

34. В редица са подредени N кутийки, като всяка кутийка съдържа точно една топка, която може да е бяла, зелена или червена. Кутийките са номерирани последователно с числата 1, 2, 3, ..., N. Позволено ви е да изпълнявате единствено операция по размяна на топките в две различни кутийки. Напишете програма flag, която създава последователност от колкото се може по-малко такива операции, след изпълнението на които топките в кутийките ще се подредят в реда на националния ни флаг – най-напред белите, след това зелените, а червените ще заемат кутийките с най-големи номера. Сигурно е, че от всеки цвят има поне по една топка и че първоначално топките не са подредени в правилната последователност.

Вход

От единствен ред на стандартния вход се въвежда символен низ, който съдържа N символа. Поредният символ задава цвета на топката в съответната кутийка и може да бъде:

W – бяла топка;

G – зелена топка;

R – червена топка.

Изход

На стандартния изход програмата трябва да изведе намерената последователност от операции в реда, в който трябва да бъдат изпълнявани. Всяка операция трябва да се изведе на нов ред и да съдържа две цели, положителни числа, разделени с един интервал – номерата на кутийките, чиито топки трябва да бъдат разменени.

Ограничения

$3 \leq N \leq 100$

Пример

Вход	Изход 1 (Правилна, но не минимална последователност от операции)	Изход 2 (Правилна, минимална последователност от операции)
GGWRWRGWG	4 7 3 9 5 7 1 9 2 8 3 7 5 9 6 8	1 5 2 8 8 4 6 9

35. Даден е символен низ, който може да съдържа малки и главни букви от латинската азбука и служебните символи:

* – (тире) . (точка)

% (, (запетая)

=)

+ :

Низът не започва и не завършва със символа *. Низът трябва да се преобразува като всички звездички (символ *) се заменят с интервали и се премахнат всички останали служебни символи (без *). В преобразувания низ, дума наричаме всяка последователност от букви, заградени от двете страни с по един или няколко интервала (разбира се, пред първата и след последната дума няма интервали). Под “правилна” дума се разбира:

- дума, която има нечетен брой букви;
- всяка буква, освен първата, е предхождана от буква, която съвпада с нея или е преди нея в азбуката (напр. думата “abVdeI” е правилна, а “ambso” не е).

Напишете програма с име word, която намира номера и дължината на най-дългата “правилна” дума. Думите в преобразувания низ се броят отляво надясно, като се започва от 1 (броят се всички думи, а не само „правилните”).

Вход:

От стандартния вход се въвежда низ, отговарящ на условието.

Изход:

На един ред на стандартния изход трябва да се изведат две цели числа, разделени с един интервал - номерът и дължината на най-дългата “правилна” дума, като се броят всички думи, а не само „правилните”. Ако липсват „правилни” думи, програмата трябва да изведе две нули, разделени с интервал. Ако има повече от една „правилни” думи с еднаква дължина, да се изведе номерът на първата (най-лявата) от тях.

Ограничения

$0 < \text{брой символи в низа} \leq 255$

Пример:

Вход

L*bbC*abda***CCDb*%-dA**bdMvV

Изход

6 5

36. Числовият триъгълник, част от който виждате, е образуван като вертикално са записани последователно по големина естествените числа, които са кратни на 1, 2, 3, 4 и т.н.

```

1
2 2
3 4 3
4 6 6 4
5 8 9 8 5
6 10 12 12 10 6
7 12 15 16 15 12 7
8 14 18 20 20 18 14 8
: : : : : : : :

```

Напишете програмата `triangle`, която по зададено цяло положително число a , пресмята сбора на числата, записани в този хоризонтален ред, в който за първи път се появява числото a .

Вход

От един ред на стандартния вход се въвежда цяло, положително число a .

Изход

На един ред на стандартния изход програмата трябва да изведе търсения сбор.

