

Име и презиме:  
Индекс:  
Број на терминал:

**Испит по предметот**  
**Структури со програмирање**  
24.06.2015

**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.** Дадена е **двојно поврзана листа** чии јазли како вредности содржат цели броеви. Листата се изминува **наизменично: почнувајќи** од почетокот, потоа од крајот, за да потоа постапката се повторува. Најпрво, се разгледува вредноста од почетниот јазол. Толку пати се поместуваме во десно и доколку вредноста на јазолот на кој сме застанале е помала од три, јазолот над кој сме застанале се отстранува од листата. Потоа се разгледува вредноста на последниот јазол; толку пати се поместуваме во лево и доколку вредноста на јазолот над кој сме застанале е помала од три, јазолот над кој сме застанале се отстранува од листата. Потоа, се продолжува со движење, повторно почнувајќи од почетокот и од крајот, се додека не се стигне до јазол кој има вредност -1.

**Забелешка:** Кодот со класите и функциите за листи, како и главната програма е сместен во датотеката **SsPZad1.cpp**

**Пример:**

Почетна листа: |3| <-> |1| <-> |2| <-> |2| <-> |5| <-> |-1| <-> |1| <-> |2| <-> |2|

Листата после отстранувањето:

|3|<->|1|<->|2|<->|5|<->|-1|<->|2|<->|2|

Се отстранува пов

Се изминува трет

Се изминува четврт и  
завршува изминувањето

Се отстранува втор

Име и презиме:  
Индекс:  
Број на терминал:

**Задача 2.** На едно место паѓаат  $N$  **топчиња** кои имаат три бои: **C** – црвена, **S** – сина, **Z** – зелена. Доколку две топчиња со иста боја паднат едно над друго, истите треба да се поништат. Доколку падне црвено топче, ги поништува сите топчиња кои паднале пред него и само тоа останува непоништено. За дадена секвенца од бои на топчиња, да се најде кои и колку топчиња ќе останат непоништени.

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

**Пример:**

Влезна секвенца од бои: S Z Z Z C C S Z S S S  
Непоништени останале 4 топчиња: C, S, Z, S

**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 помала вкупна цена од вториот одмор.
- Надвор од класите да се напише **функција najdiNajblizok()** која како аргументи прима низа од **одмори** независни од типот, број на одмори во низата, како и буџет на располагање. Функцијата треба да го најде и испечати најскапиот одмор во низата кој може да се плати со пренесениот буџет.

Главната програма е дадена во продолжение.

**Забелешка:** Кодот е сместен 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);

    najdiNajblizok(odmori, 3, 3000);
}
```

**Времетраење на испитот: 120 мин.**