

Име и презиме:
Индекс:
Паралелка:



ИСПИТ ПО ПРЕДМЕТОТ
Податочни структури и програмирање
17.09.2018

1. Во даден **влезен ред** внесени се произволен број на елементи (максимум 50). Елементите се од типот `int` броеви. Од влезниот ред да се формираат **две излезни поврзани листи** во кои ќе се запишат броевите на следниот начин. Првата листа ќе ги содржи броевите без соседни дупликати, а во втората листа ќе се запишат исфрлените дупликати. Поделбата на влезниот ред во две излезни листи да се направи во глобална функција **`podeli`** која ќе прими како аргументи покажувач кон ред и референци кон две поврзани листи и ќе ги направи соодветните операции.

Пример:

Влезен ред: | 113, 83, 83, 119, 85, 85, 101, 67, 67, 67, 114, 69, 69, 116, 83, 83, 83, 121 |
Излезна листа 1: (113->83->119->85->101->67->114->69->116->83->121)
Излезна листа 2: (83->85->67->67->69->83->83)

Забелешка: Главната програма е дадена во датотеката `PSPzad1T1.cpp`. Да се напишат само структурите и функциите кои се неопходни за работата на главната програма.

2. Да се креира хиерархија на класи за работа со **канцеларии**. За секоја канцеларија е познат реден број за идентификација, бројот на инвентарни елементи и динамички алоцирана низа со инвентарни елементи (за секој инвентарен елемент се знае неговиот тип (M за маса и S за стол) и неговото ID). Цената на секоја маса е 5000 денари, а цената на секој стол е 4000 денари. Канцелариите може да бидат **асистентска** или **професорска**.

За асистентската канцеларија дополнително се чува и бројот на испитни тетратки кои ги има во канцеларијата. Цената на секоја тетратка е 20 денари.

За професорската канцеларија дополнително се чува и информација за тоа дали има проектор во канцеларијата (може да има максимум еден проектор). Цената на проектор е 18000 денари.

За секоја класа да се напишат следните методи и преоптоварени оператори:

- Конструктори и деструктор;
- Оператор `=` за доделување;
- Оператор `+=` за додавање на инвентарен елемент на почеток од динамички алоцираната низа;
- Оператор `<<` за печатење на сите информации;

За класите за асистентска и за професорска канцеларија дополнително да се напише функција **`vrednost`** која ја пресметува вкупната вредност на сите работи што се во канцеларијата. Дополнително да се напише функција **`naјнеopremeHa`** која ќе прима листа од покажувачи кон канцеларии и нивниот број и која со помош на функцијата `vrednost` ќе ја пронајде канцеларијата со најмала вредност на сите работи што се во канцеларијата, а редниот број за идентификација на канцеларијата ќе го врати како излез од функцијата.

Забелешка: Главната програма е дадена во датотеката `PSPzad2T1.cpp`

