**Редовен изпит, Модул 2: 11.03.2018**

1. **Крипто инвестиции**

В последно време крипто валутите добиха голяма популярност. Вероятно всеки, който чуе Биткойн, Етерeум или Лайткойн, тайно си мисли, че с малко късмет би могъл да си докара сериозни печалби, ако инвестира в някоя от валутите, а защо не и в трите?

Първоначално на входа ще получите четири стойности

Настояща цена на един брой Биткойн в долари

Настояща цена на един брой Етереум в долари

Настояща цена на един брой Лайткоин в долари

Брой на извършените транзакции

В контекста на нашата задача, транзакция означава, че вие сте купили или продали определено количество от някоя от валутите. Разбира се, борсата, на която търгувате начислява комисионна за всяка продажба или покупка. Комисионната е в размер на 0.073456764216789345 от сумата на транзакцията. Сумата се определя от закупеното/продаденото количество от ценните активи, умножено по цената за брой.

За всяка транзакцията от входа ще трябва да се четат още:

Брой на закупени/продадени активи

Име на валутата

**Buy** или **Sell** в зависимост дали купувате или продавате.

**Buy** – купува се съответното количество от посочената валута, и се прибавя към общото количество за нея.

**Sell** – продава се съответното количество от посочената валута, и се вади от общото количество за нея.

След приключване на всички транзакции, от вас се изисква да изпишете на конзолата текущото състояние на вашият крипто портфейл. Това означава да вземете цялото налично количество от всяка една валута и да го умножите по цената за нея. От получената стойност трябва да извадите сумата на комисионните от всяка транзакция.

**Вход**

Цена на Биткойн - реално число в диапазона [0, 20000] с точност 28 цифри след десетичната точка

Цена на Етереум - реално число в диапазона [0, 20000] с точност 28 цифри след десетичната точка

Цена на Лайткоин - реално число в диапазона [0, 20000] с точност 28 цифри след десетичната точка

Брой транзакции – цяло число в диапазона [0, 100].

За всяка транзакция:

Брой закупени/продадени активи – цяло число в диапазона [0, 263 - 1].

Име на валутата Bitcoin или Ethereum или Litecoin – символен низ

Buy или Sell

**Изход**

Цялата наличност в крипто портфейла, форматирана до 16 знака след десетичния разделител

**Ограничения**

Позволено време за изпълнение на програмата /памет: 100ms / 16MB.

**Примери**:

|  |  |  |
| --- | --- | --- |
| Input | Output | Comments |
| 9846.3  845.3  235.6  2  1000  Bitcoin  Buy  2000  Litecoin  Buy | 9559609.8351932759329625 | 1000 \* Bitcoin = 9846300,  комисионна =  9846300 \* 0.073456764216789345  = 723277.3375077729276735,  2000 \* Litecoin = 471200,  комисионна =  471200 \* 0.073456764216789345  = 34612.827298951139364  сумата на всички комисионни: 757890.1648067240670375  9846300 + 471200 - 757890.1648067240670375  = 9559609.8351932759329625 |
| 19845.345672345612564  745.223456  322.5  3  200000000  Bitcoin  Buy  100000  Bitcoin  Sell  15  Ethereum  Buy | 3675383857200.4943232214765377 |  |

1. **Команди**

Може би не всеки обича да получава команди и да изпълнява това, което му бъде заръчано, но тази задача представлява точно това – серия от команди, които трябва да изпълните.

На входа ще получите серия от цели числа разделени от интервал, които трябва да запомните в реда, в който сте ги получили. На следващите редове ще получавате команди –

**push** {element} – добавяте елемента в края на редицата от числа

**pop** – вземате елемента, който се намира на последна позиция, принтирате го и го изтривате

**shift** – вземате последния елемент и го слагате като начален, а началния елемент отива на последно място

**addMany** {position} {element element element} – след името на командата получавате позиция, на която трябва да вмъкнете поредицата дадена към командата, ако такaва позиция съществува

**remove** {position} – премахнете елемента, който се намира на дадената позицията, ако такава позиция съществува

**print** – приключва изпълнението на програмата и принтирате резултатната колекция обърната наобратно, а всеки елемент трябва да бъде разделен от следващия със запетая и интервал

**Вход**

Серия от цели числа в диапазона [-231, 231]

Команди с един или повече елементи разделени от един интервал

**Изход**

При всяка команда pop изпишете елемента, който се премахва

Изпишете колекцията от манипулирани елементи, разделени със запетая и интервал

**Ограничения**

Позволено време за изпълнение на програмата /памет: 100ms / 16MB.

**Примери**:

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 1 2 3 4 5 6 7 8  push 9  push 21  pop  shift  addMany 3 10 20 30  remove 5  print | 21  1, 8, 7, 6, 5, 4, 20, 10, 3, 2, 9 | **push 9 ->>** 1 2 3 4 5 6 7 8 9  **push 21 ->>** 1 2 3 4 5 6 7 8 9 21  **pop ->>** Печатаме 21 и го изтриваме  1 2 3 4 5 6 7 8 9  **shift ->>** 9 2 3 4 5 6 7 8 1  **addMany 3 10 20 30 ->>**  9 2 3 10 20 30 4 5 6 7 8 1  **remove 5 ->>** 9 2 3 10 20 4 5 6 7 8 1  **print ->>** 1 8 7 6 5 4 20 10 3 2 9 |
| 10 20 30 40 50  pop  push 100  remove 2  print | 50  100, 40, 20, 10 |  |

1. **Стрингове**

Като програмисти има няколко особено неприятни ситуации, в които всеки от нас може да попадне – бавен компютър, бавен Интернет и задача за текстообработка. За късмет или не моментът е точно такъв...

