

Задачи, които не се компилират, силно нарушават принципите на ООП, имат memory leak, използват глобални променливи или използват библиотеките string и vector, ще бъдат автоматично оценени с 0т.

Чл. 163. (10) от Правилника на Софийския университет: “Студент, който при проверка на знанията въвежда в заблуждение чрез преписване, подсказване и др. п., се наказва по чл. 137, ал. 2 от Правилника на Софийския университет. Същото наказание се налага и при опит за подобно деяние.”

Задача 1:

Смолян е единствената област в България, в която няма жп транспорт. Бъдещото правителство иска да започне изграждане, за което обаче трябва да се измисли информационна система, която да не го свързва с останалите жп линии. Задачата ви е да реализирате система за жп линията. Естествено всяка гара трябва да има гари и влакове. Системата трябва да съдържа система от обвързани гари
Гарата може да бъде 3 вида - крайна гара, разпределителна или обикновена гара.

Обикновената гара съдържа:

- име на гара - низ с максимум 20 символа
- сериен номер на гара - уникален идентификатор
- упътване към следващата гара по линията
- капацитет (брой влакове които може да бъдат на гарата по едно и също време. Броят е максимум 10)
- брой влакове на гарата
- конструктор, който приема име на гара, следващата гара, капацитет и правило по което се получава серийния номер, което използва капацитетът и името
- сигнал - показва дали може произволен влак да потегли
- метод, който позволява потеглянето на влаковете
- метод, който забранява потеглянето на влаковете
- метод, който спира гарата (забранява пристигането на нови влакове)

Крайната гара съдържа:

- име на гара - низ с максимум 20 символа

- сериен номер на гара - уникален идентификатор
- капацитет (брой влакове които може да бъдат на гарата по едно и също време. Броят е максимум 10)
- брой влакове на гарата
- конструктор, който приема име на гара, капацитет и правило по което се получава серийния номер, което използва капацитетът и името
- сигнал - показва дали може произволен влак да бъде изтеглен от мрежата (да бъде изваден)
- метод, който позволява изтеглянето на влак
- метод, който забранява изтеглянето на влак

Разпределителя съдържа:

- име на гара - низ с максимум 20 символа
- сериен номер на гара - уникален идентификатор
- разклонение към следващите 2 паралелни гари по линията
- капацитет (брой влакове които може да бъдат на гарата по едно и също време. Броят е максимум 10)
- брой влакове на гарата
- конструктор, който приема име на гара, следващата гара, капацитет и правило по което се получава серийния номер, което използва капацитетът и името
- сигнал 1 - показва дали може произволен влак да потегли по 1вото разклонение
- метод, който позволява потеглянето на влаковете по 1вото разклонение
- метод, който забранява потеглянето на влаковете по 1вото разклонение
- сигнал 2 - показва дали може произволен влак да потегли по 2рото разклонение
- метод, който позволява потеглянето на влаковете по 2рото разклонение
- метод, който забранява потеглянето на влаковете по 2рото разклонение
- метод, който спира гарата (забранява пристигането на нови влакове)

Всеки влак съдържа:

- номер на влак

- тип на влак (експресен - може да минава дори да е забранено минаването към следваща гара, бърз - може да преминава напред към следващата гара само ако е позволено, пътнически - може да преминава напред ако гарата е обикновена и е позволено и на разпределител само ако е позволено да се мине и по двата разклона)
- индикатор показващ на коя гара се намира влакът
- конструктор с номер на влака, тип на влака (всеки влак започва от първата гара по линията)
- метод за преминаване на следваща гара по подаден номер на разклонение, ако е възможно (ако е невъзможно да се хвърля различна грешка в два случая, ако е забранено да се преминава заради сигнал или тип на влак, или ако не съществува разклонението)
- метод за изтегляне на влак, ако се намира на крайна гара и е възможно

Бонус: Да се реализира час на пристигане и заминаване на влак

Задача 2:

Да се реализира шаблонен клас за стандартен моноид в C++, който съдържа единичен елемент от зададения тип; бинарна операция, която затваря моноида; функция на съгване - бинарна операция в n-мерния случай.

* Да се напише метод `isIsomorphicTo`, която приема друг моноид и кандидат-функция за изоморфизъм и проверява дали двата моноида са изоморфни

* Да се напише метод `constructInfiniteMonoidNaturals`, която конструира безкраен моноид, изоморфен на естествените числа

* Да се напише метод за умножение на моноиди, в който единичните елементи се умножават, а бинарните операции се композират

Забележка: За да имаме изоморфизъм, задължително трябва да бъде първо хомоморфизъм. Това значи:

- $f(\text{единичен елемент}) = \text{единичен елемент}$

- $f(\text{елемент в моноида (бинарната операция в моноида) елемент в моноида}) = f(\text{елемент в моноида}) (\text{бинарната операция в моноида})$

Пример и подсказки: Направете си живота лесен и застопорете бинарната до единична и зациклете единичния елемент. Нека единичен елемент е 2 пък операцията е $\times 3$. Тогава моноидът ще представлява безкрайна циклична група - 2, 6, 18, 54... Направете групата крайна - ограничете я, няма безкрайни оценки в този език. В най-честия случай операцията на съгване е просто многомерно прилагане на горната операция.

Задача 3

Да се реализира имплементация на шаблонната структура Linked List, като след създаване на списъка, се правят произволен брой размествания на двойки кутии, стоящи на произволно генерирани "позиции" и да се провери дали новополучения списък е палиндром.

3->2->1->2->1

1,3 swap

2,4 swap

1->2->3->2->1

Изход: true