Programavimas C++

III-IV klasė

Simbolių eilučių masyvai

Mažeikių Gabijos gimnazija

Parengė informatikos mokytoja Lina Katarskienė

char **simbolio** tipas, tai duomuo, kuris gali būti bet kas - ***skaičius, raidė, ženklas***.

Programai jį galima pateikti iš klaviatūros arba iš tekstinio failo.

Nuskaitymo metodai:

- Operatoriaus **>>** pagalba: ***įveda vieną simbolį, praleidžia tarpus ir eilučių pabaigos simbolius*** `fd>>simb;`
- Naudojant `simb = fd.get();` *arba* `fd.get(simb);` **įveda vieną simbolį, nepraleidžia tarpų ir eilučių pabaigos simbolių.**

Simbolis yra per mažas teksto elementas. Todėl vartojami didesni – **simbolių eilutės** - **`string`**.
Tuomet būtina įterpti `#include <string>`

Nuskaitymo metodai:

- ***Operatorius*** `>>` įveda į `string` tipo eilutę ***iki tarpo*** *arba* ***iki eilutės pabaigos simbolio, jei nėra tarpų***. `fd>>eil;`
- Metodas `getline(fd, eil);` įveda į `string` tipo eilutę ***iki eilutės pabaigos*** simbolio.

getline skaito visą eilutę iki naujos eilutės (*'\n'*) arba kito nurodyto skyriklio. Jis naudojamas norint **nuskaityti visą tekstą, įskaitant tarpelius**, ir yra naudingas, kai reikia gauti vartotojo įvestį kaip visą sakinį ar frazę. *getline* pašalina skyriklį (*'\n'*) iš srauto.

- Norint įvesti ***dalį simbolių eilutės*** `getline(fd, eil, '+');` nurodžius simbolį bus nuskaitoma nebe iki naujos eilutės, bet **iki nurodyto simbolio**. (pvz. iki +)
- Metodas `fd.get(eil, n);` Skaito iki **n simbolių** ir prideda '\n' (pabaigos simbolį)

Jei norime nuskaityti eilutę ir talpinti masyve galime naudoti

```
fd.get(tekst, 10);
par[i] = tekst; //
```

nuskaitoma 10 simbolių ilgio eilutė

arba

```
getline(pr, eil);
miest[i] = eil;
```

jei skaitoma eilutė iki pabaigos

- ***Eilutės ilgio nustatymas*** `n=eil.length()` nustato eilutės simbolių kiekį

- ***Dalies eilutės*** simbolių ***kopijavimas*** `ei=eil.substr(k,n)` iš eilutės kopijuoja n simbolių, pradedant k-uoju
- `fd.ignore()` naudojama **ignoruoti** arba išvalyti vieną ar daugiau (n) simbolių iš įvesties buferio

  fd.ignore(n); n- nurodo, *kiek simbolių reikia nepaisyti* .

  fd.ignore\n); \n- nurodo, nurodo skiriamąjį *simbolį, iki kurio įvestis bus nepaisoma*.
- `fd>>ws;` praleidimo manipuliatorius- **nepaisoma tarpo**, kol neatsiras koks nors ne tarpo simbolis

Pavyzdys_1:

Miestai. Tekstinėje byloje miestai.txt nurodytas miestų pavadinimų skaičius n o kitose eilutės surašomi miestų pavadinimai. Miestų sąrašas nesutvarkytas. Sudarykite programą, kuri pateiktų miestų sąrašą surikiuotą abecėlės tvarka. Rezultate pateikite pradinį ir sutvarkytą miestų sąrašą byloje sarasas.txt. Panaudokite miestamas eilučių masyvą ir duomenų skaitymui paprogramę Void.

```
1   //miestai
2   #include <iostream>
3   #include <fstream>
4   #include <string>
5   using namespace std;
6   void skaitymas();
7   int i, n;
8   string miest[100];
9   string eil;
10  int main()
11  {
12      setlocale(LC_CTYPE, "");
13      skaitymas();
14      ofstream rz ("sarasas.txt");
15      char raid;
16      rz <<"Miestu sarasas " << endl <<endl;
17        for (i = 0; i < n; i++)
18          rz <<miest[i] <<endl;
19      rz <<endl;
20      rz <<"Miestai abeceles tvarka " << endl <<endl;
21        for (raid = 'A'; raid <= 'Z'; raid++)
22        {
23           for (i = 0; i < n; i++)
24           {
25              eil = miest[i];
26                if (eil[0] == raid)
27                rz <<miest[i] <<endl;
28           }
29        }
30     rz.close();
31  }
32  void skaitymas()
33  {
34      ifstream pr ("miestai.txt");
35      pr >>n;
36      pr.ignore(); //perkelia zymekli i kita eilute
37        for (i = 0; i < n; i++)
38        {
39            getline(pr, eil);
40            miest[i] = eil;
41        }
42      pr.close();
43  }
```

