

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача А. Островна мрежа

Архипелаг се състои от няколко групи острови, като за всеки главен остров на група от няколко острова е осигурен достъп до Интернет. Сега е време да се достави Интернет и до по-малките острови на всяка група, като те бъдат свързани с главния остров с кабели. Компанията, която полага кабелите би искала да минимизира разходите за построяване на мрежата, като цената за полагане на кабел от един остров на групата до друг е равна на разстоянието между двата острова в километри. Полагането на кабелите започва **едновременно** и с еднаква скорост от един километър на месец. Така за всеки остров е определено времето t_i , необходимо за да получи достъп до Интернет.

За всеки остров е известен броят m_i на жителите на острова, които ще получат достъп до Интернет. Компанията доставчик иска да пресметнете усреднена оценка за необходимото за реализация на проекта време по формулата

$$\frac{\sum m_i t_i}{\sum m_i}.$$

Вход.

На стандартния вход ще бъдат зададени данните за всички групи острови на архипелага. На първия ред с данни за всяка група ще бъде зададен броят n на островите в групата, последван от n реда с данните за всеки остров – координатите x_i и y_i на острова в километри и броят m_i на жителите на острова. Данните за главния остров са на първия ред. След данните за последната група ще има ред с нула.

Ограничения.

$$3 \leq n \leq 100, 100 \leq m_i \leq 5000, 0 \leq x_i, y_i \leq 1000$$

Изход.

За всяка група от острови програмата трябва да изведе на стандартния изход ред, започващ с поредния номер на групата от входа (броен от 1) и осреднената оценка за необходимото време, с две цифри след десетичната точка.

Пример:

Вход.

Изход.

5
2 7 1000
2 10 200
2 15 200
7 2 500
2 10 100
0

1 2.72

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача В. Подредици

Дадена е редица от n цели числа. Да се провери дали дадено число може да се получи като сума на членовете на някоя подредица на дадената редица.

Вход.

На стандартния вход за всеки пример са зададени: дължината на редицата n , числата от тази редица, броя на числата за проверка m и самите числа за проверка.

Ограничения.

$$2 \leq n, m \leq 100$$

Елементите на редицата и числата за проверка са в интервала $[0, 100000]$.

Изход.

За всеки пример на стандартния изход на един ред да се изведе редица, състояща се от m низа "yes" и "no" съответстваща на резултата дали съответното число за проверка може или не може да се получи като сума на членовете на някоя подредица на дадената редица.

Пример:

Вход.

Изход.

4

yes no yes

2 4 5 9

3

15 10 5

Обяснение:

За подредицата 2, 4, 9 получаваме $2 + 4 + 9 = 15$. Някоя подредица няма сума на членовете 10. Сума 5 се получава от подредица с единствен член 5.

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача С. Ша)(и Мат

Хари Каспари решил да играе шах, но когато взел шахматната си дъска, какво да види — синът му написал във всяка клетка по едно число. Хари запазил самообладание и вместо да се ядоса, просто си измислил нова игра. Решил да замени всяко просто число на дъската с отваряща скоба — (, а всички останали със затваряща —). Все пак, за да въведе правила в играта, решил да спазва следните ограничения:

1. В играта участва само една фигура — кон;
2. Задава се начална позиция на коня на дъската — номер на ред и номер на колона, където е поставен;
3. Задава се крайна позиция на коня на дъската — номер на ред и номер на колона, където е трябва да стигне;
4. Коня продължава да се движи така, както в шаха, т.е. по буква Г, до достигане на крайната позиция , без да стъпва втори път на едно и също поле от шахматната дъска;
5. От значение за играта са само клетките върху, които конят е бил поставен (е „стъпил“).

Целта на играта му била, да определи дали като се движи коня по дъската може да „събере“ такава последователност от скоби, че да се образува правилен израз от затварящи и отварящи скоби. Правилен ще наричаме израз от вида (), ((() ())), а неправилен например) (), ((), (() (, т.е. колкото отварящи скоби има, толкова съответни на тях затварящи. Търсенето на израз да спира, когато коня стигне желаната позиция на дъската.

