

[Табло](#) / [Моите курсове](#) / [Бакалаври, зимен семестър 2021/2022](#) / [КН](#)

/ [Структури от данни и програмиране \(И, ИС, КН1\), зимен семестър 2021/2022](#) / Зимна сесия 2021/22 г.

/ [Практически изпит \(31.01.2022 г.\)](#)

Започнат на Monday, 31 January 2022, 09:00

Състояние Завършен

Приключен на Monday, 31 January 2022, 12:00

Изминало време 2 часа 59 мин.

Информация

Упътване

- Решението си трябва да разпишете в шаблонния файл, който беше качен в системата.
- В заглавния коментар трябва да попълните личните си данни.
- За всяка от задачите трябва да качите точно един **.cpp** файл. Името му трябва съвпада с факултетния ви номер и номера на задачата, която решавате. Например, ако факултетният ви номер е 12345, файловете за двете задачи трябва да се казват **12345-1.cpp** и **12345-2.cpp**.
- Решението си трябва да предадете преди изтичането на времето за работа. След това формата за предаване става неактивна и решения не могат да се предават.

Въпрос **1**

Отговорен

От максимално 15,00

Трябва да напишете система за настаняване на клиенти в бар Мраз. В заведението има само един бар за сядане, където всички места са номерирани последователно от 1 до N. На всяко място в бара има кранче за някои (не задължително всички) от напитките, които се предлагат: бира, водка, уиски, текила, кола, ром, джин и мента. Вашата задача е да настанявате групи пристигащи клиенти при следните условия:

- клиентите от всяка група трябва да останат в същият ред, в който са дошли;
- клиентите от всяка група може да седат на максимум 5 места един от друг. Помежду им може да има други клиенти;
- всеки клиент иска да пие някои от напитките и може да бъде настанен само на място, на което има всички напитки, които той иска.

В задачата **не можете** да използвате алгоритми и структури от данни от STL.

а) (10 т) Напишете функция, която приема като аргумент едносвързан списък описващ местата в бара и двусвързан списък, който описва групата клиенти. В бара може да има вече настанени клиенти. Ако е невъзможно да се настани новата група, функцията трябва да върне грешка. Използвайте следният прототип (сами преценете какви членове да има в структурите, но прототипът не може да се променя):

```
struct BarSlot;  
struct Client;  
bool place(BarSlot *bar, Client *clients);
```

б) (2 т) Освен това, напишете и функция, която приема описание на бар в едносвързан списък, и масив от групи от клиенти. Функцията трябва да провери дали всички групи от клиенти могат да бъдат настанени в някакъв ред. Използвайте следният прототип:

```
bool placeAll(BarSlot *bar, Client **groups, int groupCount);
```

Не е нужно да имплементирате пълни класове за контейнери. Възможно е нещата да се направят така, че дадените прототипи да приемат директно указател към възел в списък.

в) (3 т) Демонстрирайте функцията в програма, която прочита данните за бара и групите клиенти от клавиатурата, записва ги в съответните структури от данни и опитва да ги настани. След това извежда на екрана резултата от настаняването и кои групи не могат да бъдат настанени.

⚙ [82140-1.cpp](#)

Въпрос **2**

Отговорен

От максимално 20,00

Напишете програма, която работи с дървета съдържащи във възлите си цели числа и имащи произволна разклоненост. Числата са произволни стойности от тип `int` и могат да се повтарят в дървото. Единственото ограничение е, че един възел не може да има два преки наследника с една и съща стойност.

В тази задача можете да използвате структурите от данни и алгоритмите от стандартната библиотека.

Вашата програма трябва да реализира описаните по-долу функции.

а) (5 т) Функция, която прочита дърво от подаден чрез името си текстов файл. Във файла дървото е описано по следния начин:

- всеки ред представлява едно ниво от дървото. Елементите на реда са разделени с един или повече интервали;
- в едно ниво братствата са разделени с вертикална черта `|`. Всеки ред започва и завършва с такава черта;
- ако на дадено ниво елемент е листо, то на следващото ниво братството на неговите наследници е празно `| |`;
- братства съществуват само за елементи от предното ниво (т.е. ако на ниво `N` има листо, то на ниво `N+1` имаме празно братство за него и на всяко по-долно ниво няма следа от този елемент).

