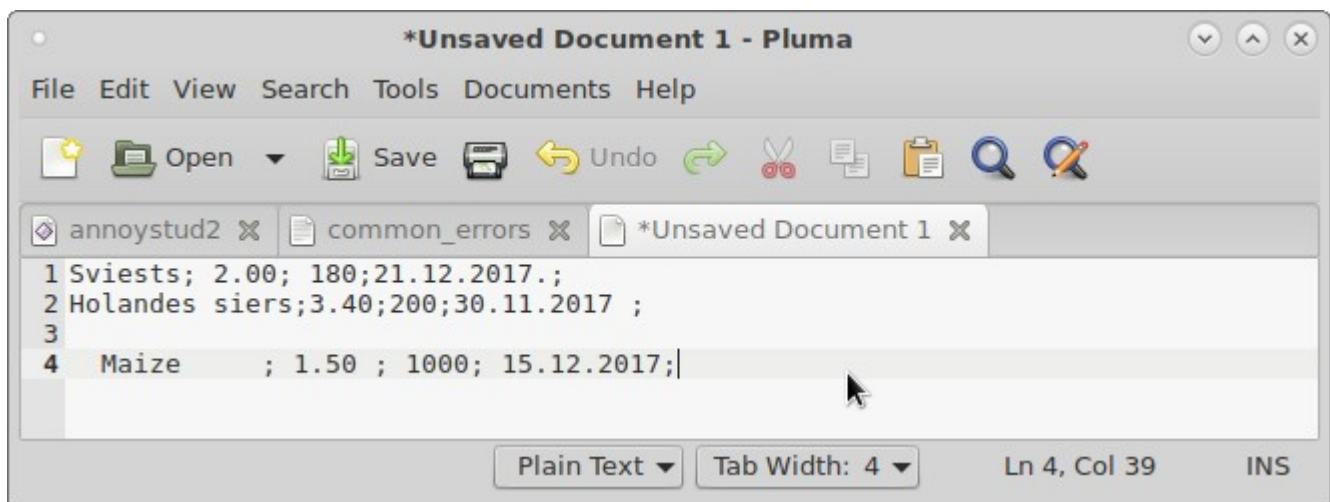


### 3. Laboratorijas darbs

#### Simbolu virknes, struktūras un failu apstrāde.

##### Uzdevums

Ar jebkura teksta redaktora palīdzību izveidot teksta failu, kas satur informāciju par struktūrām, kas aprakstītas atbilstošā variantā (failā ir jā saglabā minimums 10 struktūras, simbolu kodēšanai izmantojiet ANSI, struktūras lauku secība failā ir tieši tāda kā uzdevuma variantā). Struktūras lauki failā ir jāatdala ar semikoliem (semikolam jābūt arī aiz pēdējā elementa, taču vēlams pieņemt kā pareizu arī ievadi, kur pēdējais semikols ir izlaists), bet struktūras vienu no otras ir jāatdala ar jaunās rindas simboliem. Reālie skaitļi ir jāpieraksta ar diviem cipariem aiz komata, datumus formātā dd.mm.yyyy. (ar punktu aiz pēdējā cipara), laiku formātā hh:mm, nedēļas dienas apzīmēt ar diviem burtiem (Pr, Ot, Tr, Ce, Pt, St, Sy). Piemēram, failam ir jāizskatās sekojošā veidā:



Sastādīt programmu, kas ielasa datus no faila uz izvada tos tabulas veidā, kā arī izpilda darbības, kuras norādītas atbilstošajā variantā, izmantojot failā esošo informāciju. Ielasītie dati jā saglabā struktūru (vai klases definētā tipa) masīvā. Faila nosaukumu lietotais ievada no tastatūras. Lietotājam jāparāda saraksts ar iespējamām darbībām un jāpieprasa ievadīt darbībai atbilstošais burts vai cipars. Nederīgas ievades gadījumā pieprasījums jāatkārto.

Ievērojiet, ka fails var saturēt tukšas rindas un liekas atstarpes vai tabulācijas simbolus (pirms un pēc struktūras lauka vērtības), kuras ir jāizņem no attiecīgā struktūras elementa (bet, ja elements satur atstarpes starp vārdiem, tās ir jā saglabā). Fails var arī saturēt kļūdainu informāciju, tādējādi ir jānodrošina ievadītās informācijas pārbaude. Ja kāda no rindiņām failā satur kļūdainu informāciju (trūkst struktūras lauka vai lauku ir pārāk daudz, vai datu tips nav pareizs), tad atbilstošo rindiņu vajag ignorēt un mēģināt ievadīt datus no nākošām rindiņām. Beigās jāizvada, cik struktūras izdevās veiksmīgi nolasīt no faila, savukārt kļūdainās rindiņas jāieraksta teksta failā ar nosaukumu err.txt. Failu err.txt izvietot tajā pašā mapē, kur atrodas datu fails.

