

Символни низове - увод

Тествайте задачите от тази тема в judge: <https://judge.softuni.bg/Contests/2637/Символни-низове-увод> и <https://judge.softuni.bg/Contests/2664>

1. Кой е пощенския сървър

Създайте **метод**, който получава **низ**, съдържащ **валиден email** и връща **низ**, съдържащ в кой **пощенски сървър** е регистриран този email.

Вход

- На един ред поучавате **валиден email**

Изход

- На един ред извеждате неговия **пощенски сървър**

Упътване

Пощенският сървър е частта от email-а след знака '@'.

Примери

Вход	Изход
pesho@gmail.com	gmail.com
maria@abv.bg	abv.bg
mi1234@hotmail.com	hotmail.com

2. Разтълкуване на email адрес

Създайте **метод**, който получава **низ**, съдържащ **валиден email** и връща **низ**, обясняващ кой е **потребителя** и **пощенския сървър**, в който е регистриран този email.

Вход

- На един ред поучавате **валиден email**

Изход

- На един ред извеждате кой е **потребителя** и **пощенския сървър**

Упътване

Пощенският сървър е частта от email-а след знака '@'.

Примери

Вход	Изход
pesho@gmail.com	Това е потребител pesho в сървър gmail.com
maria@abv.bg	Това е потребител maria в сървър abv.bg
mi1234@hotmail.com	Това е потребител mi1234 в сървър hotmail.com

3. Два домейна от един TLD

Създайте **метод**, който получава **два низа**, съдържащи **валидни домейн имена** и връща **True**, ако те са от един и същ **Top Level Domain** (накратко [TLD](#)) и **False**, ако не са.

Вход

- На два реда поучавате **двата домейна**

Изход

- На един ред извеждате дали те са от един **TLD**

Упътване

TLD е частта от домейн името след последната точка до края.

Примери

Вход	Изход
gmail.com mail.bg	False
abv.bg mail.bg	True
hotmail.com microsoft.com	True

4. Проста проверка за валиден email адрес

Създайте **метод**, който получава **низ**, съдържащ **може би email** и връща **True**, ако въведеното спазва формата за email адрес (user@domain.tld) и **False** в противен случай.

Уточнение

В случая се изисква само опростена проверка - дали присъстват знаците @ и . (точка), дали са в правилния ред и дали някой от компонентите на името (**user**, **domain** и **tld**) не е празен низ. По-подробна проверка - дали низа съдържа валидни символи и т.н. ще направим в урока за регулярни изрази.

Вход

- На един ред поучавате текст, който ще проверяваме дали е **валиден email**

Изход

- На един ред извеждате **True**, ако въведеното изглежда като email адрес и **False** в противен случай.

Примери

Вход	Изход	Коментар
pesho@gmail.com	True	Това е валиден email
maria@abv.bg	True	Това е валиден email
pesho@hotmail	False	Липсва . и TLD частта от домейн името
gosho_mail.ru	False	Липсва @
ivan	False	Липсват домейна и знаците @ и .
ana.abv@bg	False	. и @ са с разменени места
ivan.petrov@abv.bg	True	Това е валиден email

ivan@.bg	False	Липсва домейн името
ivan@abv.	False	Липсва TLD частта от домейна
@mail.bg	False	Липсва кой е потребителя
@.	False	Липсва цялата информация
	False	Нищо не е въведено

5. Обръщане на низ

Създайте **метод**, който получава низ и връща **низ**, получен от същите символи, но в обратен ред.

Вход

- На един ред поучавате **символен низ**

Изход

- На един ред извеждате **обърнатия низ**

Ограничения

Символният низ да се състои от една дума, т.е. да няма интервали и да не се ползва метода **Reverse**

Упътване

Може да решите тази задача по няколко начина:

- да отпечатате всички символи на низа, като го обходите отзад напред
- да конструирате нов низ, в който да прехвърлите символите на първия, в обратен ред
- да прехвърлите низа в масив от символи и да размените първия с последния, втория с предпоследния и т.н.

Примери

Вход	Изход
a	a
aba	aba
alena f anela	alena f anela
alibaba	ababila
baba	abab

6. Палиндром

Създайте **метод**, който получава низ и връща **True** или **False** в зависимост от това дали думата е палиндром или не

Вход

- На един ред поучавате **символен низ**

Изход

- На един ред извеждате **True**, ако низа е палиндром или **False**, ако не е.

Ограничения

Символният низ да се състои от една дума, т.е. да няма интервали

Упътване

Един низ е палиндром, ако прочетен отзад напред е същия, какъвто е и когато го четем отпред назад.