Rikiavimas Burbulo metodu,lyginant kiekvieną elementą su visais po jo esančiais, o sukeitimui naudojamas laikinas kintamasis:

```
// Rikiuojame abėcėlės tvarka
for (int i = 0; i < n - 1; i++)
{
    for (int j = i + 1; j < n; j++)
    {
        if (miestai[i] > miestai[j])
        {
            string laikinas = miestai[i];
            miestai[i] = miestai[j];
            miestai[j] = laikinas;
        }
    }
}
```

Jei naudosime vektorių, programa atrodys taip:

```
// Paprogramės prototipas
void Skaityti(vector<string> &miestai, int &n);

int main()
{
    vector<string> miestai;
    int n;
//kreipinys duomenų skaitymui
    Skaityti(miestai, n);

    ofstream fr("sarasas.txt");
    fr << "Pradinis miestų sarašas:" << endl;
    for (int i = 0; i < n; i++)
        fr << miestai[i] << endl;
//vektoriaus rikiavimas
    sort(miestai.begin(), miestai.end());
//spausdiname surikiuotą sąrašą
    fr << endl;
    fr << "Surikiuotas miestų sarašas:" << endl;
    for (int i = 0; i < n; i++)
        fr << miestai[i] << endl;

    fr.close();
    return 0;
}
```

```
// Duomenų skaitymo paprogramė
void Skaityti(vector<string> &miestai, int &n)
{
    ifstream fd("miestai.txt");
    fd >> n;
    fd.ignore();
    string miestas;
    for (int i = 0; i < n; i++)
    {
        getline(fd, miestas);
        miestai.push_back(miestas);
    }
    fd.close();
}
```

```
Pradinis miestų sąrašas:
Vilnius
Kaunas
Alytus
Šiauliai
Klaipėda
```

```
Sutvarkytas miestų sąrašas:
Alytus
Kaunas
Klaipėda
Šiauliai
Vilnius
```

Pavyzdys_2:

Zodziai. Tekstinės bylos eilutes.txt pirmoje eilutėje nurodytas eilučių skaičius tekste, o sekančiose - pats tekstas. Sudarykite programą, kuri suskaičiuotų, kiek kiekvienoje eilutėje yra žodžių ir kiek žodžių iš viso. Atsakymą pateikite byloje rezultat.txt. Duomenų skaitymui naudokite paprogramę.

```cpp
//simboliniu eiluciu masyvas
#include <iostream>
#include <fstream>
using namespace std;
void skaitymas();//proceduros pasirinkimas programoje
int i, n; //kintamuju pasirinkimas kad veiktu visur ir proceduroje ir programoje
string eil;
string eilt[100];
int main()
{
    setlocale(LC_CTYPE, "");
    ofstream rz("rezultat.txt"); //nurodoma duomenu isvedimo byla
    int kiek, viso, j, m; // kintamieji kursai veikia tik programoje
    string simb;
    skaitymas();// kreipimasis i procedura
    viso = 0;
      for (i = 0; i < n; i++)
      {
          kiek = 0;
          simb = eilt[i];
          m = simb.length(); //nuskaitomas eilutes ilgis
            for (j = 1; j <= m; j++) //imami visi simboliai is eilutes
            {

                  if (simb[j] == ' ') kiek = kiek + 1;
            }
          viso = viso + kiek;
          rz <<i + 1<<" eilutėje žodžių :" <<kiek <<endl;
      }
    rz <<" Viso žodžių tekste :" <<viso;
}
//proceduros skaitymui aprasymas:
void skaitymas()
{
    ifstream pr("eilutes.txt");
    pr >>n; //nuskaitomas eilučių skaičius
    pr.ignore(); //persoka i kita eilute byloje
      for (i = 0; i < n; i++)
      {
          getline(pr, eil); //nuskaitomos eilutės į masyva
          eilt[i] = eil;
      }
}
```

```
3
Siandien yra grazi diena
Mokomes programuoti C++
Cia yra trecia eilute
```

```
1 eiluteje yra 4 zodziu.
2 eiluteje yra 3 zodziu.
3 eiluteje yra 4 zodziu.
Is viso zodziu: 11
```