##### Papildu norādījumi:

Darbs jānodod elektroniskajā formā *ORTUS* vidē, augšupielādējot pirmkoda failu (programmai ir jā sastāv no viena .c/.cpp/.py faila). Prasības pirmkoda faila nosaukumam:

Ld3\_studentsapliecibasnumurs.c vai

Ld3\_students\_apliecibasnumurs.cpp vai  
Ld3\_students\_apliecibasnumurs.py (atbilstoši izmantotajai valodai).

Ja programma tika izstrādāta ar Microsoft® Visual Studio™ palīdzību, tad programmai ir jāpiešķir vārds Vs3\_students\_apliecibasnumurs.paplašinājums .

Lūgums ievērot šīs prasības, jo tas nepieciešams pārbaudes programmas pareizai darbībai.

Programmā paredzēt informācijas par autoru izvadi (studenta apliecības numurs, vārds, uzvārds, grupa). Nedrīkst izmantot *pragma* izteiksmes, kā arī videi specifiskos iesākumfailus (*pch.h* vai *stdafx.h*), kas nevis vienkārši jāizņem, bet jāaizstāj ar vajadzīgām bibliotēkām. *Python* nedrīkst izmantot *pip* direktīvas. Jāuzmanās ar mainīgo aprakstu ar automātisko tipa noteikšanu, kas atsevišķos gadījumos nestrādā kā paredzēts.

Ja failu ar lietotāja ievadīto nosaukumu nevar atrast, jāizvada paziņojums par kļūdu. Struktūru tabulas attēlošanai katru tās rindu izvadīt atsevišķā rindā, elementus rindas ietvaros atdalīt ar tabulāciju vai atstarpēm un (vēlams) ar citiem formatēšanas elementiem (kā vertikālā svītra | vai citi elementi, atbilstoši autora mākslinieciskajai izdomai). *Python* gadījumā jānoņem kvadrātiekvāsi vai jebkuras citas iekavas. C gadījumā jālieto struktūras, C++ – struktūras vai klases, *Python* gadījumā – klases.

Darba nodošanas termiņš – 9. decembris.

### **Uzdevumu saraksts atkarībā no varianta:**

#### **1. variants (Studenta apliecības numura pēdējais cipars 1)**

Sastādīt programmu, kas apstrādā šādu struktūru par studentiem:

- Vārds
- Uzvārds
- Dzimšanas datums
- Apliecības numurs
- Vidējā atzīme

Izpildīt:

- a. Atrast studentu, kura apliecības numurs atbilst lietotāja ievadītajam
- b. Meklēt studentu pēc uzvārda
- c. Sakārtot datus par studentiem augošā secībā pēc vidējās atzīmes

#### **2. variants (Studenta apliecības numura pēdējais cipars 2)**

Sastādīt programmu, kas apstrādā šādu informāciju par klientiem:

- Uzņēmuma nosaukums
- Reģistrācijas numurs
- Reģistrācijas datums
- Adrese
- Tālruņa numurs

Izpildīt:

- a. Meklēt klientu pēc tālruņa numura
- b. Meklēt klientu pēc nosaukuma
- c. Meklēt klientu pēc adreses

#### **3. variants (Studenta apliecības numura pēdējais cipars 3)**

Sastādīt programmu, kas apstrādā šādu informāciju par grāmatām:

- Autors

- Nosaukums
- Izdevniecība
- Izdošanas gads
- Cena

Izpildīt:

- a. Meklēt grāmatas pēc nosaukuma
- b. Meklēt grāmatas pēc autora
- c. Meklēt grāmatas pēc izdošanas gada

#### **4. variants (Studenta apliecības numura pēdējais cipars 4)**

Sastādīt programmu, kas apstrādā šādu informāciju par darbiniekiem:

- Uzvārds
- Amats
- Dzimšanas diena
- Stāžs
- Alga

Izpildīt:

- a. Noteikt darbinieku skaitu, kuru stāžs ir vienāds ar lietotāja ievadīto
- b. Atrast darbiniekus, kas ir vecāki par lietotāja norādīto vecumu
- c. Izdrukāt darbinieku sarakstu, kuru uzvārds sākas ar ievadīto burtu

#### **5. variants (Studenta apliecības numura pēdējais cipars 5)**

Sastādīt programmu, kas apstrādā šādu struktūru par cilvēkiem:

- Vārds
- Uzvārds
- Dzimšanas datums
- Dzīves vieta (pilsēta)
- Bērnu skaits

Izpildīt:

- a. Izdrukāt pensionāru anketas datus (vecāki par 62 gadiem)
- b. Atrast personu, kura uzvārds atbilst lietotāja ievadītajiem datiem
- c. Atrast personu, kuras bērnu skaits ir lielāks par lietotāja norādīto bērnu skaitu

#### **6. variants (Studenta apliecības numura pēdējais cipars 6)**

Sastādīt programmu, kas apstrādā šādu informāciju par semināru:

- Priekšmets
- Pasniedzējs
- Grupas numurs
- Nedēļas dienas
- Auditorija (vesels skaitlis)

Izpildīt:

- a. Izdrukāt datus par semināriem konkrētai grupai
- b. Izdrukāt datus par semināriem, kas notiek norādītajā dienā
- c. Izdrukāt datus par semināriem, kas notiek konkrētajā auditorijā

#### **7. variants (Studenta apliecības numura pēdējais cipars 7)**

Sastādīt programmu, kas apstrādā šādu struktūru par precēm:

- Nosaukums
- Cena

- Ražotājs
- Daudzums
- Derīguma termiņš

Izpildīt:

- Izdrukāt datus par noteiktā ražotāja precēm
- Meklēt preces pēc nosaukuma
- Meklēt preces, kuru daudzums ir mazāks nekā ievadītais skaitlis

#### **8. variants (Studenta apliecības numura pēdējais cipars 8)**

Sastādīt programmu, kas apstrādā šādu struktūru par automašīnām:

- Marka
- Izlaiduma gads
- Pārnesumkārbas (a - automāts, m - mehāniskā)
- Reģistrācijas numurs
- Nobraukums (vesels skaitlis)

Izpildīt:

- Izdrukāt datus par noteiktās markas mašīnām
- Meklēt mašīnas pēc pārnesumkārbas tipa
- Meklēt mašīnas, kas tika izlaistas pēc norādītā gada

#### **9. variants (Studenta apliecības numura pēdējais cipars 9)**

Sastādīt programmu, kas apstrādā šādu informāciju par sniegtajiem pakalpojumiem:

- Datums
- Pakalpojums
- Klienta vārds, uzvārds
- Summa

Izpildīt:

- Noteikt iegūto ieņēmumu par noteikta tipa pakalpojumu, kā arī kopējo ieņēmumu
- Meklēt veiktus pakalpojumus lietotāja ievadītājam klientam
- Meklēt klientus, kas saņēma noteikto pakalpojumu

#### **10. variants (Studenta apliecības numura pēdējais cipars 0)**

Sastādīt programmu, kas apstrādā šādu informāciju par starppilsētu autobusu reisiem:

- Atiešanas stacija
- Pienākšanas stacija
- Nedēļas diena
- Laiks
- Biļetes maksa

Izpildīt:

- Atrast visus reusus uz konkrēto pilsētu
- Atrast visus reusus, kas kursē noteiktā nedēļas dienā
- Atrast visus reusus, kuru biļetes maksa nepārsniedz ievadīto

### **Vērtēšanas kritēriji:**

Maksimālais vērtējums – 10. Var tikt piešķirti papildpunkti un var tikt noņemti punkti par neatbilstošu izpildi.

### **Papildpunkti tiek piešķirti par:**

Par konstanto rādītāju/adrešu argumentu izmantošanu funkcijām (C++; skat. zemāk)	+1
!Par konstanto argumentu izmantošanu funkcijām, ja tie izmantoti arī parastajiem argumentiem ( <i>cargo cult programming</i> ; C++)	+½
Par ievades smalkāku (nekā prasīts) pārbaudi:	
ar regulārajām izteiksmēm	+1
ar <i>stringstream</i>	+1
ar iterāciju pa masīva elementiem	+1
pārbaudot arī, ka ir tikai viens punkts un mīnusa zīme tikai sākumā	+1
Par Windows termināļa pielāgošanu Unikoda simboliem	+1
<b>Par noslēdzošā semikola struktūrām un beigu punkta (datumos) neobligātumu</b>	+1
<b>Par iespēju meklēt ne tikai pēc pilnā teksta, bet arī pēc teksta daļas</b>	+1
<b>Par iespēju meklēt ar regulārajām izteiksmēm (<i>wildcards</i>)</b>	+1
Par anonīmo funkciju izmantošanu	+1
Par konteineru objektu izmantošanu (C++)	+1
Par kortežu izmantošanu konstanto argumentu masīviem ( <i>Python</i> ; atskaitot citu autoru sarakstīto bibliotēku funkciju argumentus)	+1
Par <i>try/catch</i> izmantošanu C++	+1
Par <i>map</i> izmantošanu <i>Python</i>	+1
Par īpaši skaistu programmas saskarni	+½
Par īpaši skaistu tabulas izvadi, t.sk. elementu izlīdzināšanu ar dažādu atstarpju skaitu	+½
Par <i>errno</i> izmantošanu	+½
Par <i>fenv</i> izmantošanu	+½
Par citām labām lietām	+1

