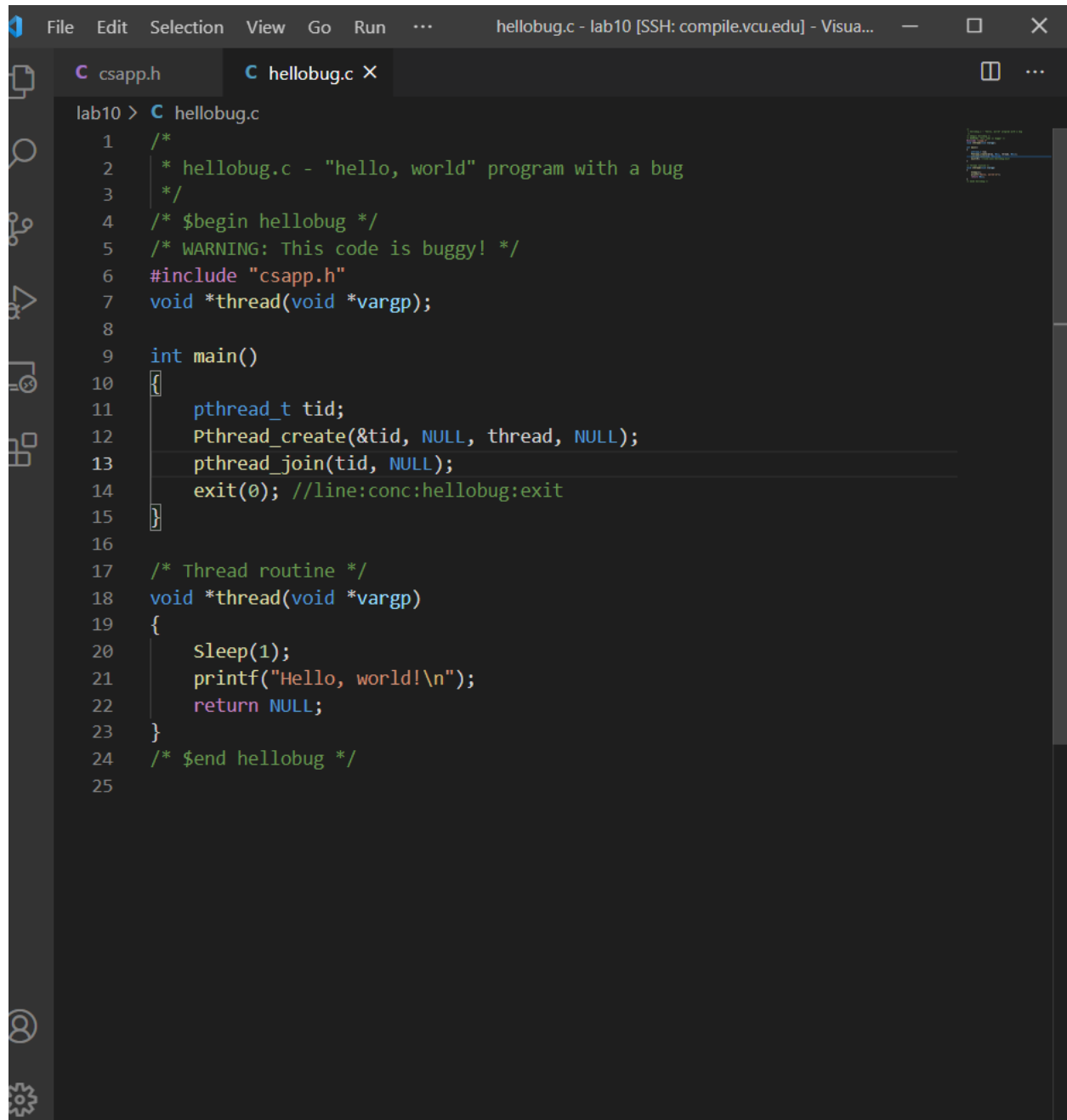
1a)



