**Поправителен изпит, Модул 2: 18.03.2018**

**1. Статистика**

Статистиката е важна наука. С нейна помощ можем да правим оценки върху изминали събития, а на база тези оценки структурираме бъдещи планове и проекти с възможно най-премерен риск, относно изпълнението им.

Собствениците на фирма “Real Estates ltd.” са убедени, че без статистика няма как да развият бизнеса си. За тях е изключително важно да могат да получават информация във всеки момент относно сделките, които осъществяват. По-конкретно интересува ги осреднената стойност на сделките за един агент. Те възлагат този малък проект на вас.

Ще разчитате на входни данни, които ще дават информация за:

Броя агенти, които работят във фирмата

За всеки един агент броя сделки, които е направил

За всяка сделка – квадратура на имота и цена за един квадрат

Вашата работа се състои в това да пресметнете паричната стойност по всяка сделка, като формулата, която се използва е квадратура \* цена на квадрат. Разбира се, за да успеете да достигнете до средната стойност на сделка за един агент трябва да сумирате всички сделки и да ги разделите на броя агенти.

В крайна сметка, изведете информация за получената стойност, като я форматирате до 3-тия знак, след десетичния разделител.

**Вход**

Брой агенти - цяло число в диапазона [0, 20]

Брой на сделките на текущия агент - цяло число в диапазона [0, 20]

За всяка сделка:

Квадратура – цяло число в диапазона [0, 263 - 1]

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

**Изход**

Изведете средната стойност на сделка за агент, форматирана до 3-тия знак, след десетичния разделител.

Ограничения

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

**Примери**:

|  |  |  |
| --- | --- | --- |
| Input | Output | Comments |
| 2  1  1000  75.85  2  300  25.6778  600  13.33 | 45775.670 | 2-ма агенти.  Първият има 1 сделка, със стойност  1000 \* 75.85 = 75850  Вторият има 2 сделки  300 \* 25.6778 = 7703.34  600 \* 13.33 = 7998  75850 + 7703.34 + 7998 = 91551.34  91551.34 / 2 = 45775.670 |
| 1  2  565  37.875969504503939  10  33.6 | 21735.923 |  |

**2. Ученици**

Настана изпитно време. Нека проверим кой какво е научил...

Ще получите поредица от цели числа – всяко число представлява текущия успех на всеки един ученик. Трябва да имате предвид, че системата за изпитване работи по малко по-различен начин от стандартната.

В началото на часа, учителят ще посочва колко ученици ще бъдат изпитани. За всяко изпитване получавате номер на изпитания ученик и поставената оценка. Както знаем в линейните структури съществува индексиране – нека всеки индекс бъде номер на ученик. С други думи, ако получите команда от типа 0 2 това означава, че ученик с номер 0 е получил оценка 2.

Оценяването – получили сте първоначални стойности, ако при изпитване поставената оценка на ученика е по-ниска от текущия му сбор от оценки, тогава трябва да намалите резултата му с размера на оценката. От друга страна, ако получи оценка, по-висока от текущия резултат – увеличете резултата с размера на оценката.

Трябва да се има още нещо предвид – учителят е доста строг. Ако се случи така, че изпитва един или повече пъти ученик и текущият му резултат стане ако текушият резултат е по-малък от половината на първоначалния, ученикът бива изключен. Тоест премахваме го от нашия списък с ученици.

Какво ще стане, ако учителят иска да изпита например 10 ученици, но по време на изпитването се окаже, че всички са изключени? Необходимо е прекратите изпълнението на програмата и да изведете съобщение - All studets has been expelled due bad results!

**Вход**

Първоначални резултати на учениците - серия от цели числа, разделени с интервал

Брой на изпитвания, които предстоят – цяло число в диапазона [1, 20]

За всяко изпитване:

На един ред две цели числа, разделени с интервал.