б) (5 т) Функция, която по дадени две дървета проверява дали второто се среща някъде в първото. Това означава, че в първото съществува поддърво, чийто корен може да се съпостави на корена на второто и при премахване на елементи от това поддърво ще се получи второто дърво. Позволено е пренареждане на елементите в братство, но не и между нивата на дървото. При съвпадение на два възела трябва да се съблюдава съвпадение на стойностите им.

в) (4 т) Функция, която по подадени две дървета премахва всяко срещане (съгласно точка б) на второто от първото. Ако при премахване на срещането останат под-дървета, за които бащата на коренът им е премахнат, то те да се премахнат също, но сумата от елементите им да се съхрани като един елемент, който да се постави на мястото на корена на премахнатото дърво.

г) (3 т) Функция, която записва дърво във текстов файл, съгласно формата от точка а)

д) (3 т) Напишете програма, която получава имената на три файла от командния ред (чрез `argv/argc`), прочита две дървета от първите два файла, премахва всички срещания на второто дърво от първото, съгласно описаната в точка в) операция и записва резултата в нов файл с име третия аргумент.

Пример:

входно дърво 1

```
| 8 |
| 3 2 9 |
| 2 | 8 6 | 2 | | |
| 8 6 | 11 | 12 | 8 5 |
| 11 | 4 3 | | 11 | 6 |
| | | | |
```

входно дърво 2

```
| 2 |
| 8 6 |
| 11 | |
| |
```

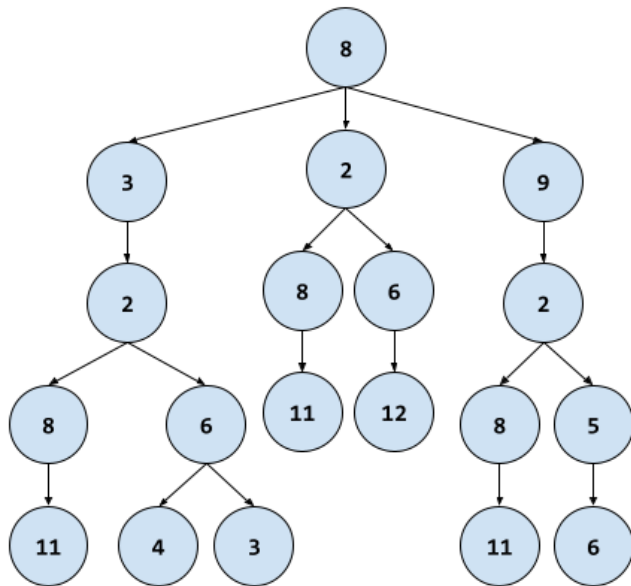
Резултат:

```

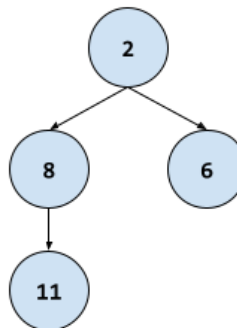
| 8 |
| 3 12 9 |
| 7 | | 2 |
| | 8 5 |
| 11 | 6 |
| | |

```

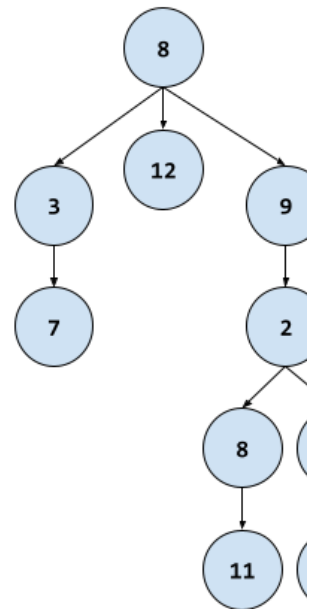
дърво 1



дърво 2



резултат



⚙ [82140-2.cpp](#)

◀ [Шаблон за практическата част](#)

Отиди на ...

[Списък с препоръчвани източници за C/C++](#) ▶