a)

sum muuttujan pitäisi olla ohjelman ajo lopussa 20000000

b)

sum muuttuja vaihetelee arvo vaihtelee 9868529-11911599 välillä.

Koodi ajettu noin 20 kertaa.

c)

Molemmat threadit yrittävät muuttaa sum muuttujan arvoa samanaikaisesti,

jonka takia päädytään ei toivottuun lopputulokseen.

esim.

tid\_1 lukee sum muuttujan arvon omaan muistiin

tid\_2 lukee sum muuttujan arvon omaan muistiin

tid\_2 lisää omaan lukuunsa 1

tid\_2 kirjoitta sum muuttujaan uuden arvon

tid\_1 lisää omaan lukuunsa 1

tid\_1 kirjoitta sum muuttujaan uuden arvon

Tässä tilanteessa sum muuttujan arvo nousee yhdellä eikä kahdella.

Jos toinen thread käy loopin useampaan kertaan läpi sillä aikaa,

kun toinen thread käy loopin kerran läpi, sum muuttujan arvon tila

voi näyttää seuraavanlaiselta:

5 -> 6 -> 7 -> 8 -> 5 -> 6

d)

Yhden mahdollisen ratkaisun void \*countgold(void \*param); funktio:

void \*countgold(void \*param) {

int i; // local to each thread

for (i = 0; i < 10000000; i++) {}

sum += i;

return NULL;

}

Tässä ratkaisussa sum:n arvoa muutetaan vasta countgold funktion lopussa

ja vain kerran.

Tämä ratkaisukaan ei ole täydellinen, koska on mahdollista että molemmissa

threadeissä ajetaan sum += i; samanaikaisesti.

Kaikista varmin ratkaisu olisi odottaa tid1 ajon loppuun ennen kuin tid2

käynnistettään.