

### L3. Lygiagretusis programavimas grafiniams procesoriams

Duomenų failai naudojami tie patys, kaip L1 ir L2 metu. Duomenų failo pavyzdys:

```
{
  "students": [
    {"name": "Antanas", "year": 1, "grade": 6.95},
    {"name": "Kazys", "year": 2, "grade": 8.65},
    {"name": "Petras", "year": 2, "grade": 7.01},
    {"name": "Sonata", "year": 3, "grade": 9.13},
    {"name": "Jonas", "year": 1, "grade": 6.95},
    {"name": "Martynas", "year": 3, "grade": 9.13},
    {"name": "Artūras", "year": 2, "grade": 7.01},
    {"name": "Vacys", "year": 2, "grade": 8.65},
    {"name": "Robertas", "year": 3, "grade": 6.43},
    {"name": "Mykolas", "year": 1, "grade": 6.95},
    {"name": "Aldona", "year": 3, "grade": 9.13},
    {"name": "Asta", "year": 2, "grade": 7.01},
    {"name": "Viktoras", "year": 2, "grade": 8.65},
    {"name": "Artūras", "year": 5, "grade": 8.32},
    {"name": "Vytas", "year": 3, "grade": 7.85},
    {"name": "Jonas", "year": 1, "grade": 6.95},
    {"name": "Zigmas", "year": 3, "grade": 9.13},
    {"name": "Artūras", "year": 2, "grade": 7.01},
    {"name": "Simas", "year": 3, "grade": 6.43}
  ]
}
```

Duomenis, nuskaitytus iš failo, padalinti pasirinktam GPU gijų kiekiui (duomenų kiekis turi **nesidalinti** iš gijų kiekio). Kiekvienam įrašui iš savo duomenų rinkinio GPU gija suskaičiuoja pasirinktą **tekstinio tipo** rezultatą ir jei jis tenkina pasirinktą sąlygą, įrašo į bendrą visoms gijoms rezultatų masyvą (galima nustatyti, kiek simbolių išskirti vienam rezultatų įrašui, ir jei liko neužpildytos vietos, ją užpildyti tarpais). Jei ne visi elementai atitinka filtro sąlygą, rezultatuose tarpų likti neturi — visi rezultatai turi būti rašomi į rašymo metu esančią pirmą laisvą masyvo vietą.

Programą realizuoti naudojant CUDA gijas ir CUDA atminties valdymo funkcijas.

Rezultatų failo pavyzdys (vardas pakeistas į didžiąsias raides, prijungti metai bei pagal pažymį nustatytas įvertinimas nuo A iki F; atrinkti įrašai, kurių pirma raidė abėcėlėje yra toliau, nei P):

SONATA-3A      VACYS-2B      ROBERTAS-3D      VIKTORAS-2B      VYTAS-3C      ZIGMAS-3A      SIMAS-3D

#### Testas

Testas atliekamas Moodle aplinkoje laboratorinių darbų metu. Testo klausimai sudaryti iš teorijos temų „C ir C++ atminties valdymas“, „Lygiagretusis programavimas CUDA“ ir „Thrust“. Testui skiriama 20 minučių.

---

#### L3 programų vertinimas

- L3a — 6 taškai (15 sav.)
- Kontrolinis — 4 taškai (16 sav.)

---

LD programų veikimą demonstruoti užsiėmimų laiku pagal tvarkaraštį, programų, duomenų ir rezultatų failus pateikti iki atsiskaitymo savaitės pabaigos.

---

Norint atlikti darbą reikalinga NVidia vaizdo plokštė.

## **CUDA diegimas savo kompiuteryje su NVidia GPU**

CUDA galima atsisiųsti iš NVidia puslapio. Svarbu patikrinti, kad pasirinkta CUDA versija palaikytų kompiuteryje esantį GPU.

### **Darbas su auditorijos kompiuteriais**

Auditorijos kompiuteriuose įdiegta CUDA, tačiau juose įdėtos senos vaizdo plokštės (6 serijos), nepalaikančios naujausios CUDA versijos. Dėl šios priežasties CUDA projektus reikia kurti su **Visual Studio 2012**.

### **CUDA serveris**

CUDA serveris pasiekiamas per SSH adresu `cudalab.int.ktu.lt`, prisijungimo duomenys tokie patys, kokie buvo sugeneruoti prisijungimui prie MPI serverio. Neturintys prisijungimo duomenų gali kreiptis į savo laboratorinių darbų dėstytoją. Serveryje įdiegta CUDA 9.1 versija, programos kompiliavimui žr. „CUDA kompiliavimas Linux aplinkoje“.

### **CUDA projekto kūrimas Visual Studio**

File -> New Project -> NVIDIA -> CUDA 9.x.

Programos failo plėtinys - .cu.

Programoje GPU funkcijoje norint naudoti `printf` reikia nurodyti:

Project -> Properties -> Configuration properties -> CUDA C / C++ -> Device

-> Code Generation: `compute_20,sm_20`.

### **CUDA kompiliavimas Linux aplinkoje**

Naudojamas kompiliatorius NVCC. Naudojimas analogiškas gcc, pvz:

```
nvcc program.cu -o program
```

**Pastaba:** nlhmann biblioteka nesikompiluoja su NVCC 9.