Žodžių paieška galima atlikti ir kitaip - naudoti funkciją **`stringstream`**

`stringstream ss(eilute);` sukuria objektą ss, iš kurio galime skaityti žodžius po vieną, panašiai kaip skaitome duomenis iš failo. Svarbu: `stringstream` pats žodžių neskaičiuoja. Jis tik leidžia juos skaityti po vieną iš jau perskaitytos eilutės.

Tuomet `ss >> zodis` perskaito tik vieną žodį, nes operatorius `>>`, skaitydamas tekstą į `string` kintamąjį, sustoja ties tarpu. Kitą kartą jis perskaito kitą žodį. Tarpai praleidžiami.

Įsiminkite tris svarbiausius veiksmus:

1. `stringstream ss(eilutes[i]);` – pasiruošiame skaityti vieną teksto eilutę žodžiais.
2. `while (ss >> zodis)` ir `kiek++;` – skaitome žodžius po vieną ir skaičiuojame.
3. `viso += kiek;` – pridedame tos eilutės žodžius prie bendro skaičiaus.

Trumpai: `getline()` perskaito visą eilutę, o `stringstream` leidžia tą eilutę perskaityti po vieną žodį.

`stringstream` C++ kalboje naudojamas tada, kai norime skaityti arba išskaidyti tekstą į atskiras dalis (žodžius).

Tam reikalinga biblioteka: **`#include <sstream>`**

Naudojant vektorių:

```cpp
// Paprogramės prototipas
void Skaityti(vector<string> &eilutes, int &n);

int main()
{
    vector<string> eilutes;
    int n;
    Skaityti(eilutes, n);
    ofstream fr("rezultat.txt");
    int viso = 0;
    for (int i = 0; i < n; i++)
    {
        stringstream ss(eilutes[i]);
        string zodis;
        int kiek = 0;
        while (ss >> zodis)
            kiek++;
        fr << i + 1 << " eiluteje yra "
           << kiek << " zodziu." << endl;
        viso += kiek;
    }
    fr << "Is viso zodziu: " << viso << endl;
    fr.close();
    return 0;
}
```

```cpp

// Duomenu skaitymo paprogramė
void Skaityti(vector<string> &eilutes, int &n)
{
    ifstream fd("eilutes.txt");
    fd >> n;
    fd.ignore();
    string eilute;
    for (int i = 0; i < n; i++)
    {
        getline(fd, eilute);
        eilutes.push_back(eilute);
    }
    fd.close();
}
```

`stringstream` naudojimo atvejai

1. Kai reikia suskaičiuoti žodžius eilutėje
2. Kai iš vienos eilutės reikia išskirti skaičius
3. Kai vienoje eilutėje pateikti keli duomenys
4. Kai reikia konvertuoti tekstą į skaičių

```cpp
string eilute = "Labas rytas visiems";
stringstream ss(eilute);
string zodis;
int kiek = 0;

while (ss >> zodis)
{
    kiek++;
}
```

Rezultatas: `kiek = 3`.

```cpp
string eilute = "12 25 37";
stringstream ss(eilute);
int x;

while (ss >> x)
{
    cout << x << endl;
}
```

Eilutėje esantys skaičiai perskaitomi po vieną: `12`, `25`, `37`.

```cpp
string eilute = "Jonas 16 9";
stringstream ss(eilute);

string vardas;
int amzius, pazymys;

ss >> vardas >> amzius >> pazymys;
```

Gauname vardą `Jonas`, amžių `16` ir pažymį `9`.

```cpp
string tekstas = "125";
stringstream ss(tekstas);

int skaicius;
ss >> skaicius;
```

Gauname skaičių `125`, kurį jau galime naudoti skaičiavimuose.

