Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

Първо домашно

курсове *Структури от данни;*
Структури от данни и програмиране

специалности *Информатика, Информационни системи и*
Компютърни науки (1-ви поток)

зимен семестър 2021/22 г.

Редакции

25.11 -> Добавена секция с информация за тестове.

5.12 -> Оправен реда на банани/швепс, при входа за информация за клиент.

Условие на задачата

Вие сте шеф на магазин МразМаг, който продава само два продукта – банани и швепс, от които имате неограничени количества на склад. В магазина ви идват клиенти, които искат да пазаруват определено количество банани и швепс. Имате на разположение служители, които можете да изпращате да зареждат количества от бананите и швепса.

Зареждате банани и швепс като изпратите някой от служителите си да ги достави. Това отнема 60 минути и зарежда точно 100 количество от едното – банани или швепс.

В магазина пристигат клиенти и за всеки от тях ще получите следната информация:
- В коя минута от началото на деня е дошъл;
- Количество банани което иска да купи;
- Количество швепс което иска да купи;
- Максимално време (в минути), което ще чака.

Клиентите, идващи в магазина, закупуват исканите продукти, когато има налични. Ако няма налични, изчакват известен период от време за да заредите наличности, а като изтече този период си тръгват, като купуват колкото може от наличните продукти. Когато

клиент си тръгва от магазина заради изтекло време за чакане, той взима каквото може от банани/швепс независимо дали има други чакащи клиенти дошли преди него.

Вашата програма трябва да прецени кога да праща работници да зареждат швепс и банани. Може да изпращате работник да зарежда банани или швепс само **ако няма да имате налични продуктите преди вече пристигнал клиент да си тръгне**. Ако клиент ще си тръгне защото нямате достатъчно банани и швепс, но имате само един наличен работник, тогава изпращате работника да достави това което липсва повече. (Например, имате 0 банани и 0 швепс и идва клиент, който иска да купи 10 банана и 50 швепс, а вие имате само 1 работник. Тогава изпращате работника да зареди швепс). В случай, че и от двете имате еднаква необходимост – зареждате банани. В даден момент можете да изпратите повече от един работник, ако имате свободни.

На входа на вашата програма ще получите цяло положително число W – брой работници, с кото разполагате. След това ще получите цяло положително число C – брой клиенти, които ще обслужите по време на работа. Следват C на брой реда с информация за идващите клиенти. За всеки клиент ще се подадат четири цели числа на един ред – момент на пристигане, желано количество банани, желано количество швепс и максимално време за изчакване.

Вашата програма трябва да изведе съобщение за всяко от следните събития:

- Когато изпращате работник, на един ред изписвате:
    - Символът ‘W’
    - В коя минута
    - Да зарежда какво (banana/schweppes)
- Когато работник се връща изписвате:
    - Символът ‘D’
    - В коя минута
    - Зарежда какво
- Когато клиент си тръгва, на един ред изписвате:
    - Пореден номер на клиента
    - В коя минута
    - Количество банани
    - Количество швепс

Редовете които извеждате трябва да са подредени по времето, в което се случват (минутата). Ако повече от едно нещо се случва в дадена минута, ги извеждате в реда даден по-горе (изпращане, връщане, продажба). Ако изпращате повече от един работник първо да се изведе информация за работниците, изпратени за банани и след тях за тези, изпратени за швепс. Същото се отнася и при едновременно завръщане на повече от един работник. Денят започва в минута 0, като времето отмервате само в минути (не ковертирате в часове). В началото нямате заредени никакви продукти.

Нямате право да изпращате работници да зареждат продукт ако няма клиенти за този продукт във времето в което работника трябва да тръгне. Т.е не може да "гледате в бъдещето" когато изпращате работници преваентивно.

Примерно изпълнение на програмата

Примерен вход:

```
5 4
0 10 0 10
45 35 0 30
46 30 20 100
200 10 10 1
```

Изход:

```
W 0 banana
0 10 0 0
W 46 schweppes
D 60 banana
1 60 35 0
D 106 schweppes
2 106 30 20
3 200 10 10
```

Тестове

Тестове за задачата ще намерите в хранилището на следния адрес https://github.com/semerdzhiev/sdp-2021-22/tree/main/tests/1 Вашето решение трябва като минимум да минава тези тестове. Ако решението ви минава всички тестове това не означава автоматично получаване на пълен брой точки. Тестовете проверяват няколко конкретни случая и не са изчерпателни. Препоръчваме да добавите свои такива, които обхващат повече случаи.

За да може тестовете да проверяват коректността на вашето решение трябва да спазите няколко условия при реализация на кода си.

В interface.h ще намерите класа **`Store`**, който описва интерфейса, който вашето решение трябва да спазва за да може да бъде тествано от автоматичните тестове. Трябва да

имплементирате всички чисто виртуални методи в наследник на този клас. Освен това трябва да имплементирате и функцията `Store *createStore();` която трябва да връща нова инстанция на вашият клас, наследяващ `Store`.

В implementation.cpp ще намерите примерна, празна имплементация. Тя може да ви послужи като начална точка за вашето решение, но може и да не я използвате. Ако решите да не я използвате, трябва да **не** я включвате във вашият проект.

В tests.cpp ще намерите всички тестове, които вашето решение трябва да минава успешно. Тестовете трябва да минават без променяна на кода в този файл. Всеки тест би трябвало да принтира съобщение описващо какво се тества. Ако то не е достатъчно информативно прочетете кода на теста.

Ако смятате че някой тест противоречи на условието, пишете в дискорд канала и тагнете някой от екипа.

В пакпата vs-2019 може да намерите Visual Studio 19 проект, който пуска тестовете срещу празната имплементация. Може да го използвате за база на вашето решение, или за пример. Ако използвате друга среда за разработка и изпитвате затруднения в създаването на проект, ползващ тестовете - обърнете се за помощ към някой от екипа.

ВАЖНО: Не забравяйте, че решението на задачата трябва да може да чете данни от клавиатурата и да пише резултата обратно в конзолата. Същото това решение трябва и да може да бъде тествано с автоматичните тестове. Това означава че трябва да има 2 под проекта (project във Visual Studio) - единият да пуска тестовете, а другият да имплементира вход/изход от потребител.

При реализацията е разрешено да използвате само класовете `string` и `vector` от стандартната библиотека. Всички останали структури от данни или алгоритми, които са ви необходими при решаването на задачата трябва да реализирате сами.