

Навчальна практика з програмування

Мета навчальної практики з програмування: поглиблення і закріплення здобутих за час навчання теоретичних і практичних знань, умінь та навичків з навчальної дисципліни "Алгоритмічних мови та програмування".

Під час практики студенти під керівництвом викладача закріплюють набуті навички програмування та опрацьовують різноманітні теми з програмування.

Орієнтовний регламент навчального дня практики:

- обговорення з керівником практики поточної теми;
- індивідуальне опрацювання комплексних задач;
- документальне оформлення ходу розв'язання найскладнішої проблеми.

Під час проходження практики студенти щодня документально оформлюють (звіт) виконання найскладніших та/або найцікавіших фрагментів отриманих практичних завдань з програмування. При цьому звіт має містити опис завдання, деталізацію інформації стосовно реалізації програми, опис та обґрунтування використання змінних, опис формалізації даної задачі та кроки виконання програми.

Під час проходження студентами навчальної практики керівник практики контролює роботу студентів, виконання поставлених завдань та їх якість, проводить консультації зі студентами під час виконання завдань з метою перевірки роботи та надання допомоги.

Перелік тем, які опрацьовуються під час практики:

- Тип struct. Функції. Робота з файлами.
- Теорема суми / добутку елементів масиву, теорема вирішення, сортування вибором, алгоритм підрахунку, знаходження максимального і мінімального елементу масиву.
- Типові задачі на вибір, методи їх розв'язування.
- Типові задачі на вибір, нові методи їх розв'язування.
- Задачі на багаторівневі впорядкування.
- Методи програмування задач на об'єднання впорядкованих множин.
- Метод простого впорядкування.
- Знаходження мінімуму та максимуму серед елементів із даною властивістю.
- Пошук k-того найбільшого та найменшого елементів.

Приклад комплексної задачі, яка опрацьовується студентами під час практики:

Завдання «Вимірювання»

Є файл *meresek.txt*, який містить дані вимірювань експерименту. У файлі може бути максимум 300 вимірювань. Кожен рядок містить одне вимірювання з його ідентифікатором, типом вимірювання (від 0 до 4), значеннями F і Z вимірювання.

Наприклад:

```
R1458 1 24.76 7.25
R1459 1 21.77 10.78
R1460 2 49.21 24.14
R1461 0 11.58 5.74
R1462 2 54.64 27.21
```

У першому рядку вищезазначеного прикладу знаходиться вимірювання з ідентифікатором R1458, типом 1. Значення F цього вимірювання становить 24.76, а значення Z - 7.25.

Напишіть програму, яка, використовуючи дані файлу *meresek.txt*, відповідає на наступні питання! Збережіть вихідний код програми під назвою *meresek*! Перед виведенням на екран результатів завдань, що вимагають виведення на екран, виведіть номер завдання (наприклад: **Завдання 3**). Якщо ви запитуєте дані у користувача, покажіть на екрані, яке значення ви очікуєте.

1. Прочитайте дані з файлу *meresek.txt*. Виведіть, скільки вимірювань містить файл!
2. Визначте, скільки вимірювань було проведено з кожного типу!
3. Визначте, чи вимірювання з найбільшими значеннями F та Z були проведені в одному й тому ж вимірюванні!
4. Визначте, скільки значень F перевищили середнє!
5. Відомо, що будь-яке значення F вимірювання не може перевищувати 100. Створіть файли, в яких збереже вимірювання з діапазонів 0-25, 25-50, 50-75 та 75-100! Назви файлів мають бути діапазонами вимірювань. Перший рядок файлу містить кількість вимірювань, що в ньому знаходяться.
6. Значенню вимірювання, яке часто вимірюється у певному типі, приділяється важлива роль. Визначте серед значень Z для кожного типу, які з них є такими значеннями!