Kaupiamasis

Uždavinių sprendimas įgūdžiams formuoti

Lankytinų vietų sąrašas. Sudarytas Lietuvõs lankytinų vietų sąrašas. Kiekvienam vietovardžiui faile skirta viena eilutė. Parašykite programą, kuri surikiuotų vietovardžius abėcėliškai.

Pirmoje failo eilutėje įrašytas lankytinų vietų skaičius n ($1 \le n \le 300$). Toliau kiekvienoje eilutėje nurodytas lankytinos vietos pavadinimas.

Pradinių duomenų ir rezultatų failų pavyzdys

Pradiniai duomenys	Rezultatai
5 Kuršių nerija Trakai Novaraisčio ornitologinis draustinis Raigardo slėnis Grūto parkas	---------- Pradiniai duomenys ---------- Kuršių nerija Trakai Novaraisčio ornitologinis draustinis Raigardo slėnis Grūto parkas ---------- Surikiuoti duomenys ---------- Grūto parkas Kuršių nerija Novaraisčio ornitologinis draustinis Raigardo slėnis Trakai

1.

Miestai

Parenkite programą, kuri iš miestų sąrašo išrinktų pavadinimus, kuriuos sudaro daugiau kaip vienas žodis. Kiekvienam miesto pavadinimui faile skirta viena eilutė. Žodžiai pavadinime atskirti vienu tarpo simboliu. Prieš miesto pavadinimą ir po jo tarpo simbolių nėra.

Pradiniai duomenys	Rezultatai
5 Druskininkai Kazlų Rūda Pasvalys Kuršėnai Naujoji Akmenė	---------- Pradiniai duomenys ---------- Druskininkai Kazlų Rūda Pasvalys Kuršėnai Naujoji Akmenė ---------- Išrinkti miestai ---------- Kazlų Rūda Naujoji Akmenė

2.

3.

Populiariausia raidė

Parenkite programą, kuri apskaičiuotų, kiek kokių raidžių yra tekste. Tekstą sudaro *n* eilučių. Didžiąsias ir mažąsias raides skaičiuokite kartu. Kuri raidė dažniausiai pasikartoja nurodytame tekste?

Pradiniai duomenys	Rezultatai
2 Šiandien šviečia saulė. Rytoj tikriausiai lis lietus.	Aa 5 Ąą 0 Bb 0 Cc 0 Čč 1 Dd 1 Ee 3 Ęę 0 Ėė 1 Ff 0 Gg 0 Hh 0 Ii 10 Įį 0 Yy 1 Jj 1 Kk 1 Ll 3 Mm 0 Nn 2 Oo 1 Pp 0 Rr 2 Ss 4 Šš 2 Tt 3 Uu 3 Ųų 0 Ūū 0 Vv 1 Zz 0 Žž 0 Dažniausiai kartojasi raidė Ii.

4.

Referendumas

Zuikių partija nutarė surengti referendumą dėl lapių ir vilkų teisių apribojimo. Tam reikia surinkti *n* miško gyventojų parašų. Pasibaigus parašų rinkimo terminui, visi duomenys buvo perkelti į failą. Pirmoje failo eilutėje įrašytas pasirašiusiųjų skaičius *n*. Tolesnėse eilutėse nurodyti kiekvieno pasirašiusiojo duomenys: vardas, pavardė, gyvenamoji vieta. Jie skiriami vienu tarpo simboliu. Parenkite programą, kuri patikrintų, ar pakanka parašų referendumui surengti. Tam reikia suformuoti sąrašą, kuriame nebūtų pasikartojančių (vienodų) eilučių, ir patikrinti, ar naujajame sąraše yra ne mažiau kaip *n* eilučių.

Pradiniai duomenys	Rezultatai
7 Baikštutis Ilgaausis Laukymė3 Drąsutis Ilgaausis Pamiškė2 Greitutis Trumpauodegis Laukymė1 Baikštutis Trumpauodegis Laukymė2 Greitutis Ilgaausis Pamiškė1 Drąsutis Trumpauodegis Pamiškė3 Greitutis Trumpauodegis Laukymė1	--- Sąrašas --- Baikštutis Ilgaausis Laukymė3 Drąsutis Ilgaausis Pamiškė2 Greitutis Trumpauodegis Laukymė1 Baikštutis Trumpauodegis Laukymė2 Greitutis Ilgaausis Pamiškė1 Drąsutis Trumpauodegis Pamiškė3 --- Išvada --- Referendumui parašų neužtenka.