### **Punkti tiek noņemti par šādiem pārkāpumiem:**

Nav ietverta nepieciešamā bibliotēka	-½
Ietverts vairāk par 3 nevajadzīgām bibliotēkām	-½
Funkcijas primitīvs nav norādīts pirms <i>main()</i> funkcijas	-½
Problēmas ar vārdtelpu ( <i>namespace</i> ) lietošanu	-½
Mainīgais vai funkcija tiek deklarēti vairākas reizes	-½
Mainīgais vai funkcija tiek deklarēti vairākas reizes un ar dažādiem tiem (C/C++)	-½
<i>main()</i> funkcijai ir tips <i>void</i>	-½
Netiek izvadīta informācija par programmas autoru	-½
Programmas saskarne ir pavisai uzrakstīta un/vai izvade necaurskatāma	-½
Programmā tiek vienlaicīgi izmantoti rīki darbam ar <i>cin</i> un <i>stdin</i> vai ar <i>cout</i> un <i>stdout</i>	-½
Pirms matemātisko darbību turpināšanas starprezultātam izmantota noapaļošanas funkcija, nevis formatēšanas funkcija ( <i>Python</i> )	-¾
Skaitļi netiek noapaļoti ar diviem cipariem aiz komata	-¾
Tabulas izvade ir vizuāli nepievilcīga	-¾
Programmā tiek vienlaicīgi izmantoti rīki darbam ar <i>cin</i> un <i>stdin</i> vai ar <i>cout</i> un <i>stdout</i> un tas noved pie problēmām	-1
Tabula ar ielasītajiem ierakstiem netiek attēlota	-1
Robojuma ( <i>indentation</i> ) kļūda ( <i>Python</i> )	-1
Rekursijas izmantošana, ja tas noved pie kaut kādām problēmām	-1
<b>Tiek izdzēstas visas atstarpes elementa ietvaros (t.i., ne elementa sākumā/beigās)</b>	-1
<b>Ne visas kļūdainās rindiņas no datu faila tiek pārceltas uz <i>err.txt</i></b>	-1
Netiek iztīrīts buferis pēc nederīgās ievades	-1
Tiek izmantoti mainīgā garuma masīvi C	-1
Netiek strādāts ar dinamiski izdalāmo atmiņu ( <i>heap</i> )	-1

Netiek izdzēsti objekti dinamiski izdalāmajā atmiņā, vai arī to dara nepareizi	-1
Mēģināts izmantot <i>string</i> objektu funkcijas <i>char</i> tipa masīviem	-1
<b>Netiek pārbaudīts, vai fails <i>err.txt</i> ir atvērts*</b>	-2
<b>Netiek pārbaudīts, vai fails ir atvērts (neattiecas uz <i>err.txt</i>)</b>	-2
<b>Failu atver nepareizā režīmā</b>	-2
<b>Netiek nodzēstas atstarpes un tabulācija struktūras lauku sākumā un/vai beigās</b>	-2
<b>Kad ir jāattēlo vairāki ieraksti meklēšanas rezultātos, attēlo tikai vienu</b>	-2
<b>Pēc darba beigām fails(-i) netiek aizvērts(-i)</b>	-3
<b>Netiek izveidots fails <i>err.txt</i></b>	-3
<b>Programma nelasa failu (sīkas kļūmes dēļ)</b>	-3
<b>Struktūru vietā izmanto vairākus masīvus</b>	-3
Segmentēšanas kļūda ( <i>segmentation fault</i> )	-3
Izpildlaika kļūda, atbrīvojot atmiņu (t.sk. atmiņas noplūde)	-3
<b>Datu faila nosaukums ir iekodēts programmas tekstā kā literālis</b>	-4
Drukas kļūdas, kas neļauj kompilēt programmu (acīmredzot tas nekad netika darīts)	-5
<b>Programma nelasa failu (fundamentāla kļūda kodā)**</b>	-8
Citas kļūdas, kas nepieļauj programmai strādāt (par katru)	-1

\* iespējams, ka darbs tiek palaists uz failu sistēmas, kas piemontēta tikai lasīšanas režīmā

\*\* uz to neattiecas tas, vai pēc autora idejas failam obligāti jābūt ar paplašinājumu (taču labāk paredzēt ievadi no faila ar jebkuru paplašinājumu).