

Oppgave 1

a)

- Programmet gir forskjellig resultat hver gang fordi `count++` og `count--` ikke er atomiske operasjoner (**en operasjon som skjer i ett eneste udelelig steg**). Begge trådene kan lese og endre den delte variabelen samtidig, slik at oppdateringer overskriver hverandre. Dette kalles en race condition.

b)

- Se endringer i `oppgave_1.c`

Oppgave 2

I dette programmet ser det ut som begge trådene burde skrive ut meldinger, siden begge kaller `printf()`. Likevel skjer det aldri. Grunnen er at de to trådene låser ressursene i motsatt rekkefølge. Den ene tråden låser først `mutex1` og venter så på `mutex2`. Den andre tråden låser først `mutex2` og venter så på `mutex1`. Når dette skjer samtidig, får vi en situasjon som kalles **deadlock** i C verdenen.

Oppgave 3

a)

Se `oppgave_3.c`

b)

Når programmet kjøres med få tråder `n=5`, stopper det etter en stund. Dette skjer fordi alle trådene kan ha kastet 1 og dermed blitt blokkert på den betingede variabelen.

Når programmet derimot kjøres med mange tråder `n=50`, fortsetter det lenge. Grunnen er at det nesten alltid er noen tråder som ikke er blokkert, og disse kaster før eller siden 6.