Напишете програма, която по въведена матрица a_{ij} ($N \times N$) от цели числа образува дъската на Хари Каспари, като заменя простите числа с отваряща скоба (, а останалите със затваряща), и определя дължината на най-дългия правилен израз от скоби, които са се събрали при всички възможни ходове на коня по най-късия път от дадена позиция (M_1, N_1) , до друга дадена позиция (M_2, N_2) , където M_1 е номера на реда, а N_1 е номера на колоната, както и крайна позиция на движение M_2 и N_1 .

Вход.

За всеки тестов пример на първия ред от стандартния вход се задава число N — размерността матрицата, на следващия ред числата M_1 и N_1 — номер на ред и номер на колона за начална позиция на коня. На третия ред са числата M_2 и N_2 — номер на ред и номер на колона за крайна позиция на коня върху дъската. На следващите N реда са редовете на матрицата — цели числа, разделени с интервал. Входът съдържа много примери.

Ограничения.

$$3 \leq N \leq 13;$$

$$1 \leq N_i, M_i \leq N, i = 1, 2;$$

$$0 \leq a_{ij} \leq 10^5, i, j = 1, 2, \dots, N.$$

Изход.

За всеки тестов пример на стандартния изход на отделен ред да се изведе дължината на най-дългия правилен израз от всички скоби натрупани от всички възможни ходове по пътя на коня. Ако задачата няма решение, да се отпечата числото 0.

Пример:

Вход.

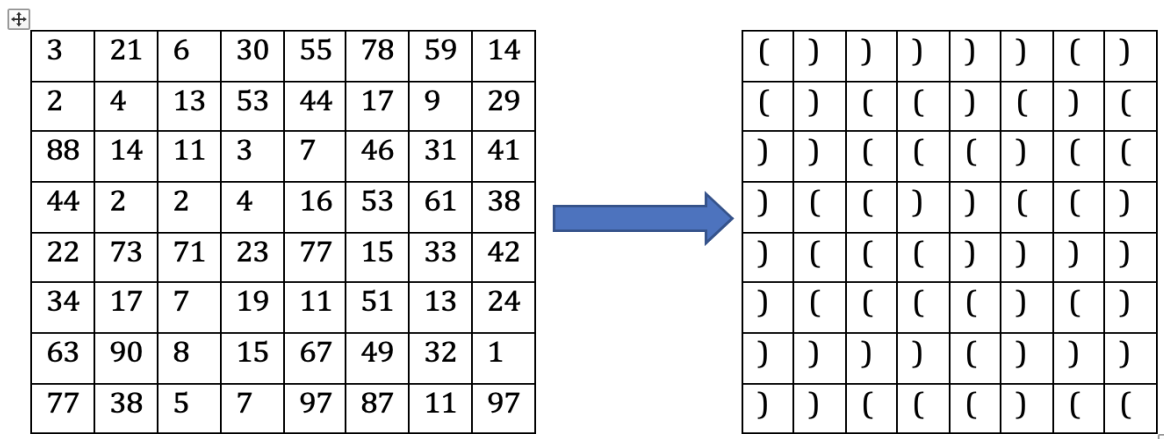
Изход.

```

8
7 1
8 3
3 21 6 30 55 78 59 14
2 4 13 53 44 17 9 29
88 14 11 3 7 46 31 41
44 2 2 4 16 53 61 38
22 73 71 23 77 15 33 42
34 17 7 19 11 51 13 24
63 90 8 15 67 49 32 1
77 38 5 7 97 87 11 97
    
```

12

Обяснение на примера.



Изразът, който се формира е $(((((())())()))(($, а най-дългият правилен израз в него е $(((((())())()))$ и е съставен от 12 символа. Коня е минал през следните позиции: (7, 1), (5, 2), (6, 3), (8, 3), (3, 1), (3, 3), (4, 4), (6, 4), (7, 3), (4, 2), (5, 1), (8, 2), (5, 5), (7, 5), (8, 4).

