1 Summan laskeminen rinnakkaisesti (1p)

Kirjoita ohjelma, joka laskee (suuren) taulukon (esim. vektorin) elementtien summan käyttäen useita säikeitä. Jaa taulukko yhtä suuriin osiin, anna jokainen osa eri säikeelle ja anna niiden laskea oman osansa summa. Lopuksi yhdistä kaikkien säikeiden tulokset saadaksesi kokonaissumman. Pääohjelma luo säikeet ja odottaa niiden päättymistä. Varmista, että tulos on oikein.

Luo säikeet std::thread-luokan avulla.

2 Tililtä nostaminen (1p)

Kirjoita ohjelma, jossa on kaksi säiettä, ja jotka käsittelevät yhteistä pankkitiliä. Käytä säikeiden luomiseen std::thread-luokkaa. Ensimmäinen säie tallettaa rahaa tilille, kun taas toinen nostaa sitä. Nosto- ja talletustapahtumia tulisi olla paljon (tuhansia). Pääohjelma luo säikeet ja odottaa niiden päättymistä.

Tarkista tapahtumien jälkeen, että tilin saldo on korrekti. Jos/kun saldo on väärä, käytä mutexia tapahtumien suojaamiseen. Muokkaa ratkaisua lopuksi niin, että mutexia käytetään std::lock_guard:n avulla, eksplisiittisten lock()- ja unlock()-kutsujen sijasta.

3 Rinnakkaista pelin laskentaa (2p)

Oletetaan, että pelin suorittamille toimenpiteille (tekoälyn ajaminen, pelimaailman päivitys, jne.) on määritelty yhteinen yliluokka Game_Task. Luokassa on määritelty puhdas virtuaalifunktio perform(), joka suorittaa ko. toimenpiteen:

```
virtual void Game_Task::perform() = 0;
```

Kaikki toimenpiteet ovat toisistaan riippumattomia (eivät käytä samaa dataa tms.).

Peli ylläpitää vektoria toimenpiteistä. Toimenpiteet suoritetaan peräkkäin silmukassa:

```
std::vector<Game_Task*> tasks;
…
int number_of_tasks = tasks.size();
for (int i = 0; i < number_of_tasks; i++)
{
        task[i]->perform();
}
// Continue only after all tasks are complete!
```

Hahmottele, miten em. silmukka voitaisiin rinnakkaistaa käyttäen std::thread -luokkaa. Etsi ohjelmallinen tapa selvittää, kuinka montaa samanaikaista säiettä laitteistosi pystyy suorittamaan fyysisesti; käytä rinnakkaistamiseen yksi säie vähemmän.

Luo testaamista varten yliluokka Game_Task ja sille muutama aliluokka, joiden perform-funktio kuluttaa sopivasti prosessoriaikaa johonkin. Mittaa rinnakkaistamisesta saamasi suoritusaikahyöty.