Може да проверите това по няколко начина:

1. да проверите дали низа е равен на обрнатия низ (вижте в предната задача как да го обрнете)
2. да сравните в цикъл първия с последния символ, втория с предпоследния и т.н. до средата на низа. Ако някои от сравняваните символи са различни, извеждаме, че не е палиндром и приключваме програмата с **return**. Ако цикълът приключи и всички символи са еднакви, извеждаме, че е палиндром.

Примери

Вход	Изход
a	True
aba	True
alenaanel	True
alibaba	False
baba	False

7. Сбор на големи числа

Входните данни са два реда – въвеждат се две числа, които може да са големи (от 0 до 10^{50}). Трябва да изведете сбора на тези числа.

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

Примери

Вход	Изход	Вход	Изход
23	46	9999	10000
23		1	

Вход	Изход
923847238931983192462832102 934572893617836459843471846187346	934573817465075391826664309019448

8. Умножаване на големи числа

Входните данни са два реда – на първия се въвежда голямо число (от 0 до 10^{50}). 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	46	9999	89991	923847238931983192462832102	93457381746507539182666430
2		9		4	9019448

9. Преобразуване от 10-ична в N-ична ПБС

Напишете програма, която получава число в 10-ична бройна система и го преобразува в число в N-ична бройна система, където $2 \leq N \leq 10$. Входът се състои от 1 ред, съдържащ две числа, разделени с един интервал. Първото число е основа N, към която трябва да преобразувате. Вторият е число в 10-ична бройна система. **Не използвайте никакви вградени функционалности за преобразуване на числа, опитайте се да напишете свой собствен алгоритъм.**

Упътване

За алгоритъм (от 10-ична в 2-ична) можете да прочетете тази [статия](#).

Алгоритъмът за преобразуване на число от 10-ична в 2-ична бройна система е подобен: вместо “% 2”, ползвайте “% N”.

Вход

- На един ред въвеждате **основа на бройната система и число в 10-ична бройна система**

Изход

- На един ред извеждате **числото в N-ична бройна система**

Примери

Вход	Изход
7 10	13
3 154	12201
5 123	443
4 1000	33220
9 3487	4704

10. Преобразуване от N-ична в 10-ична ПБС

Напишете програма, която взема N-ично число и го преобразува в 10-ично число (0 до 1050), където от $2 \leq N \leq 10$. Входът се състои от 1 ред, съдържащ две числа, разделени с един интервал. Първото число е основата N, към която трябва да преобразувате. Второто е числото N, което трябва да се преобразува. **Не използвайте никакви вградени функционалности за преобразуване, опитайте се да напишете свой собствен алгоритъм**

Вход

- На един ред въвеждате **основа на бройната система и число в N-ична бройна система**

Изход

- На един ред извеждате **числото в 10-ична бройна система**

Упътване

Вижте тази картина за повече яснота за преобразуване от 2-ична в 10-ична БС. Отново, алгоритъмът за преобразуване от N-ична БС е подобен.

Примери

Вход	Изход
------	-------

7 13	10
3 12201	154
5 443	123
4 33220	1000
9 4704	3487

11. Unicode символи

Напишете програма, която преобразува символен низ в последователност от Unicode символни кодове.

Вход

- На един ред въвеждате **символен низ**

Изход

- На един ред извеждате **Unicode** на всеки символ

Примери

Вход	Изход
Hi!	\u0048\u0069\u0021
What?!?	\0057\0068\0061\0074\003f\0021\003f

12. Умножаване на символни кодове

Създайте **метод**, който получава два низа като аргументи и връща сбора от техните произведения от символни кодове на съответни позиции (умножете `str1.charAt(0)` с `str2.charAt(0)` и ги добавете към сбора). След това продължете със следващите два знака. Ако един от низовете е по-дълъг от другия, добавете останалите символни кодове към сбора без умножение.

Вход

- На един ред въвеждате **два низа**

Изход

- На един ред извеждате **сбора от техните произведения от символни кодове на съответни позиции**

Примери

Вход	Изход
Gosho Pesho	53253
123 522	7647

13. 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 "aabbcbb" и "nnoorppzz" 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 'o' и 'd' не се съдържат в "Clint").

Примери

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

14. *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 низове**. 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	330.0 0	12/1=12, 12+2=14, 17*19=323, 323-7=316, 14+316=330
P34562Z q2576f H456z	46015 .13	
a1A	0.00	