Получавате символен низ, който съдържа неопределен брой думи и шаблон. Задачата Ви е да премахнете всяко последно съвпадение на шаблона в текста.

На този етап звучи доста лесно, нали?   
  
За да стане малко по-интересно – след всяко едно премахване на елемент, трябва да извършите и манипулация по шаблона. **Всеки път премахвайте елемента от шаблона, който съответства на индекса, равен на дължината на шаблона, разделена целочислено на 2. След това обърнете шаблона наобратно**. Потърсете за ново съвпадение в текста, ако има такова – премахнете го.

Нека тази логика се повтаря **докато шаблона не бъде изцяло изтрит** или **не намерите повече съвпадения** в текста.

**Вход**

Входът се състои от 2 реда

На първия ред – текст от случайни символи и думи

На втори ред – шаблон

**Изход**

След всички манипулации, входният текст следва да бъде изменен от оригиналното си състояние – изпечатайте го на конзолата

**Ограничения**

Двата низа може да съдържат всякакви ASCII символи.

Позволено време за изпълнение на програмата /памет: 100ms / 16MB

**Примери**:

|  |  |  |
| --- | --- | --- |
| Input | Output | Comments |
| alyakeykala kalakey  key | alakeala kala | След първата промяна текстът бива промен на: alyakeykala kala, шаблонът бива изменен key -> ky -> yk  Текстът се изменя отново: alyakeala kala  Шаблонът също ky -> y  Отново търсим съвпадение - alyakeala kala  Премахваме съвпадението и средния символ на шаблона, в този случай шаблон вече не съществува и прекратяваме изпълнението на програмата. Печатаме текста |
| abcIloveC#def  IloveC# | abcdef | Премахваме последното /в случая и единствено/съвпадение на шаблона и текстът се променя - abcIloveC#def -> abcdef  Обработваме шаблона – махаме средния елемент - IloveC# -> IloeC#, обръщаме шаблона -> #CeolI, търсим за ново съвпадение в текста, такова не е намерено. Приключва програмата с печатане на текста. |
| mmmtm@! sktm  mtm | @! skt | -> mm**mtm**@! sktm    шаблон: mtm  -> **mm**@! sktm       шаблон: mm  -> @! skt**m**         шаблон: m  Изтриваме и последния елемент от шаблона. Печатаме останалото от текста ->  @   ! skt |

**Задача 4. Туристически дестинации**

В последно време туристическа агенция „Слънце“ се радва на изключителен интерес. Агенцията предлага екскурзии до все повече и повече дестинации... Назрял е моментът, в който се взема решение за създаване на база данни, в която ще се съдържа информация за всички дестинации, до които се организират пътувания.

За всяка нова дестинация ще получавате команда в следния формат:

**Add** {country} {city}

Във времето определени дестинации отпадат и биват изтрити от базата данни с командата

**Remove** {city}

Вашата задача е да запаметите всички държави и градовете в тях, до които се организират пътувания. При команда **Remove** {city} се очаква да изтриете посочения град. Ако такъв град не съществува, трябва да бъде изпечатано съобщение **City** {city} not found

При команда **End** програмата приключва и трябва да изпишете на конзолата цялата събрана информация за всяка държава във формат :

{country} has {cities count} cities and they are: {city1}, {city2}

**Вход**

**Команда: Add {country} {city} – добавя град към дадената държава.**

**Команда: Remove {city} – премахва дадения град, ако съществува такъв.**

**Команда: End – прекратява изпълнението на програмата и печата краен резултат.**

**Изход**

Изход в следния формат: Germany has 2 cities and they are: Berlin, Hannover

При команда Remove {city}, изпечатайте City {city} not found, ако няма наличен град с такова име.

Държавите следва да бъдат подредени по брой на градове. Ако броят градове е еднакъв, то подредете държавите по азбучен ред

Градовете се подреждат по реда на влизането

Всички градове трябва да бъдат изброени на един ред, разделени по запетая и интервал.

**Ограничения**

Държавите и градовете са уникални стойности, които не се повтарят. Две държави не могат да имат градове с едно и също име.

Всяко име на държава и град се състои от една цяла дума

Позволено време за изпълнение на програмата /памет: 100ms / 16MB.

**Примери**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Add Germany Berlin  Add Germany Hannover  Add Bulgaria Sofia  Add Bulgaria Plovdiv  Remove Berlin  Add Bulgaria Varna  Add Bolivia Sucre  End | Bulgaria has 3 cities and they are: Sofia, Plovdiv, Varna  Bolivia has 1 cities and they are: Sucre  Germany has 1 cities and they are: Hannover |
| Add Germany Berlin  Add USA Miami  Add USA NewYork  Add UK London  Add UK Bristol  Remove Kaspichan  End | **City Kaspichan not found**  UK has 2 cities and they are: London, Bristol  USA has 2 cities and they are: Miami, NewYork  Germany has 1 cities and they are: Berlin |