



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 ноември 2014 г.

А. Придвижване

Дадена е таблица с размери N на N , съдържаща символите $+$ (плюс) или $-$ (минус). От дадена клетка може да се придвижвате в съседна на нея (на изток, запад, север или юг), като времето ви за придвижване е равно на A , ако двете клетки са с еднакъв символ и B - в противен случай. Да приемем, че винаги се движите оптимално, т.е. за най-кратко време, от всяка клетка, до всяка друга. Интересуваме се кое е най-лошото възможно време, измежду всички тези времена.

На първия ред на стандартния вход е зададен броят на тестовите примери. На първия ред на всеки от тях са зададени числата N ($1 \leq N \leq 30$), A ($0 \leq A \leq 1000000$) и B ($0 \leq B \leq 1000000$). Следват N реда, с по N символа $+$ или $-$.

За всеки тест на отделен ред на стандартния изход изведете търсеното време.

Вход	Изход
1 3 1 2 +++ +-+ ++-	5

Пояснение към примера: Най-лошото време е при придвижване от горния ляв ъгъл до долния десен и то е 5.



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 ноември 2014 г.

В. Среднощно кодене

Освен всичко друго Панко е и луд програмист, коди и денем и нощем! Той трябва да напише програма, която се състои n от реда код. Нощем силите му намаляват, за това той действа по следния начин: първоначално пише v реда код, след което изпива една бира! След това пише още $\lfloor \frac{v}{k} \rfloor$ реда, пие нова бира, после пише $\lfloor \frac{v}{k^2} \rfloor$ реда, нова бира и т.н: $\lfloor \frac{v}{k^3} \rfloor, \lfloor \frac{v}{k^4} \rfloor, \lfloor \frac{v}{k^5} \rfloor, \dots$

Израза $\lfloor \frac{a}{b} \rfloor$ се изчислява, като резултат от целочисленото делене на a и b .

В момента, в който $\lfloor \frac{v}{k^p} \rfloor$ стане равно на 0, Панко моментално заспива и се събужда чак на сутринта, когато програмата вече трябва да е готова.

Панко се чуди, кое е най-малкото целочислено v , такова че той да може да напише не по-малко от n реда код.

Вход

Всеки тестов пример е зададен на отделен ред на стандартния вход и се състои от целочислените n и k , $1 \leq n \leq 10^9$, $2 \leq k \leq 10$. Края на входа е маркиран с нула.

Изход

За всеки тестов пример извеждайте на отделен ред на стандартния изход – търсеното v .

Вход	Изход
7 2	4
59 9	54
0	

Пояснение към изхода: В първия тест отговора е 4. Панко пише код на съответните порции: първо 4 реда, след това 2, след това 1 ред и после заспива. Той успява да напише $4 + 2 + 1 = 7$ реда код, с което задачата му е изпълнена. Във втория тест първоначално написва 54 реда, след което още 6 и заспива. Той дори е успял да напише 1 ред повече от планираното, т.е. преизпълнил е плана. Написал е $54 + 6 = 60$ реда, а е бил планирал да напише 59. Браво, Панко!



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 ноември 2014 г.

С. Шапки долу!

N ($2 \leq N \leq 100000$) човека са разположени на различни позиции на числовата права. i -я човек е на целочислена позиция x_i ($0 \leq x_i \leq 1000000000$) и може да има шапка на главата, а може и да няма. Няма двама на една и съща позиция.

Иска се да намерите максималната последователност от хора, в която броят на тези с шапки е равен на тези без шапки. Дължината на максималната последователност е разликата в минималната и максималната позиции на хората, участващи в последователността. Имате възможност да поискате от хора с шапки да ги свалят, ако считате, че това ще максимизира резултата ви.

На първия ред на стандартния вход е зададен броят на тестовите примери. Всеки от тях започва с броя на хората – N . Следват N реда, като всеки ред съдържа позиция, на която е разположен човек и символа W или S . W показва, че човека е с шапка, а S – човека е без шапка.

За всеки тест извеждайте по една число на отделен ред – дължината на търсената последователност.

Вход	Изход
1 5 8 W 11 S 3 W 10 W 5 S	7

Пояснение към изхода: Максималната последователност от хора започва от позиция 3 и завършва в позиция 10. В нея трима човека са с шапки, а един – без. Ако помолите някой от тримата с шапки да си я свалят, то ще получите равен брой хора с и без шапки.



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 ноември 2014 г.

D. Геокешинг

Панко вече си има GPS и веднага тръгнал в планината да търси първия си кеш. Преходът му се сторил изключително тежък, но за отказване и дума да не става! Няма как да не се похвали на приятелите си за приключението, в което се е вкарал. За това решил на всяка минута да си записва надморската височина (в метри) на мястото, където се намира. След като погледнал редицата от записани стойности, решил да намери колко минути е най-дългият интервал, когато непрекъснато се е изкачвал. Това число е неговата гордост! Помогнете му, като напишете програма, която пресмята това.

Вход

Всеки тестов пример е зададен на отделен ред на стандартния вход. Първото число задава броя на записаните от Панко числа в редицата. Следват записаните стойности на височини. Между всеки две числа има по един интервал. Края на входа е 0.

Изход

За всеки тест - едно цяло число, равно на търсената дължина в минути, изведено на отделен ред на стандартния изход.

Ограничения

Броят на записаните числа не е по-голям от 1000.

Всяко от записаните числа е цяло с възможни стойности от 0 до 1000.

