# Символни низове - обработка

Практически упражнения към курса [**"Programming Fundamentals" за ученици**](https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Fundamentals).

Тествайте задачите от тази тема в judge: [https://judge.softuni.bg/Contests/2664](https://judge.softuni.bg/Contests/2664/Символни-низове-обработка)

## Магически променящи се думи

Напишете **метод**, който приема като вход два низа и връща **True** или **False**, ако те са заменяеми, или не. Заменяеми са думи, където символите в първия низ може да бъдат заменени и да се получи втория низ. Пример: "**egg"** и "**add"** са заменяеми, но "aabbccbb" и "nnooppzz" не са. (Първото "b" отговаря на “o”, но тогава то също така отговаря на"z"). Двете думи може да нямат една и съща дължина, ако случаят е такъв, те са заменяеми, само ако по-дългата няма повече от видовете букви на първата ("Clint**"** и"Eastwaat" са заменяеми защото "a" и “t” вече са заменени като "l" и " n " но "Clint**"** и"Eastwood" не са заменяеми защото ‘о‘ и ‚‘d‘ не се съдържат в"Clint").

### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| gosho hapka | true |
| aabbaa ddeedd | true |
| foo bar | false |
| Clint Eastwood | false |

## Сбор на големи числа

Входните данни са два реда – въвеждат се две числа, които може да са големи (от 0 до 1050). Трябва да изведете сбора на тези числа.

Забележка: не използвайте BigInteger или BigDecimal класове за решаването на този проблем.

### Примери

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 23  23 | 46 | 9999  1 | 10000 |

|  |  |
| --- | --- |
| **Input** | **Output** |
| 923847238931983192462832102934572893617836459843471846187346 | 934573817465075391826664309019448 |

## Умножаване на големи числа

Входните данни са два реда – на първия се въвежда голямо число (от 0 до 1050). На втория - едноцифрено число (0-9). Трябва да се изведе произведението на тези числа. Забележка: не използвайте класовете BigInteger или BigDecimal за решаването на този проблем.

### Примери

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вход** | **Изход** |  | **Вход** | **Изход** |  | **Вход** | **Изход** |
| 23  2 | 46 | 9999  9 | 89991 | 923847238931983192462832102  4 | 934573817465075391826664309019448 |

## \*Обработка на числа с представки и наставки

Наков обича математиката. Но той също се интересува от английската азбука много. Той е изобретил игра с цифри и букви от английската азбука. Играта е проста. Получавате низ, състоящ се **от число между две букви**. В зависимост от това дали буквата е пред числото или след него ще извършвате различни математически операции с числото за постигане на резултат.

Първо започнете с буквата преди числото.

* Ако тя е **главна**, **делите** на позицията на буквата в азбуката.
* Ако тя е **малка**, **умножавате** числото по позицията на буквата в азбуката.

**После** преминаваш към **буквата след** числото.

* Ако тя е **главна** **изваждате** позицията си от полученото число.
* Ако тя е **малка** **добавяте** позицията си към полученото число.

Но играта става твърде лесно за Наков и наистина се справя бързо. Той решава да я усложни малко, като правилата са същите, но с **множество** низове, като се иска **общата сума** на всички резултати от стринговете. След като той започна да решава задачата с повече низове и по-големи числа, ставаше доста трудно да смята наум. Така той любезно ви моли да напишете програма, която **изчислява сумата на всички числа, след извършените операции на всяко число.**

**Например**, Дадена е последователността **"A12b s17G"**: имаме два низа **-"A12b" и "s17G".** Извършваме операциите на всяко от числата и ги събираме. Започваме с буквата преди числото на първия низ. А е **главна** и позицията в азбуката е 1. Така че разделяме числото 12 на позиция 1 (12/1 = 12). Тогава минаваме към буквата след числото. b е **малка** и неговата позиция е 2. Така че ние **добавяме** 2 към полученото число (12 + 2 = 14). По същия начин за втория низ s е **малка** и нейната позиция е 19, така че ние **умножаваме** числото (17 \* 19 = 323). Тогава ние имаме **главна** буква G с позиция 7, така че ние **изваждаме** от резултата 7 (323 – 7 = 316). И накрая ние събираме 2 резултата и получаваме **14 + 316 = 330.**

### Вход

Входът е на **един ред, съдържащ последователност от символни ни**зове. Низовете са разделени от един или повече интервали.

Входните данни винаги ще бъде валидни и в описания формат. Няма нужда да го проверите изрично.

### Изход

Печат на конзолата на едно число: общата сума от всички обработени числа, закръглени до две цифри след десетичния разделител

### Ограничения

* The **count** of the strings will be in the range **[1 … 10].**
* The numbers between the letters will be integers in range **[1 … 2 147 483 647].**
* Time limit: 0.3 sec. Memory limit: 16 MB.
* Броят на низовете ще бъдат в интервала [1... 10].
* Числата между буквите ще бъде цели числа в диапазона [1... 2 147 483 647].
* Време: до 0,3 сек, памет : до 16 MB.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментари** |
| A12b s17G | 330.00 | 12/1=12, 12+2=14, 17\*19=323, 323–7=316, **14+316=330** |
| P34562Z q2576f H456z | 46015.13 |  |
| a1A | 0.00 |  |

## Министерство на образованието и науката (МОН)

* Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".



* Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен** **лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