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача D. Пресичания

Дадени са хоризонтални и вертикални отсечки в равнината с целочислени краища. Да се намери общия брой на точките на пресичане на хоризонтална с вертикална отсечки.

Вход.

На стандартния вход за всеки пример са зададени: броя на отсечките и координатите на техните краища. Входът съдържа много примери.

Ограничения.

Броят на отсечките е по-малък от 2000.

Координатите на отсечките са цели числа в интервала $[-10^9, 10^9]$.

Дължините на отсечките са по-големи от 0.

Изход.

За всеки пример на стандартния изход на отделен ред да се изведе търсения брой.

Пример:

Вход.

Изход.

4
0 0 5 0
1 1 1 -1
0 2 5 2
-1 0 3 0
2
0 0 0 1
0 0 1 0

1
1

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача Е. Прозорци

Даден е правоъгълен екран от пиксели с $M + 1$ реда и $N + 1$ стълба, номерирани от 0 до M и от 0 до N , съответно. Прозорец е правоъгълник от пиксели, с горен ляв ъгъл с координати (x, y) , с r реда и c стълба, разположен изцяло на екрана. Прозорците, които се разполагат на екрана, не могат да имат общи пиксели. Напишете програма, която да управлява разполагането на прозорците на екрана. В началото на екрана няма никакви прозорци. Програмата трябва да може да изпълнява следните команди:

- $O\ x\ y\ r\ c$ – отваря на екрана прозорец с параметри $x\ y\ r$ и c ;
- $C\ x\ y$ – заваря прозореца, който съдържа пиксела с координати (x, y) ;
- $R\ x\ y\ r\ c$ – променя размера на прозореца, съдържащ пиксела с координати (x, y) , до r реда и c стълба;
- $M\ x\ y\ r\ c$ – премества прозореца, съдържащ пиксела с координати (x, y) вертикално на r пиксела или хоризонтално на c пиксела, където точно едно от двете r и c е различно от 0. При положително c изместването е надолу, при отрицателно — нагоре. При положително r изместването е надясно, а при отрицателно — наляво.

При изпълнението на операциите може да възникнат изключения, за които програмата трябва да съобщи във вида

<номер на команда> <вид на командата> <вид на изключението>

като номерирането на командите за всеки тестов пример е от едно, видът е O, C, R или M, а възможните изключения са:

- **Doesn't exist** — когато в команда C, R или M съответният прозорец не е на екрана;
- **Can't fit** — когато в команда O, R или M резултатът от операцията не може да се разположи на екрана — част от прозореца излиза извън него или застъпва друг прозорец.

Вход.

Програмата трябва да реши няколко тестови примера при едно извикване. На първия ред на стандартния вход за всеки тест ще бъдат зададени параметрите M и N на екрана и броят на командите Q . На всеки от следващите Q реда ще бъде зададена по една команда. След последния тест ще има ред с три нули.

Ограничения.

$$100 \leq M \leq 1000, 100 \leq N \leq 1000, 5 \leq Q \leq 100$$

Изход.

За всяка команда, която не може да бъде изпълнена програмата трябва да изведе на стандартния изход съответното съобщение за изключение, а след изпълнение на всички команди от един тест – броя на останалите отворени прозорци.

Пример:

Вход.

```
320 200 12
0 -5 -5 50 50
0 50 50 10 10
R 60 60 300 300
0 70 55 10 10
0 90 50 10 10
R 70 60 40 40
R 55 55 40 40
R 55 55 15 15
M 0 0 40 0
C 55 55
C 300 200
M 95 55 0 -100
0 0 0
```

Исход.

```
1 0 Can't fit
3 R Can't fit
9 M Doesn't exist
11 C Doesn't exist
12 M Can't fit
2
```

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача F. Скакалец

Дадени са 4 цели числа: k, d_1, d_2, n и редица от числа с дължина n . Скакалец се намира на целочислената права, застанал в точката на числото k . От тази позиция започва да прави скокове по целочислената права с дължини d_1 или d_2 в двете посоки, т.е. може да отиде в точките $k + d_1, k + d_2, k - d_1$ и $k - d_2$. Напишете програма, която намира кои точки от дадената редица са достъпни за скакалеца с произволен брой скокове.

