

Име и презиме:

Индекс:

Паралелка:



ИСПИТ ПО ПРЕДМЕТОТ
Податочни структури и програмирање
07.06.2018

1. Да се дефинира **една единствена податочна структура** која ќе овозможи работа и како **ред и магацин** истовремено. Елементите се од типот знак (`char`), а може да се чуваат максимум 50 елементи. Да се напишат потребните функции за читање и од почетокот и од крајот и запишување на крајот на редот/магацинот, како и функции за проверка дали е празен и полн.

Да се дефинира и **класа за работа со единечно поврзана листа**. Во класата да се имплементира `default` конструктор кој ќе иницијализира празна листа. Да се реализира и деструктор во класата кој ќе ја избрише поврзаната листа. Да се преоптоварат следните оператори:

- Оператор `+= (char e1)` за додавање на `e1` на крајот на листата и зголемување на должината за еден;
- Оператор `-= (char e1)` за додавање на `e1` на почетокот на листата и зголемување на должината за еден;
- Оператор `<<` за печатење на поврзаната листа во форматот `(a1 a2 ... aN)`.

Да се имплементира и глобална функција `podeli` која ќе прими како аргументи покажувач кон ред/магацин и референца кон единечно поврзана листа и ќе ги направи следните операции. Од редот/магацинот **наизменично** ќе извлекува елементи од почетокот и од крајот, се додека не го испразни. Елементите што ги извлекува од почетокот ќе ги вметнува на почетокот на поврзаната листа (со операторот `-=`), а елементите што ги извлекува од крајот ќе ги вметнува на крајот на поврзаната листа (со операторот `+=`).

Забелешка: Главната програма е дадена во датотеката PSPzad1T1.cpp

2. Да се креира хиерархија на класи за работа со **возови**. За секој воз е познат серискиот број, бројот на вагони и динамички алоцирана низа со вагони (за секој вагон се знае неговата тежина и неговото ID). Возовите може да бидат **дизел** или **електрични**.

За дизел возовите дополнително се чува и основна потрошувачка во литри на 100 километри, просечна цена по литар и висина на еко такса која дополнително се плаќа. Нивната цена за поминати 100 km се пресметува кога на еко таксата ќе се додаде потрошувачката во литри на 100 km (се добива кога основната потрошувачка во литри ќе се зголеми за 5% за секој тон тежина на вагоните) помножена со цената по литар.

За електричните возовите дополнително се чува и основна потрошувачка во kWh на 100 километри и просечната цена по kWh. Нивната цена за поминати 100 km се пресметува кога потрошувачката во kWh на 100 km (се добива кога основната потрошувачка во kWh ќе се зголеми за 10% за секој тон тежина на вагоните) ќе се помножи со цената по kWh.

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

- Конструктори и деструктор;
- Оператор `--` за бришење на вагон на крај од композицијата на вагони;
- Оператор `<<` за печатење на сите информации;

За класите за дизел и за електрични возови дополнително да се напише функција `cena100()` која ја пресметува цената за поминати 100 km. Дополнително да се напише функција `најекономичен` која ќе прима низа од покажувачи кон возови и нивниот број и која со помош на функцијата `cena100()` ќе го пронајде возот со најмала потрошувачка, а неговиот реден број ќе го врати како излез од функцијата.

Забелешка: Главната програма е дадена во датотеката PSPzad2T1.cpp

