

Име и презиме:  
Индекс:

**Испит по предметот**  
**Структури со програмирање**  
11.01.2016

**УПАТСТВО ЗА ЧУВАЊЕ НА ЗАДАЧИТЕ:**

1. Отворете Eclipse (во командна линија напишете: **eclipse &**)

2. Сменете ја работната патека:

**File -> Switch workspace ->**

<b>322 А и 322 В</b>	<b>/export/home/mrezhiBR/ispit/indeks_godina</b>
<b>322 Б, 121 А и 121 Б</b>	<b>/home/ispit/ispit/indeks_godina</b>

3. Отворете пребарувач ( во командна линија напишете: **firefox &**)

Датотеките се наоѓаат на адреса **192.168.0.10** во **фолдерот SSP**

4. Прокетите именувајте ги како **SsPZadX** (**X** е бројот на задачата), додека датотеките именувајте ги како **SsPZadX.cpp** (**X** е бројот на задачата).

5. Во секоја датотека, на почеток напишете **коментар со вашето име и презиме и индекс!!!**

**Задача 1.** Кога се продаваат акции на некоја компанија, стекнатиот капитал (или понекогаш загуба) се пресметува како разлика помеѓу продажната цена на акциите и цената за која тие биле првично купени. Според стандарден сметководски принцип, најпрво се продаваат акциите кои **најдолго** се задржале во компанијата (**најстарите** акции). На пример, доколку во **Ден 1** компанијата **купи** 5 акции од по \$20, во **Ден 2** **купи** 3 акции од по \$15, а во **Ден 3** **купи** 7 акции од по \$25 и во **Ден 4** одлучи да **продаде** 10 акции од по \$23, тогаш од вкупно 10 продадени акции, 5 биле купени во Ден 1, 3 биле купени во Ден 2, додека 2 биле купени во Ден 3. Стекнатиот капитал изнесува:

$$(5 * (\$23 - \$20)) + (3 * (\$23 - \$15)) + (2 * (\$23 - \$25)) = 5*\$3 + 3*\$8 + 2*(-\$2) = \$35.$$

На влез најпрво се внесува бројот на купени акции, па во секој ред се внесува бројот на акции и нивната цена, одделени со празно место, во хронолошки редослед. На крај се внесува бројот и цената на продадени акции. Да се пресмета стекнатиот капитал.

**Не е дозволено користење на низа!!!**

**Пример за влез:**

3  
5 20  
3 15  
7 25  
10 23

**Забелешка:** Кодот со класите и функциите за магацини и редови е сместен во датотеката **SsPZad1.cpp**

Име и презиме:

Индекс:

**Задача 2.** Дадена е двојно поврзана листа (не е кружна, терминирана е на двата краеви) чии јазли содржат целобројни вредности. Од тастатура се весува еден цел број кој го означува бројот на јазли низ кои треба да се придвижите во листата (доколку внесениот број од тастатура го надминува бројот на јазли во листата, откако ќе стигнете до последниот јазол, се враќате на првиот јазол).

После првото придвижување, откако ќе застанете на одреден јазол, низ листата се движите онолку пати колку што изнесува неговата вредност, се додека не застанете на јазол чија вредност е еднаква на бројот на движења од претходниот јазол.

**Пример:** Ако е дадена листата  $|5| \leftrightarrow |2| \leftrightarrow |3| \leftrightarrow |4| \leftrightarrow |7| \leftrightarrow |6|$  и од тастатура се внесе 5, јазолот на кој треба конечно да застанете е јазолот со вредност 6 (Првиот пат застанувате на петтиот јазол  $|7|$ . Потоа се придвижувате 7 пати со што застанувате на јазолот  $|6|$ . Од  $|6|$  се придвижувате уште 6 пати и застанувате пак на јазолот  $|6|$  кој има иста вредност со бројот претходните движења, при што прекинувате).

**Забелешка:** Кодот со класите и функциите за листи е сместен во датотеката **SsPZad2.cpp**

**Задача 3.** За секој **одмор** се знае името на дестинацијата (динамички алоцирана низа од знаци), колку дена трае (цел број), основната цена на аранжманот (цел број), број на понудени факултативни посети, како и цена за секоја понудена факултативна посета (динамички алоцирана низа од цели броеви). Одморот може да биде **летен** или **зимски**. За **летниот** одмор дополнително се чува информација за типот на превоз (автомобил - 0, автобус - 1, авион - 2), како и информација дали аранжманот е полупансион - 1 или цел пансион - 2. За **зимскиот** одмор дополнително се чува информација дали е уплатена ски-карта, како и цена за ски-картата за еден ден.

- Да се напише **функција presmetajCena()** која ја пресметува вкупната цена која треба да се плати. Цената се пресметува на тој начин што основната цена се собира со цената на сите факултативни посети. Доколку станува збор за летен одмор, за полупансион се доплаќа 100, а за цел пансион се доплаќа 150. Ако превозот е со авион се доплаќа уште 100. Доколку станува збор за зимски одмор, за уплатена ски карта се собира и цената на ски картатата за секој ден од одморот.
- Да се преоптовари **операторот >** кој проверува дали првиот одмор (независно од типот) има поголема вкупна цена од вториот одмор.
- Надвор од класите да се напише **функција najdiSporedKategorija()** која како аргументи прима низа од **одмори** независни од типот, број на одмори во низата, број на денови и име на дестинација. Функцијата треба да го најде и испечати најскапиот одмор во низата на дадената дестинација, кој не трае повеќе од бројот на денови пренесени како аргумент.

Главната програма е дадена во продолжение.

**Забелешка:** Кодот сместете го во датотеката **SsPZad3.cpp**

```
int main()
{
    Odmor *odmori[3];
    odmori[0] = new Leten("Maldivi", 10, 1000, 4, {50, 30, 20, 60}, 2, 2);
    odmori[1] = new Zimski("Kopaonik", 7, 500, 3, {20, 25, 20}, true, 40);
    odmori[2] = new Leten("Santorini", 7, 700, 3, {10, 15, 30}, 2, 1);

    najdiSporedkategorija(odmori, 3, "Maldivi", 8);
}
```

**Времетраење на испитот: 120 мин.**