Вход.

За всеки пример на стандартния вход са дадени по два реда. На първия ред са числата k, d_1, d_2 и n , разделени с по един интервал, а на втория — редицата от числа, отново разделени с интервал. Входът съдържа много примери.

Ограничения.

$$2 \leq n \leq 700$$

$$1 \leq k, d_1, d_2 \leq 1000$$

Елементите на редицата в интервала $[-1000, 1000]$.

Изход.

За всеки пример от стандартния вход на стандартния изход се извежда редица от нули и единици (разделени с по един интервал), която показва кои числа от входната редица са достъпни за скакалеца.

Пример:

Вход.

10 4 6 3

10 15 20

8 3 2 2

9 4

10 4 6 10

12 33 40 100 -200 201 1000 2 4 6

Изход.

1 0 1

1 1

1 0 1 1 1 0 1 1 1 1

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача G. Сигнали

При предаване на двоични низове по канал за предаване на данни се редуват единици и нули. Ако в низ от единици и нули преобладават единиците, които в канала са кодирани с по-високо напрежение от нулите, тогава такъв низ наричаме тежък, защото може да претовари апаратурата. Задачата ви е, по зададен двоичен низ, да се намери броя на тежките поднизове.

Вход.

На ред на стандартния вход ще бъде зададен T тестови случая. Всеки тестов случай се състои от ред, съдържащ двоичен низ и нека с N означим дължината на низа.

Ограничения.

$$4 \leq N \leq 1000000$$

Изход.

За всеки тест, на отделен ред на стандартния изход, програмата трябва да изведе броя на тежките поднизове на низа.

Пример:

Вход.

Изход.

2

4

0011

5

0110

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача Н. Джип

Родителите на Крис му купиха първата кола: един чудовищен джип! Крис разбра, че макар да има кола, която може да смачка всички останали автомобили, паркирането на автомобил с размерите на 4 нормални коли може да бъде доста сложно. Приятелят му Боби работи във фирмата, която се грижи за градските паркинги. Той периодично изпраща на Крис карта на града им с обозначените места за паркиране, на която са маркирани вече заетите места. Картата може да бъде представена като таблица с R реда и C колони. Всяка клетка от таблицата може да съдържа сграда (означена със символа „#“), паркиран автомобил (означен със символа „X“) или безплатно място за паркиране (означено със символа „.“). Чудовищният джип на Крис е огромен и заема при паркиране квадрат 2×2 клетки. Помогнете на Крис да се ориентира за броя на възможните места, на които може да паркира, групирани по броя на колите, които трябва да се преместят, за да паркира на дадено място. Интересува ни само броя автомобили, които трябва да се преместят от място за паркиране, а не броя на колите, които ще трябва да се отместят по пътя на джипа, за да достигне мястото. Крис не може да паркира върху сграда, както и не може да преминава през сграда.

Вход.

От първия ред на стандартния вход се въвежда едно цяло число N – броя на тестовите случаи. Първият ред на всеки тест съдържа две цели числа, R и C , броят на редовете и колоните на картата. Всеки от втория до R -тия ред на теста съдържат по C знака, които могат да съдържат само символите „#“, „X“ и „.“. „X“ е главна латинска буква.

Ограничения.

$$2 \leq R, C \leq 50$$

Изход.

За всеки тестов случай на стандартния изход програмата трябва да изведе по пет реда: общия брой паркоместа, на които може да паркира Крис, ако трябва да се отместят 0 коли (на първия ред), 1 кола (на втория ред), 2 коли (на третия ред), 3 коли (на четвъртия ред), 4 коли (на петия ред).

Пример:

Вход.

Исход.

3	1
4 4	1
#..#	2
..X.	1
..X.	0
#XX#	9
4 4	0
....	0
....	0
....	0
....	2
4 5	1
..XX.	1
..#XX.	0
..#..	1
.....	