{номер на ученик} {текуща оценка} -> 0 2

**Изход**

При изключване на всички ученици, изведете съобщение – All studets has been expelled due bad results!

Изпишете колекцията с крайни оценки, разделени с точка и запетая и интервал “; “

**Ограничения**

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

**Примери:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 10 20 30  2  0 5  1 25 | 5; 45; 30 | **10 20 30** – първоначални оценки  **2** – брой на изпитвания  **0 5** –Ученик, номер 0, получава оценка 5. Тъй като оценката е по-ниска от текущия му резултат– намаляваме сбора, съответно получаваме 5 20 30  **1 25** – ученик, номер 1, получава оценка 25. Оценката е по-висока от текущия му резултат, тоест я добавяме към него.  Получаваме 5 45 30  Край на изпитванията. |
| 15 25 35  4  2 40  1 35  0 5  0 6 | 60; 75 | 4 изпитвания.  **2 40** – Ученик, номер 2, с оценка 40 – добавяме я към резултата  получаваме 15 25 75  **1 35** – Ученик, номер 1, с оценка 35 - добавяме я към резултата  получаваме 15 60 75  **0 5** – Ученик, номер 0, с оценка 5, по-ниска от резултата, тоест вадим я от него.  получаваме 10 60 75  **0 6** - Ученик, номер 0, с оценка 6, по-ниска от резултата, тоест вадим я от него. Резултата на ученика става -1, при проверка дали текущия резултат е двойно по-малък от първоначални, разбираме, че е така, съответно ученика бива премахнат.  Остават двама 60 75 |
| 40 50 60  15  2 31  1 26  0 21 | All studets has been expelled due bad results! | 15 изпитвания  **2 31** - Ученик, номер 2, с оценка 31, която е по-ниска от резултата, тоест вадим я от него. Новия резултат е двойно по-нисък от първоначалния – премахваме ученика.  **1 26** - Ученик, номер 1, с оценка 26 която е по-ниска от резултата, тоест вадим я от него. Новия резултат е двойно по-нисък от първоначалния – премахваме ученика.  **0 21** - Ученик, номер 0, с оценка 21, която е по-ниска от резултата, тоест вадим я от него. Новия резултат е двойно по-нисък от първоначалния – премахваме ученика.    Всички ученици бяха изключени. Прекратяваме изпълнението на програмата. |

**3. Пунктуация**

Изтеглихте късата клечка. Вие ще се колегата, който трябва да измисли логиката за обработка на входен текст.

Задачата се състои в следното – получавате символен низ на един ред от конзолата. Необходимо е да създадете нов низ, който ще се състои от символи, които сте взели от първия, но с леки корекции:

**Ето и правилата:**

Всички малки букви от латинската азбука [a-z] нека станат главни

Всички главни букви от латинската азбука [a-z] нека станат малки

Всички цифри, следва да бъдат умножени по 2

Всички знаци: ! # % & ( ) \* - . : , ? @ [ ] \_ { }, следва да бъдат заменени с . (точка).

Заменяйте само за посочените знаци!

Проверете всеки един символ и ако символът попада в изброените по-горе – добавете го с новата му стойност към резултатния низ. Всички други символи следва да бъдат пропускани. Напр. интервал.

**Вход**

Символен низ, съдържащ всякакви символи от ASCII таблицата

**Изход**

Изведете получения символен низ, в обратен ред

**Ограничения**

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

**Примери**:

|  |  |  |
| --- | --- | --- |
| Input | Output | Comments |
| abala L\*ANica1203.!ze{LEee | EEel.EZ..6042ACIna.l.ALABA | abala L\*ANica1203.!ze{LEee  ->  ABALA.l.anICA2406..ZE.leEE  следва да обърнем текста  ->  EEel.EZ..6042ACIna.l.ALABA |
| Lorem IpSum dOlOr Sit amet, a. Earu}m m0oles2tiae ni6][si offi?ciaa. | .AAIC.IFFOIS..21INEAIT4SELO0MM  .URAe.A.TEMATIsRoLoDMUsPiMEROl |  |

**4. Ученически паралелки**

Екипът, отговорен за продажбите във фирмата, в която работите, е получил интересно запитване. Проектът, за който става дума е изцяло насочен към училищното образование. Идея му е да предоставите работещо решение, което да подпомага училищното ръководство да менажира всички паралелки и техните ученици.