5.

Mokinių statistika

Parenkite programą, kuri apskaičiuotų, kiek mokykloje yra mergaičių, jeigu žinoma, kad berniukų vardai baigiasi raide *s*. Pirmoje failo eilutėje nurodytas mokinių skaičius. Tolesnėse eilutėse pateikiamas mokinių sąrašas. Kiekvienoje failo eilutėje įrašyti vieno mokinio pavardė ir vardas, atskirti bent vienu tarpo simboliu.

Pradiniai duomenys	Rezultatai
7 Petraitis Rokas Augė Artūras Mikalauskaitė Aušra Slivka Donatas Stakėnaitė Ieva Skrebė Domas Bruzgaitė Akvilė	3 Mikalauskaitė Aušra Stakėnaitė Ieva Bruzgaitė Akvilė

Kodas

Viename pradinių duomenų faile įrašytas sakinys, kurio pabaigoje yra taškas, o kitame – raidžių kodai. Sakinį sudaro tik žodžiai, atskirti tarpo simboliu. Parenkite programą, kuri užkoduotų ir išspausdintų atskirose eilutėse kiekvieną sakinio žodį.

Pirmoje antrojo pradinių duomenų failo eilutėje įrašytas koduojamų raidžių skaičius. Kitose eilutėse abėcėliškai surašytos visos sakinyje esančios raidės ir nurodyti jų kodai, atskirti vienu tarpo simboliu. Kodo ilgis gali būti nuo 1 iki 4 simbolių.

Atskirose rezultatų failo eilutėse turi būti išspausdinti užkoduoti pradinio teksto žodžiai. Kodai vienas nuo kito atskiriami tarpo simboliu.

Pradiniai duomenys	Rezultatai
SVEIKI IR SUDIE.	*** *** _ * ** _ * _ **
8 D _ ** E * I ** K _ * _ R * _ * S *** U ** _ V *** _	** * * _ *** ** _ _ ** ** *

6.

Simbolinių reikšmių įvestis ir išvestis (1 metodas)

```
const int skirt = 32;
                // skirtumas tarp mažųjų ir didžiųjų raidžių
void TForm1::Konvertuoti()
{
 ifstream fd(CDuom);
 int ch;
 ofstream fr(CRez);
 while((ch = fd.get()) != EOF) {
  if(ch >= 'a' && ch <= 'z')    //  if(ch >= 97 && ch <= 122)
   ch -= skirt;
  fr << (char)ch;
 }
 fd.close();
 fr.close();
}
```

Raidžių konversija (2 metodas)

```
void TForm1::Konvertuoti()
{
 ifstream fd(CDuom);
 int ch;
 ofstream fr(CRez);
 while((ch = fd.get()) != EOF)
  fr << (char)tolower(ch);
 fr << endl;
 fd.close();            // uždaro failą
 fd.open(CDuom);  // nustato į failo pradžią
 while((ch = fd.get()) != EOF)
  fr << (char)toupper(ch);
 fd.close();
 fr.close();
}
```

Raidžių skaičiavimas - metodas

```
char raides[256];  // raidžių dažniai.  Aprašome TForm1 klasėje
bool TForm1::RaidziuDazniai()
{
 ifstream fd(CDuom);
 char ch;            // įvedamas simbolis
 ofstream fr(CRez);
 for (int i = 0; i < 256; i++)
  raides[i] = 0;     // išvalome dažnių masyvą
 while(fd.get(ch))
  if(isalpha(ch))
   raides[ch] += 1;
 fd.close();
 fr << "Tekste yra tokios raidės su tokiais dažniais\n";
 for (int i = 0; i < 256; i++)  // rezultatų spausdinimas
  if(raides[i] > 0)
   fr << (char)i << " " << (int)raides[i] << endl;
 fr.close();
}
```

Funkcijos su simboliais

Bibliotekoje ctype, jei yra iostream, nereikia	
int tolower(char c)	konvertuoja raidę į mažąją
int topupper(char c)	konvertuoja raidę į didžiąją
int isalpha(char c)	1, jei raidė
int isalnum(char c)	1, jei raidė arba skaičius
int iscntrl(char c)	1, jei valdantis simbolis (kodai 0 - 037, 0177)
int isspace(char c)	1, jei tarpas, tabuliacija, nauja eilutė