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

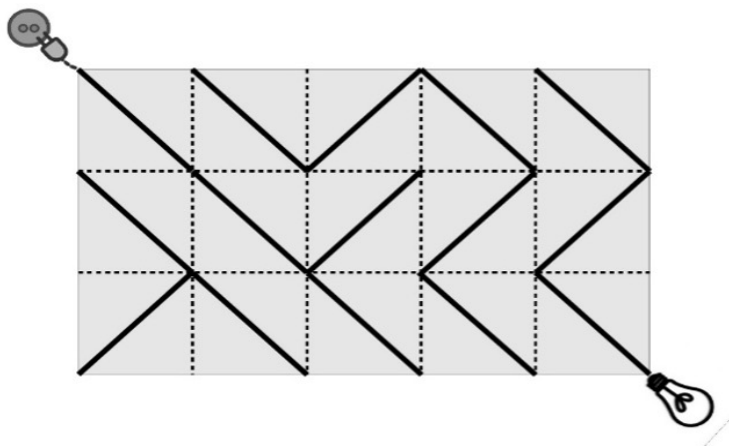
Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача I. Включете лампата

При едно от посещенията си в Националния политехнически музей, Петър попада на интересен експонат. В правоъгълна рамка са подредени плътно една до друга квадратни теракотни плочки. Те образуват правоъгълна решетка, която има N реда и M стълба. Два (от четирите) противоположни ъгъла на всяка плочка са свързани с проводник. Източник на електрическо захранване е свързан към горния ляв ъгъл на рамката. Към долния десен ъгъл на рамката е свързана лампа. Лампата свети само тогава, когато има път от проводници, свързващи източника на електрическо захранване с лампата.



На Фигурата, показана по-горе, лампата е изключена. За да светне лампата, някои от плочките трябва да бъдат завъртани на 90° (без значение в коя от двете посоки), така че проводниците да образуват път от електрическия контакт до лампата. Ако коя да е от плочките, от четвъртата колона на Фигурата бъде завъртяна на 90° , източникът на захранване и лампата ще се свържат и лампата ще светне. Петър се опитва да намери минималния брой завъртания на плочки за светване на лампата. Помогнете му, като напишете програма `lamp`, която използвайки конфигурацията на плочките да определи минималния брой завъртания на 90° за включване на лампата.

Вход.

Първият ред на стандартния вход съдържа едно цяло число T , което задава броя на тестовете. Първият ред на всеки тест съдържа две цели числа N и M — броя на редовете и колоните плочки в рамката. Във всеки от следващите N реда са разположени плътно един до друг символа — \setminus или $/$, които показват посоката на проводника, който свързва срещуположните върхове на съответната плочка.

Ограничения.

$$1 \leq N, M \leq 500$$

Изход.

За всеки тест на един ред на стандартния изход трябва да се изведе едно цяло число: минималния брой плочки, които трябва да се завъртят на 90° , за да се включи лампата. Ако това е невъзможно, да се изведе „Impossible“.

Пример:

Вход.

Изход.

2
3 5
\\/\n
\\///
/\n\n
3 3
///
/>\n
///

1
3

Забележка: Първият тест съответства на Фигурата.

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача J. Зигзаг

Една крайна редица от цели числа се нарича зигзаг, ако всеки елемент на редицата, без първия и последния, е или по-голям от двата му съседа или по-малък от двата съседни елемента. Да се напише програма за определяне най-дългата част в дадена редица, която е зигзаг.

Вход.

На стандартния вход се задават числови редици – всяка на отделен ред с разделител един интервал между числата.

Ограничения.

Всички числа са в интервала $[-1000, +1000]$.

Дължината на редицата е най-малко 3 и най-много 10^6 .

Изход.

За всяка редица на стандартния изход се извежда на отделен ред дължината на най-дългата зигзаг част от дадената редица.

Пример:

Вход.

Изход.

2 30 2 3 2 3 2 2	7
1 2 3 4 5 6 7 81	0
1 1 1 2 1 1 3 2 3 3	4
1 2 2 1	0

Обяснение на примера:

2 30 2 3 2 3 2 е зигзаг редица.