Ограничения

$1 \leq a \leq 1000000$

Пример

Вход

6

Изход

20

37. Дадена е квадратна матрица с N реда и N стълба, съдържаща естествени числа не по-големи от N^2 . За всеки ред, колона, главен или второстепенен диагонал на матрицата се търсят множествата от елементи, съдържащи последователни стойности. За всяко открито множество се начисляват точки, според следната таблица:

№	Брой последователни стойности	В ред	В колона	По главен или второстепенен диагонал
1	две	10	20	30
2	три	40	50	60
3	четири	70	80	90
4	пет	100	110	120
5	повече от 5	150	160	170

Не е задължително елементите, съдържащи последователни стойности да са поредни.

Напишете програма `points`, която пресмята сумарния брой точки, които са начислени за дадена матрица.

Възможно е някой ред, колона или диагонал да съдържа повече от едно наредено множество.

Например, ако $N=7$ и 11, 6, 14, 8, 12, 5, 13 е втория ред на матрицата, то този ред съдържа две множества – едното от два елемента (5, 6) и другото от четири елемента (11, 12, 13, 14). Точките са сумата от $10 + 70 = 80$.

Ограничения

$3 \leq N \leq 16$

Вход

От първия ред на стандартния вход се въвежда едно цяло число N .

От следващите N реда се въвеждат по N цели числа – съответния ред от матрицата.

Изход

На единствения ред на стандартния изход да се изведе едно цяло число – получения максимален брой точки.

Пример

Вход

5

1 1 7 1 2

2 10 2 13 2

5 12 13 5 7

3 3 3 11 3

4 12 4 13 12

Изход

360

38. Първокласниците вече са грамотни и могат да пишат думички с латински букви.

Госпожата им диктува думи и тяхната задача е да определят най-дългата дума. Вие може да помогнете, като напишете програма `words`, която намира най-дългата дума и броя на буквите в нея. Ако има няколко най-дълги думи, програмата да изведе думата, чиято първа буква е по-напред в азбуката. Ако и тези думи са няколко, програмата да изведе първата от тях по реда на въвеждането им.

Вход

От файл `dumi.txt` се четат думи, разделени с поне един интервал.

Изход

Програмата извежда на единствения ред на стандартния изход броя на буквите на най-дългата дума, както и самата дума, разделени с един интервал.

Ограничения

Думите се изписват само с малки латински букви. Броят на зададените символи не надхвърля 100.

Пример

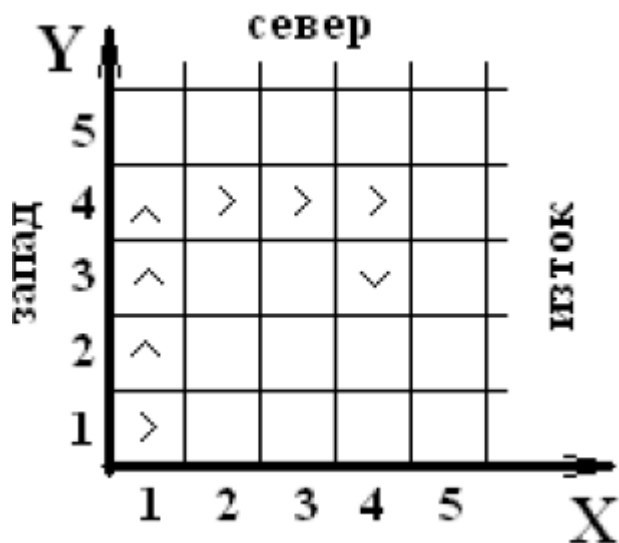
Вход

`oko banan liniq more`

Изход

`5 banan`

39. Робот е насочен в една от посоките изток (I), запад(Z), север(S) или юг (J). Подава му се последователност от команди: дясно (D), ляво (L), кръгом (K) и напред (N). При „дясно“ и „ляво“ той се обръща надясно или наляво, при „кръгом“ – на 180°. Тези команди се изпълняват в квадратчето, в което се намира. След подаване на командата „напред“, роботът се придвижва едно квадратче в посоката, в която гледа. В началото роботът е в колона 1 и в ред 1 ($X=1$ и $Y=1$) и командите се подават така, че по всяко време координатите му са положителни числа.

