

Lab 6 - Stein Hanssen Sæstad

Github @ steinhs

3.1 – Exercise 1

Output:

```
$ ./Exercise1
Initial values: a = 0, b = 1000
Decreasing b finished thanks to the main thread!
Increasing a finished thanks to the second thread!
Final values: a= 100, b= 100
```

3.2 - Exercise 2

Thread 2 message is missing because ThreadPrint2 is never called. In the second thread created (t2), it is calling ThreadPrint1 a second time instead of ThreadPrint2.

Also main is finishing too quick for the second thread to be executed and sometimes too quick for thread one.

This is because we do not tell the program to wait for any running threads before finishing.

Adding `pthread_join()` for thread one and two in front of "I am main thread", we can make sure that they are run before main print its message and finishes.

```
pthread_join(t1, NULL);
pthread_join(t2, NULL);
printf("I am the main thread");
```

Output:

```
$ ./Exercise2
This is Thread 1
This is Thread 2
I am the main thread
```

3.3 - Exercise 3

Not completed.