

Oppgave1.a)

```
aleksawo@itstud:~/htdocs/oblig5$ gcc -pthread -o oppg1 oppgave_1.c
aleksawo@itstud:~/htdocs/oblig5$ ./oppg1
139510 134152 248908 289585 348168
aleksawo@itstud:~/htdocs/oblig5$ ./oppg1
100000 140564 241208 240370 340370
aleksawo@itstud:~/htdocs/oblig5$ ./oppg1
100000 143014 237241 256906 343246
aleksawo@itstud:~/htdocs/oblig5$ ./oppg1
94555 139094 220427 254755 306657
aleksawo@itstud:~/htdocs/oblig5$ ./oppg1
100000 129091 214493 247241 229091
```

Du vil se at tallene som skrives ut av trådene forandrer seg i hver kjøring. Gi en kort forklaring på hvorfor dette skjer: Fordi flere tråder prøver å oppdatere globale variabelen samtidig uten synkronisering.

Oppgave2.a)

```
aleksawo@itstud:~/htdocs/oblig5$ time ./oppgave_2_1
.
.
.
.
.
.
.
.
.
.
1000000000 iterations completed!

real    0m2.581s
user    0m2.581s
sys     0m0.001s
```

2.b)

```
aleksawo@itstud:~/htdocs/oblig5$ gcc -pthread -o oppgave_2_2 oppgave_2_2.c
aleksawo@itstud:~/htdocs/oblig5$ time ./oppgave_2_2
.
.
.
.
.
.
.
.
.
.
1000000000 iterations completed!

real    0m1.198s
user    0m2.177s
sys     0m0.008s
```

2.c). Gi en forklaring på hvorfor trådprogrammet blir omtrent dobbelt så raskt:

Fordi oppgaver deles mellom to tråder som kan kjøre parallelt. Hvis CPU har minst 2 kjerner kan trådene kjøre samtidig. Bruk av ressursene er mer effektivt.