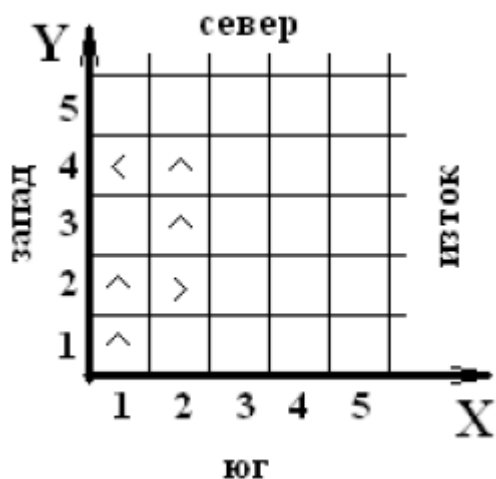


На картинката: Роботът гледа първоначално на изток и изпълнява следната поредица от команди **LNNNDNNNDN**. След първата команда **L** („ляво“), той се обръща наляво, вследствие на което ще гледа на север. Следващата команда е **N** („напред“), и той се премества в квадратчето на колона 1 и ред 2 (1,2). След изпълнение на следващите две команди **N** той ще се намира в (1,4). Командата **D** ще изпълни в квадратче (1,4) и след като се обърне надясно, ще гледа в посока „запад“ и т.н. Дадена е посоката P1 в която роботът гледа първоначално. След изпълнение на всичките M команди, той ще е в квадрат с координати (XM, YM) и ще гледа в посока P2. Вашата задача е да напишете програма **robo**, която извежда:

1. Координатите на квадратчето (XM, YM), в което ще е роботът след изпълнение на всички команди
2. Посоката P2, в която гледа роботът, след изпълнение на всички команди
3. Минималния брой команди, които могат да заменят дадените, така че, след изпълнението им, роботът тръгвайки от (1,1), където гледа в посока P1, да стигне в квадратчето (XM, YM) и да гледа в посока P2.

Вход

От първия ред на стандартния вход се въвежда посоката в която първоначално гледа роботът. От втория ред се въвежда символен низ, който съдържа поредица от най-много 100 команди.



Изход

На първия ред на стандартния вход трябва да се изведат стойностите на XM, YM и P2, разделение с по един интервал. На втория ред да се изведе минималният брой команди от т.3.

Пример 1 Пример 2:

Вход Вход

IS
LNNNDNNNDN NDNLNNLN

Изход Изход

4 3 J 1 4 Z
7 4

Оценяване

Решенията ви ще бъдат тествани върху 50 теста. За всеки тест ще получите:

- 1 точка за правилно определени стойности на XM, YM и P2;

40. Малкият Петърчо иска да обиколи всички полета на дъската с последователни ходове на коня, без да повтаря поле, в което вече е бил. Той намерил няколко решения за дъска с размери 3 x 4:

1	4	7	10
8	11	2	5
3	6	9	12

1	4	7	10
12	9	2	5
3	6	11	8

10	7	2	5
1	4	9	12
8	11	6	3

Намерете решение за дъска с размери 4 x 5:

Поставете в полетата числата 1, 2, ..., 20, така че от полето с числото 1 с ход на коня да се отива на полето с числото 2, от там в полето с числото 3 и т.н.. Резултата изкарайте във файл с име peter.txt

41. Напишете програма **WORDS**, която чете последователност от думи и съставя възможно най-дългата редица от думи, за която всяка следваща дума започва с последната буква на предходната (*new - word - door ...*), като ако съществува такава дума, то тя е единствена. Всяка дума може да се използва само по веднъж.

Вход

От първия ред на words.txt се въвежда *n* – брой думи. На всеки от следващите *n* реда се въвежда по една дума.

Изход

На първия ред на solution.txt се извежда число *b* - броя на думите в редицата и на всеки от следващите редове по една дума от нея.

Ако съществуват две редици с еднаква дължина се извежда тази, чиято първа дума е по-напред по азбучен ред.

Ако липсват думи, отговарящи на условието се извежда 0.

Ограничения

$0 < n \leq 26$

максимална дължина на дума – 10 букви

ПРИМЕР

Вход

5
bear
new
door
sky
word

Изход

3
new
word
door