Няма част, която да е зигзаг редица.

1 2 1 и 1 3 2 3 са зигзаг редици.

Няма част, която да е зигзаг редица.

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача К. Вход-изход

Дадена е крайна редица от низове, които се получават от софтуерна система, обработват се и се изпращат. Дължината на низа определя за колко такта на системата се обработва този низ, т.е. след колко такта на системата може да бъде изпратен. На всеки такт в системата постъпва нов низ и може да бъде изпратен вече обработен от системата низ. Ако няколко низа са готови за изпращане, изпраща се най-късия, при повече от един най-къси се избира по-малкия в лексикографската наредба.

Вход.

На стандартния вход се задава редица от низове – постъпващите на всеки такт низ в системата.

Ограничения.

Низовете са образувани от малки латински букви и не са по-дълги от 100 букви.

Изход.

На стандартния изход за всеки входен низ се извежда по един ред, съдържащ входния низ и низът, който се изпраща на този такт. Ако няма готов за изпращане низ се печата -. Изходът завършва, когато в системата няма нови постъпващи низове и не останат низове за изпращане. Точният формат на изхода е даден в примера по-долу.

Пример:

Вход.

Изход.

axxxx xb abc da bb c

```
1 in:axxxx out:-
2 in:xb out:-
3 in:abc out:-
4 in:da out:xb
5 in:bb out:-
6 in:c out:da
7 in:- out:c
8 in:- out:bb
9 in:- out:abc
10 in:- out:axxxx
```

Обяснение на примера:

На първия такт постъпва низ axxxx, който ще се обработва от системата 5 такта, следователно ной-ранното му изпращане е на такт 6. Няма готови низове за изпращане.

На втория такт постъпва низ xb, който ще се обработва от системата 2 такта, следователно ной-ранното му изпращане е на такт 4. Няма готови низове за изпращане.

На третия такт постъпва низ abc, който ще се обработва от системата 3 такта, следователно ной-ранното му изпращане е на такт 6. Няма готови низове за изпращане.

На четвъртия такт постъпва низ da, който ще се обработва от системата 2 такта, следователно ной-ранното му изпращане е на такт 6. Изпраща се низ xb.

На петия такт постъпва низ bb, който ще се обработва от системата 2 такта, следователно ной-ранното му изпращане е на такт 7. Няма готови низове за изпращане.

На шестия такт постъпва низ с, който ще се обработва от системата 1 такт, следователно най-ранното му изпращане е на такт 7. Готови за изпращане са низовете: ахxxx, abc и da. Изпраща се най-късия – da.

На седмия такт няма постъпващ низ. Готови за изпращане са низовете: ахxxx, abc, da и с. Изпраща се най-късия – с.

На следващите тактове няма постъпващи низове, всички низове в системата ахxxx, abc и da са готови. Изпращат се последователно според дължините им.

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Департамент *Информатика*

XXXIII Републиканска студентска олимпиада по програмиране

8 май 2021 г.

Задача L. Общ подниз

Дадени са два низа p и q , образувани от малки латински букви. Да се намери дължината на най-дългия общ подниз, който е по-дълъг от зададена стойност k .

Вход.

На стандартния вход за всеки пример на един ред се задават низовете p и q и числото k .

Ограничения.

Числото k по-голямо от една десета от дължината на по-дългия низ и по-малко или равно на дължината на по-късия низ.

Дължината на низовете е в интервала $[2, 20000]$.

Изход.

На стандартния изход за всеки пример на отделен ред да се отпечати дължината на общия подниз. Ако тази дължина е по-малка от k , да се отпечати 0.

Пример:

<i>Вход.</i>	<i>Изход.</i>
abcd ab 2	2
qwerty eeertyuu 3	4
xz hey 2	0
xuzzz xxxyz 4	0

Обяснение на примера:

Общият подниз ab е с дължина $2 = 2$.

Общият подниз $erty$ е с дължина $4 > 3$.

Няма общ подниз.

Общият подниз xuz е с дължина $3 < 4$.