Programni kodi iz wykonanih zvitov robot studentov:

```
#include <iostream>
#include <fstream>
using namespace std;
int loertek(string lanc)
{
    int pont=lanc.length()*20;
    int p1=0,minusz=0;
    while(p1<lanc.length())
    {
        if(lanc[p1]!='+') pont-=minusz;
        else
        {
            pont-=20;
            minusz++;
        }
        p1++;
    }
    return pont;
}
int main()
{
    ifstream f_versemy("VERSENY.TXT");
    int p1;
    f_versemy>>p1;
    string versemy[p1];
    int p2=0;
    while(p2<p1)
    {
        f_versemy>>versemy[p2];
        p2++;
    }
    f_versemy.close();
    cout<<"\t2.feladat\nsortalatok: ";
    p2=0;
    while(p2<p1)
    {
        int p3=1;
        while(p3<versemy[p2].length())
        {
            if(versemy[p2][p3]!='+'&&versemy[p2][p3-1]!='+')
            {
                cout<<p2+1<<" ";
                break;
            }
            p3++;
        }
        p2++;
    }
    int maxi=0;
    p2=0;
    while(p2<p1)
    {
        if(versemy[p2].length()>versemy[maxi].length()) maxi=p2;
        p2++;
    }
    cout<<"\n\t3.feladat\nlegtobb lovest leado: "<<maxi+1<<endl;
    cout<<"\n\t5.feladat\nVERSENYZO SORSZAMA: ";
    int valasz1;
    cin>>valasz1;
    cout<<"a: celt ero lovesek: ";
    p2=0;
    int p3=0,p4=1,p5=1;
    maxi=0;

```

```
while(p2<versemy[valasz1-1].length())
{
    if(versemy[valasz1-1][p2]!='+')
    {
        cout<<p2+1<<" ";
        p3++;
    }
    p2++;
    if(versemy[valasz1-1][p4]!='+'&&versemy[valasz1-1][p4-1]!='+') p5++;
    else
    {
        if(p5>maxi) maxi=p5;
        p5=1;
    }
    p4++;
}
if(p5>maxi) maxi=p5;
cout<<"\n\t: eltalalt korongok szama: "<<p3<<"\n\t: leghosszabb sorlatolat: "<<maxi<<"\n\t: elert pontszam: "<<loertek(versemy[valasz1-1]);
int pontszam[2][p1];
p2=0;
while(p2<p1)
{
    pontszam[0][p2]=loertek(versemy[p2]);
    pontszam[1][p2]=p2+1;
    p2++;
}
for(int i=0;i<p1;i++)
{
    maxi=i;
    for(int j=i;j<p1;j++)
    {
        if(pontszam[0][maxi]<pontszam[0][j]) maxi=j;
    }
    int csere1=pontszam[0][i];
    pontszam[0][i]=pontszam[0][maxi];
    pontszam[0][maxi]=csere1;
    int csere2=pontszam[1][i];
    pontszam[1][i]=pontszam[1][maxi];
    pontszam[1][maxi]=csere2;
}
ofstream f_sorrend("SORREND.TXT");
p2=0;
int hely=1;
while(p2<p1)
{
    int pont=pontszam[0][p2];
    while(pont==pontszam[0][p2])
    {
        f_sorrend<<hely<<" "<<pontszam[1][p2]<<" "<<pontszam[0][p2]<<endl;
        p2++;
    }
    hely++;
}
f_sorrend.close();
return 0;

```

```
char szinek[26]='A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z';
if(sajat==0)
{
    cout<<"szin amelyre atfetheto a kerites: ";
    kovetkezo=1;
    for(int j=sajat+1;j<p;j++)
    {
        if(s_adat[j].oldal==s_adat[sajat].oldal)
        {
            kovetkezo=j;
            break;
        }
    }
    elozo--1;
    for(int j=sajat-1;j>0;j--)
    {
        if(s_adat[j].oldal==s_adat[sajat].oldal)
        {
            elozo=j;
            break;
        }
    }
    for(int i=0;i<26;i++)
    {
        if(s_adat[sajat].kerites==szinek[i]) continue;
        else if(elozo!=-1&&s_adat[elozo].kerites==szinek[i]) continue;
        else if(kovetkezo!=-1&&s_adat[kovetkezo].kerites==szinek[i]) continue;
        cout<<szinek[i]<<" ";
        break;
    }
}
cout<<"\nuj szin: ";
char valasz2;
cin>>valasz2;
s_adat[valasz1].kerites=valasz2;
}

```

```
int seged = 0;
for(int i = 0; i <= db; i++){
    if(seged < v[i].pontszam)
        seged = v[i].pontszam;
}

int dij = 1;
for(int i = 0; i <= db; i++){
    if(v[i].pontszam == seged)
        cout << dij << ".dij(" << v[i].pontszam << "pont):" << v[i].id << endl;
}

dij++;
while(dij < 4){
    int legnagyobb = 0;
    for(int i = 0; i <= db; i++){
        if(maxi < v[i].pontszam && v[i].pontszam < seged)
            legnagyobb = v[i].pontszam;
    }
    seged = legnagyobb;
    for(int i = 0; i <= db; i++){
        if(v[i].pontszam == seged)
            cout << dij << ".dij(" << v[i].pontszam << "pont):" << v[i].id << endl;
    }
    dij++;
}

