

Име и презиме:
Индекс:
Паралелка:



ВТОР КОЛОКВИЈУМ ПО ПРЕДМЕТОТ
Структури со програмирање, Термин 1
04.06.2015

1. Дадени се две двојно поврзани листи чии јазли имаат **две инфо полиња**, а секое инфо поле претставува цел број. Да се формира трета двојно поврзана листа чии јазли исто така ќе имаат две инфо полиња во која **наизменично** ќе се вметнуваат јазли од двете листи на следниов начин: од **првата** листа се вметнуваат јазли кои се наоѓаат на **непарни позиции**, почнувајќи од **почеток на листата**, а од **втората** листа се вметнуваат јазли кои се наоѓаат на **парни парни** позиции, но почнувајќи од **крајот на листата**. Првиот јазол во првата листа е на позиција 1 и последниот јазол во втората листа е на позиција 1.

Притоа од **првата листа** се додава само **првото инфо поле** од јазолот кој се вметнува, а второто се поставува на **0**, додека од **втората листа** се додава само **второто инфо** поле од јазолот кој се вметнува, а првото се поставува на **0**.

Ако една од листите се измине пред другата, од листата во која се уште има јазли се додаваат **сите јазли**, со двете инфо полиња.

Забелешка: Кодот со класите и функциите за листи, како и главната програма е сместен во датотеката **Termin1Zad1.cpp**

Пример:

Листа 1: |2|5|<->|5|6|<->|7|6|<->|9|8|<->|3|1|<->|2|4|

Листа 2: |3|1|<->|8|5|<->|7|1|<->|1|6|<->|5|7|<->|3|1|<->|4|8|<->|4|5|<->|3|8|

Нова листа: |2|0|<->|0|5|<->|7|0|<->|0|1|<->|3|0|<->|0|6|<->|7|1|<->|8|5|<->|3|1|

Бодување:

- **наизменично изминување (20 поени)**
- **вметнување во трета листа (15 поени)**
- **изминување на листа со преостанати елементи (10 поени)**
- **внесување на втора листа во главна програма (5 поени)**

Најчести случаи на одземени поени:

- **неточна проверка дали даден елемент има два соседи после/пред него (-5 поени)**
- **неизминување на остаток од една листа (-5 поени)**

2. За даден **збор** се знае текстуалниот запис (стринг) и неговиот реден број во дадена реченица.

Именките се зборови кои дополнително имаат род (машки - 0, женски - 1 или среден - 2) и се во единица - 0 или множина - 1. **Глаголите** се зборови кои дополнително имаат време (минато - 0, сегашност - 1 или иднина - 2).

• За секој збор треба да се напише функција **presmetajDolzhina()** која го враќа бројот на букви во зборот. Ако зборот е именка и е во множина, доколку завршува на „и“, „е“ или „а“ од дужината се одзема еден, додека ако завршува на „ње“ или „ња“ од дужината се одзема три.

Ако зборот е глагол и е во минато време, ако завршува на „в“, од дужината се одзема еден, а ако завршува на „вme“ од дужината се одзема три. Ако глаголот е во идно време и започнува на „kje“ од дужината се одзема четири.

• Да се напише функција **pechatiTzbor()** која го печати зборот на следниов начин, ако зборот е глагол се печати „Zborot <zbor> e glagol i ima <dolzhina> bukvi“, додека ако зборот е именка се печати „Zborot <zbor> e imenka i ima <dolzhina> bukvi“.

• Во основната класа **Zbor** да се преоптовари **операторот <** кој треба да провери дали зборот кој ја повикува функцијата(операторот) има помалку букви од зборот кој се пренесува како аргумент.

Име и презиме:

Индекс:

Паралелка:



Да се креира класа **Rechenica** која е составена од низа од зборови (и глаголи и именки), а познат е и бројот на зборови во реченицата. Во класата да се напише функција **sortiraj()** која ги сортира зборовите во реченицата во растечки редослед според бројот на букви и на крај ја печати сортираната реченица (ги печати само зборовите).

Главната програма е дадена на другата страна.

Помош: За пресметка на должината на стринг str може да ја користите функцијата **str.length()**. Последната буква од стрингот str може да ја добиете со **str[str.length()-1]**.

Забелешка: Главната програма е сместена во датотеката **Termin1Zad2.cpp**

```
int main()
{
    Zbor *zborovi[5];
    zborovi[0] = new Imenka("fakulteti", 1, 0, 1);
    zborovi[1] = new Glagol("odevme", 2, 0);
    zborovi[2] = new Glagol("kje pishuvame", 3, 2);
    zborovi[3] = new Imenka("vremenja", 4, 2, 1);
    zborovi[4] = new Glagol("peam", 5, 1);
    Rechenica r(zborovi, 5);
    r.sortiraj();
}
```

Бодување:

- основна класа и две изведени, членови и конструктори (10 поени)
- функција **presmetajDolzhina()** (10 поени)
- функција **pechatiTzbor()** (5 поени)
- оператор < (5 поени)
- класа **Rechenica** со членови (5 поени)
- функција **sortiraj()** (15 поени)

Најчести случаи на одземени поени:

- функции кои не се дефинирани како **virtual** (-5 поени)
- неправилно сортирање (-10 поени)
- грешка во пристап до букви на збор (-5 поени)
- заборавен двоен покажувач во класа **Rechenica** (-2 поени)

Име и презиме:
Индекс:
Паралелка:



ВТОР КОЛОКВИМ ПО ПРЕДМЕТОТ
Структури со програмирање, Термин 2
04.06.2015

1. Дадени се **две двојно поврзани листи** чии јазли имаат **две инфо полиња**, а секое инфо поле претставува знак (+ или -). Да се формира трета двојно поврзана листа чии јазли исто така ќе имаат две инфо полиња во која **наизменично** ќе се вметнуваат јазли од двете листи на следниов начин: од **првата** листа се вметнуваат јазли кои се наоѓаат на **парни позиции**, почнувајќи од **крај на листата**, а од **втората** листа се вметнуваат јазли кои се наоѓаат на **непарни парни позиции**, но почнувајќи од **почеток на листата**. Првиот јазол во првата листа е на позиција 1 и последниот јазол во втората листа е на позиција 1. Притоа од **првата листа** се додава **првото инфо поле** од јазолот кој се вметнува, а второто се поставува со знак 'X', додека од **втората листа** се додава само **второто инфо** поле од јазолот кој се вметнува, а првото се поставува со знак 'Y'.

Ако една од листите се измине пред другата, од листата во која се уште има јазли се додаваат **сите јазли**, со двете инфо полиња.

Забелешка: Кодот со класите и функциите за листи, како и главната програма е сместен во датотеката **Termin2Zad1.cpp**

Пример:

Листа 1: | + | - | <-> | - | + | <-> | - | - | <-> | + | + | <-> | - | + | <-> | + | + |
Листа 2: | - | + | <-> | + | + | <-> | + | - | <-> | + | - | <-> | - | + | <-> | - | + |
Нова листа: | - | X | <-> | Y | + | <-> | - | X | <-> | Y | - | <-> | + | X | <-> | Y | + | <-> | - | - | <-> | + | + | <-> | - | + |

Бодување:

- **наизменично изминување (20 поени)**
- **вметнување во трета листа (15 поени)**
- **изминување на листа со преостанати елементи (10 поени)**
- **внесување на втора листа во главна програма (5 поени)**

Најчести случаи на одземени поени:

- **неточна проверка дали даден елемент има два соседи после/пред него (-5 поени)**
- **неизминување на остаток од една листа (-5 поени)**

2. За секој **човек** се знае името, годината на раѓање и адресата на живеење. Вредностите на овие атрибути единствено може да се внесат со функцијата **vnesi()** и затоа предефинирано се поставени на нула (за број) и празен стринг (за стринг).

Дете е човек за кого дополнително се чуваат податоци за името на училиштето и одделението кое го посетува (дозволени вредности се помеѓу 1 и 9). Вредностите и за атрибутите и на детето се внесуваат од тастатура со функција **vnesi()**, а предефинирано се поставени на нула (за број) и празен стринг (за стринг).

Вработениот е човек за кого дополнително се чуваат податоци за фирмата во која работи и и бројот на години стаж. Вредностите и за атрибутите и на вработениот се внесуваат од тастатура со функција **vnesi()**, а предефинирано се поставени на нула (за број) и празен стринг (за стринг).

- Функцијата **vnesi()** треба за детето да бара информации за името, годината на раѓање, адресата, училиштето и одделението, додека за вработениот треба да бара податоци за името, годината на раѓање, адресата, фирмата и годините стаж.

- Да се напише функција **pechat()** која за детето ги печати информациите за името, годината на



Име и презиме:

Индекс:

Паралелка:

раѓање, адресата, училиштето и одделението, додека за вработениот ги печати информациите за името, годината на раѓање, адресата, фирмата и годините стаж.

- Во основната класа да се преоптовари **операторот >** кој треба да провери дали човекот кој ја повикува функцијата(операторот) е постар од човекот кој се пренесува како аргумент.

Да се креира класа **Semejstvo** која е составена од низа од луѓе (и деца и вработени), а познат е и бројот на членови во семејството. Во класата да се напише функција **sortiraj()** која ги сортира членовите во семејството во растечки редослед според бројот на години и ја печати сортираната низа.

Главната програма е дадена на другата страна.

Забелешка: Главната програма е сместена во датотеката **Termin2Zad2.cpp**

```
int main()
{
    Chovek *lugje[4];
    lugje[0] = new Dete();
    lugje[0].vnesi("Toni", 2007, "Adresa 1", "Uchilishte 1", 3);
    lugje[1] = new Vraboten();
    lugje[1].vnesi("Ana", 1980, "Adresa 1", "Firma 1", 10);
    lugje[2] = new Dete();
    lugje[2].vnesi("Marija", 2005, "Adresa 1", "Uchilishte 1", 5);
    lugje[3] = new Vraboten ();
    lugje[3].vnesi("Mitko", 1980, "Adresa 1", "Firma 2", 12);

    Semejstvo s(lugje, 4);
    r.sortiraj();
}
```

Бодување:

- основна класа и две изведени, членови и конструктори (10 поени)
- функција **vnesi()** (10 поени)
- функција **pechatit()** (5 поени)
- оператор **>** (5 поени)
- класа **Semejstvo** со членови (5 поени)
- функција **sortiraj()** (15 поени)

Најчести случаи на одземени поени:

- функции кои не се дефинирани како **virtual** (-5 поени)
- неправилно сортирање (-10 поени)
- грешка во бројот на пренесени аргументи во функцијата **vnesi()** во основната класа (-5 поени)
- заборавен двоен покажувач во класа **Semejstvo** (-2 поени)