Ще получавате набор от команди, които покриват основните училищни действия, а именно – приемане на ученици в дадена паралелка, преместване на ученици от един клас в друг и обединяване на класове.

Команда **Add** {име на ученика} {име на учебна паралелка} – тук от Вас се очаква към посочената учебна паралелка да добавите новия ученик. Разбира се, ако се опитаме да добавим ученик в паралелка, която не съществува – трябва да я създадем.

Команда **Transfer** {име на ученика} **From** {име на паралелка1} **To** {име паралелка2} – тук идеята е да премахнете първият срещнат ученик с даденото име от паралелка 1 и да го добавите към паралелка 2. Разбира се, ако се опитаме да добавим ученик в паралелка, която не съществува – трябва да я създадем. В случай, че ученикът, който местим от паралелка 1 е последният, тоест паралелката остава с 0 ученици – закриите я.

Команда **Merge** {име на паралелка1} {име на паралелка2} – тук от Вас се очаква да вземете всички ученици от паралелка 1 и да ги добавите към учениците от паралелка 2. След това закриите паралелка 1.

Команда **End** – приключва изпълнението на Вашата програма като трябва да дадете информация на конзолата за всички паралелки и учениците в тях.

**Вход**

**Add** {име на ученика} {име на учебна паралелка} - Символен низа, съдържащ три елемента, разделени с по един интервал.

**Transfer** {име на ученика} **From** {име на паралелка1} **To** {име паралелка2} - Няма да се случи така, че да местим несъществуващ ученик от паралелка 1, но може да се случи, така че паралелка 2 да не е налична и да трябва да бъде създадена.

**Merge** {име на паралелка1} {име на паралелка2} – Гарантирано и двете паралелки съществуват

**Изход**

От вас се очаква да изведете изхода по следния начин:

Подредете всички паралелки по брой на учениците в тях в обратен ред, а след това и по азбучен ред по имената на паралелките.

За всяка паралелка: Class name – {име на паралелката}

За всеки ученик: ###{име на ученика}

**\*Имена на учениците трябва да останат в реда на приемането си дадената паралелка!**

**Ограничения**

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

**Примери**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| Add Vankata 10a  Add Misho 10a  Add Peshkata 10b  Add Mimi 10v  Add Ivakis 10g  Transfer Vankata From 10a To 10b  End | Class name - 10b  ###Peshkata  ###Vankata  Class name - 10a  ###Misho  Class name - 10g  ###Ivakis  Class name - 10v  ###Mimi | Добавяме няколко ученици в паралелки 10а, 10b, 10v, и 10g  След това Vankata от 10а трябва да бъде преместен в 10b  При печатане, паралелките подредени по брой ученици в намаляващ ред, а след това и по име. Учениците са подредени по реда на добавяне. |
| Add Vankata 10a  Add Misho 10a  Add Peshkata 10b  Merge 10a 10b  End | Class name - 10b  ###Peshkata  ###Vankata  ###Misho | Добавяме няколко ученици в паралелки 10а и 10b  **Merge** - Всички ученици от 10а преминават в 10b. Старата паралелка се закрива.  Печатаме 10b |
| Add Vankata 10a  Add Misho 10a  Add Peshkata 10b  Transfer Peshkata From 10b To 10a  End | Class name - 10a  ###Vankata  ###Misho  ###Peshkata | **Transfer** – ученик Peshkata бива преместен от 10а в 10b. 10a остава без ученици, следователно се закрива. |