The screenshot shows a Visual Studio Code window with a terminal open. The terminal is titled '2: bash' and shows the following commands and output:

```
lab10 > C hellobug.c
1 /*
[dubejz@compile lab10]$ ls -a
. .. lab10
[dubejz@compile lab10]$ cd lab10
[dubejz@compile lab10]$ gcc -lpthread csapp.c hellobug.c -o hellobug
[dubejz@compile lab10]$ ./hellobug
[dubejz@compile lab10]$ gcc -lpthread csapp.c hellobug.c -o hellobug
[dubejz@compile lab10]$ ./hellobug
Segmentation fault
[dubejz@compile lab10]$ gcc -lpthread csapp.c hellobug.c -o hellobug
[dubejz@compile lab10]$ ./hellobug
Hello, world!
[dubejz@compile lab10]$
```

1b)



The image shows a Visual Studio Code editor window with the title bar 'hellobug.c - lab10 [SSH: compile.vcu.edu] - Visua...'. The editor has two tabs: 'csapp.h' and 'hellobug.c'. The 'hellobug.c' tab is active, showing the following C code:

```
lab10 > C hellobug.c
1  /*
2  * hellobug.c - "hello, world" program with a bug
3  */
4  /* $begin hellobug */
5  /* WARNING: This code is buggy! */
6  #include "csapp.h"
7  void *thread(void *vargp);
8
9  int main()
10 {
11     pthread_t tid;
12     Pthread_create(&tid, NULL, thread, NULL);
13     pthread_join(tid, NULL);
14     exit(0); //line:conc:hellobug:exit
15 }
16
17 /* Thread routine */
18 void *thread(void *vargp)
19 {
20     Sleep(1);
21     printf("Hello, world!\n");
22     return NULL;
23 }
24 /* $end hellobug */
25
```

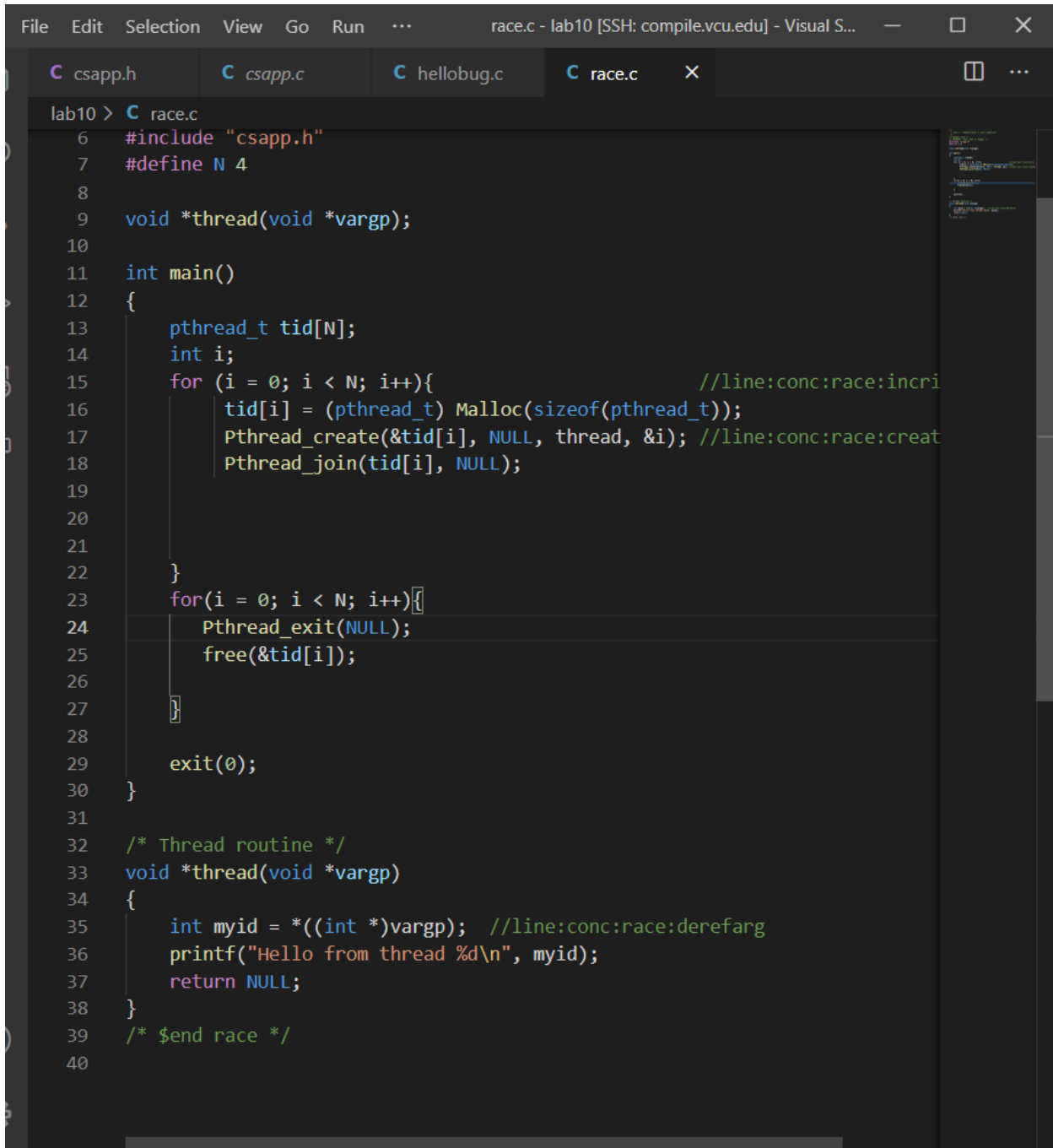
2a) (20 iterations of ./race)

