Име и презиме:
Индекс:

# Испит по предметот
# Структури со програмирање
09.09.2015

## УПАТСТВО ЗА ЧУВАЊЕ НА ЗАДАЧИТЕ:

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

2. Сменете ја работната патека:

   **File -> Switch workspace ->**

| | |
|---|---|
| **322 А и 322 В** | **/export/home/mrezhiBR/ispit/indeks_godina** |
| **322 Б, 121 А и 121 Б** | **/home/ispit/ispit/indeks_godina** |

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. Од тастатура се внесува еден цел број кој ја претставува позицијата на јазолот во листата до кој треба да се придвижите. Доколку соседите на овој јазол се истовремено поголеми од него или се истовремено помали од него, јазолот на кој сте застанале треба да се отстрани од листата. Доколку овој услов не е исполнет, соседните јазли на овој јазол треба да се отстранат од листата. Не смеат да се користат готови функции за отстранување на јазол во листа, тоа треба да се направи со менување (преповрзување) на врски.
Да се внимава на специјалните случаи кога јазолот на кој сте застанале нема два соседи или соседите на јазолот на кој сте застанале немаат два соседи.

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

**Пример**:
Почетна листа: |3| <-> |8| <-> |1| <-> |2| <-> |5| <-> |6|
Ако внесениот број од тастаура е 4, треба да застанете на јазолот со инфо поле 2 и да ги избришите неговите соседи, јазлите со инфо полиња 1 и 5.

Листата после отстранувањето:
|3|<->|8|<->|2|<->|6|

Име и презиме:
Индекс:

**Задача 2.** Пред канцеларијата на асистентката по математика се наоѓаат N студенти кои чекаат за увид. За секој студент се знае името, презимето и бројот на поени за четирите задачи кои ги полагале на испитот (секоја задача носи максимум 25 поени). Студентите влегуваат еден по еден, но пред да влезат на увид прават прераспределба во два реда, според следниве критериуми: доколку студентот во редот има доволно поени за да го положи испитот (минимум 60), останатите студенти го препраќаат во нов ред за положени студенти. Доколку студентот го нема положено испитот треба да оди во редот за неположени студенти, но студентите кои имаат освоено помалку од 30 поени, во редот за неположени студенти треба да се најдат позади студентите кои имаат повеќе од 30 поени. На крај да се испечатат имињата на студентите кои чекаат во редот за неположени студенти, а потоа имињата на студентите кои чекаат во редот за положени студенти.

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

**Пример**:
Влезен ред: |Krste Krstev 10 30 20 0| |Mia Melova 10 0 20 0| |Teodora Taneva 20 20 20 10| |Petar Petrov 5 5 10 5| |Kate Kochova 10 10 10 0| |Marija Mahova 20 0 20 0| |Teodor Pecov 25 25 20 10| |Ivo Ivkov 15 10 15 25| |David Dekov 10 10 10 10|
На излез се печатат студентите во следниов редослед:
Неположени: Marija, David, Mia, Petar, Kate
Положени: Krste, Teodora, Teodor, Ivo

**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("Maldivi", 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.**