```

```
string letrak[otos]={};
int indk=0,db[otos]={};
for(int i=0; i<otos; i++){
    bool van = false;
    string kozep = othossz[i].substr(1,3);
    for(int j=0; j<indk; j++){
        if(kozep==letrak[j]){
            van=true;
            db[j]++;
        }
    }
    if(!van){
        letrak[indk]=kozep;
        db[indk]++;
        indk++;
    }
}
for(int i=0; i<indk; i++){
    if(db[i]>1){
        for(int j=0; j<otos; j++){
            string kozep = othossz[j].substr(1,3);
            if(kozep==letrak[i]){
                letra << othossz[j] << '\n';
            }
        }
        letra << '\n';
    }
}

```

```

#include <iostream>
#include <fstream>
#include <ctime>
#include <cstdlib>

using namespace std;
struct kerdesek
{
    string kerdes, temakor;
    int helyes, pont;
    bool elso=true;
} k[100];

int main()
{
    srand(time(0));
    ifstream f("felszam.txt");
    int db=0;
    while(!f.eof())
    {
        getline(f, k[db].kerdes);
        f>>k[db].helyes>>k[db].pont>>k[db].temakor;
        f.get();
        db++;
    }
    cout<<"2. feladat\n"<<db<<" feladat van az adatfajlban.\n";
    int matek=0, egy=0, ketto=0, harom=0;
    int kezd=k[0].helyes, veg=0;
    for(int i=0; i<db; i++)
    {
        if(k[i].temakor=="matematika")
        {
            matek++;
            if(k[i].pont==1)
                egy++;
            else if(k[i].pont==2)
                ketto++;
            else
                harom++;
        }
        if(kezd>k[i].helyes)
            kezd=k[i].helyes;
        if(veg<k[i].helyes)
            veg=k[i].helyes;
    }
    cout<<"3.feladat\nAz adatfajlban "<<matek;
    cout<<" feladat van, 1 pontot er "<<egy;
    cout<<" feladat, 2 pontot er "<<ketto;
    cout<<" feladat, 3 pontot er "<<harom<<" feladat\n";
    cout<<"4. feladat:\nA valaszok szamerterke "<<kezd<<"-tol "<<veg<<"-ig terjed\n";
    cout<<"5. feladat:\nTemakorok: ";
    for(int i=0; i<db; i++)
    {
        if(k[i].elso)
        {
            int szamol=1;
            cout<<k[i].temakor<<" (";
            for(int j=i+1; j<db; j++)
            {
                if(k[i].temakor==k[j].temakor)
                {
                    k[j].elso=false;
                    szamol++;
                }
            }
            cout<<szamol<<" ";
        }
    }
}

```

```

cout<<"\n6.feladat\nMilyen temakorbol szeretne kerdest kapni? ";
string temakor;
cin>>temakor;
int tomb[db];
int t_db=0;
for(int i=0; i<db; i++)
{
    if(k[i].temakor==temakor)
    {
        tomb[t_db]=i;
        t_db++;
    }
}
int index=tomb[rand()%t_db];
cout<<k[index].kerdes<<" ";
int valasz;
cin>>valasz;
cout<<"A valasz ";
if(valasz==k[index].helyes)
    cout<<k[index].pont<<" pontot er.\n";
else
    cout<<"0 pontot er.\nA helyes valasz: "<<k[index].helyes<<endl;
int generalas[10];
for(int i=0; i<10; i++)
{
    generalas[i]=rand()%db;
    bool van=false;
    while(!van)
    {
        bool seged=false;
        for(int j=0; j<i; j++)
        {
            if(generalas[i]==generalas[j])
            {
                seged=true;
                break;
            }
        }
        if(seged)
            generalas[i]=rand()%db;
        else
            van=true;
    }
}
ofstream t("tesztfel.txt");
int osszpont=0;
for(int i=0; i<10; i++)
{
    int szam=generalas[i];
    t<<k[szam].pont<<" "<<k[szam].helyes<<" "<<k[szam].kerdes<<endl;
    osszpont+=k[szam].pont;
    if(i==9)
        t<<"A feladatsorra osszesen "<<osszpont<<" pont adható.";
}
}

```