[illegible]

[illegible]

[illegible]

2b)



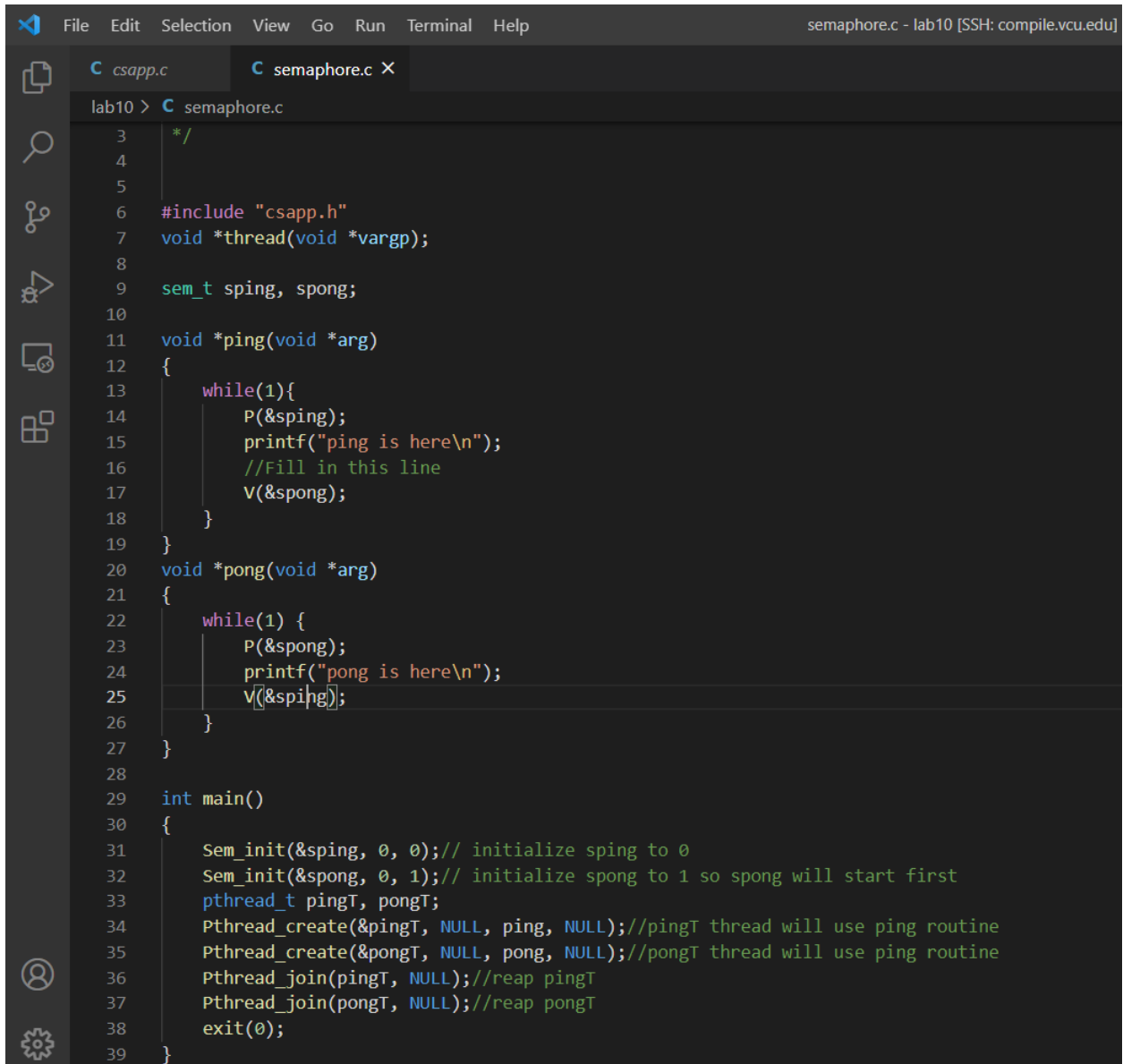
```
File Edit Selection View Go Run ... race.c - lab10 [SSH: compile.vcu.edu] - Visual S...
C csapp.h C csapp.c C hellobug.c C race.c X ...

lab10 > C race.c
6 #include "csapp.h"
7 #define N 4
8
9 void *thread(void *vargp);
10
11 int main()
12 {
13     pthread_t tid[N];
14     int i;
15     for (i = 0; i < N; i++){ //line:conc:race:incritid
16         tid[i] = (pthread_t) Malloc(sizeof(pthread_t));
17         Pthread_create(&tid[i], NULL, thread, &i); //line:conc:race:creatid
18         Pthread_join(tid[i], NULL);
19     }
20
21     for(i = 0; i < N; i++){
22         Pthread_exit(NULL);
23         free(&tid[i]);
24     }
25
26     exit(0);
27 }
28
29 /* Thread routine */
30 void *thread(void *vargp)
31 {
32     int myid = *((int *)vargp); //line:conc:race:dereferarg
33     printf("Hello from thread %d\n", myid);
34     return NULL;
35 }
36
37 /* $end race */
38
39
```

3a)

[illegible]

3b)



```
File Edit Selection View Go Run Terminal Help semaphore.c - lab10 [SSH: compile.vcu.edu]
C csapp.c C semaphore.c X
lab10 > C semaphore.c
3  /*
4
5
6  #include "csapp.h"
7  void *thread(void *vargp);
8
9  sem_t sping, spong;
10
11 void *ping(void *arg)
12 {
13     while(1){
14         P(&sping);
15         printf("ping is here\n");
16         //Fill in this line
17         V(&spong);
18     }
19 }
20 void *pong(void *arg)
21 {
22     while(1) {
23         P(&spong);
24         printf("pong is here\n");
25         V(&sping);
26     }
27 }
28
29 int main()
30 {
31     Sem_init(&sping, 0, 0); // initialize sping to 0
32     Sem_init(&spong, 0, 1); // initialize spong to 1 so spong will start first
33     pthread_t pingT, pongT;
34     Pthread_create(&pingT, NULL, ping, NULL); // pingT thread will use ping routine
35     Pthread_create(&pongT, NULL, pong, NULL); // pongT thread will use ping routine
36     Pthread_join(pingT, NULL); // reap pingT
37     Pthread_join(pongT, NULL); // reap pongT
38     exit(0);
39 }
```