Вход	Изход
13 1 2 3 2 3 1 5 6 7 8 9 2 3	6
3 123 123 123	1
8 1 2 1 2 1 2 1 2	2
0	



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 ноември 2014 г.

Е. Оцветяване

Даден е граф с N ($1 \leq N \leq 50000$) върха, номерирани за удобство с числата от 1 до N , свързани чрез M ($1 \leq M \leq 100000$) двупосочни ребра. Ребро i свързва връх A_i ($1 \leq A_i \leq N$) с B_i ($1 \leq B_i \leq N$), $A_i \neq B_i$. Възможно е две ребра да свързват една и съща двойка върхове.

Искате да оцветите върховете на графа с два различни цвята X и Y . Два върха трябва да бъдат оцветени с различен цвят, ако са свързани с ребро. Имате обаче предпочитание към цвят X и искате възможно най-много от върховете да са оцветени с този цвят. Намерете максималния брой върхове, които могат да бъдат оцветени с цвят X .

Всеки тестов пример започва с числата M и N . Следват M реда, на които са зададени числата A_i и B_i , указващи двупосочно ребро между двата върха.

За всеки тестов пример извеждайте на отделен ред търсения брой върхове или -1 , ако оцветяване, отговарящо на гореописаните правила не е възможно.

Вход	Изход
4 4 1 2 2 3 3 4 4 1 0 0	2

Пояснение към изхода: върхове 1 и 3 могат да бъдат боядисани с цвят X , алтернативно 2 и 4 също.



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 ноември 2014 г.

Ф. Състезания

Почти сме сигурни, че сте експерти в четенето на входа, но дали е така и с изхода? Разполагате със списък от участници в състезания. Всеки запис се състои от името на участника S (уникална последователност от малки и/или големи латински букви, не по-дълга от 10 символа), броя решени задачи - R, както и времето за което са решени задачите – T ($0 \leq R, T \leq 10^4$).

Напишете програма, която извежда K-те най-добре представили се студенти на поредното състезание. Класирането става първо по брой решени задачи. При равен брой такива се гледа времето, за което са решени и ако и то е еднакво, то тогава по-напред в класирането трябва да се появи участника, чието име е по-напред в лексикографската подредба.

Програмата ви трябва да обработи няколко тестови примера. Всеки от тях съдържа по-малко от 100 реда, описващи резултатите на всеки от състезателите, участвали в поредното състезание. На i-я ред са зададени R_i , T_i и S_i , разделени с интервал. На последния ред е зададено K.

Първия ред на изхода за всеки тест трябва да съдържа “Summary for contest C (top K of N participants):”, където C е номера на поредния тестов пример. Следва празен ред, последван от класацията на състезателите. Символите използвани за визуализирането на таблицата са '-', '#', '#' и '|' (ASCII кодове 45, 35 и 124). Първата колона показва номера на съответния състезател в класирането, втората – името му, третата – броя задачи, които е решил, а четвъртата – времето, за което ги е решил. Числата в първата колона се разполагат в две позиции (на една позиция може да седи точно един символ) и са дясно подравнени. Имената на състезателите се разполагат в колона с десет позиции и трябва да са ляво равнени. В последните две колони числата са разположени на пет позиции и са дясно равнени. Имената на всички колоните са дясно равнени. Броят символи '-' на всеки ред трябва да е точно 27. Всеки два теста трябва да са разделени с по един празен ред. Изхода трябва да изглежда точно така, като е показано:

Вход	Изход
4 256 Drago 2 387 Yasen 2 52 Spaska 1 9 Deqn 2 387 Bobo 0 0 Toni 10 788 Panko 5 891 Stefan 2 387 Asen 1 12 Denis 0 0 Pencho 10	Summary for contest 1 (top 10 of 11 participants): ----- # name tasks time ----- 1 Panko 10 788 2 Stefan 5 891 3 Drago 4 256 4 Spaska 2 52 5 Asen 2 387 6 Bobo 2 387 7 Yasen 2 387 8 Deqn 1 9 9 Denis 1 12 10 Pencho 0 0 -----



Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 22 ноември 2014 г.

G. Израз

Да се напише програма, която пресмята аритметични изрази. За всеки пример програмата чете аритметичен израз - редица от реални числа и аритметичните операции събиране, изваждане, умножение и деление и отпечатва резултата на нов ред на изхода. Програмата трябва да спазва приоритета на операциите.

Входът съдържа повече от един пример и започва с цяло число, което оказва броя на примерите.

Резултата от пресметнатия израз да се извежда на отделен ред на стандартния изход.

Вход	Изход
2	11
1+2*5	0.5
4-7*2+10.5	



Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 ноември 2014 г.

Н. Метро

Панко започна да пътува и с метрото! Обикновен билет за едно пътуване струва a лева. Специален билет за m пътувания струва b лева. Помогнете на Панко да си купи билети за общо n пътувания на възможно най-ниска обща цена.

Всеки тест е зададен на отделен ред на стандартния вход и се състои от четири цели положителни числа - n, m, a, b ($1 \leq n, m, a, b \leq 1000$).

Изхода за всеки тест е търсената цена, която трябва да се извежда на отделен ред на стандартния изход.

Вход	Изход
6 2 1 2	6
5 2 2 3	8



I. Морски шах

Играта морски шах се играе от двама играчи на квадратна дъска с размери 3 на 3. Единия играч използва X, за да маркира своите ходове, а другия – 0. Първи винаги е играча с X. Двамата играчи се редуват и този, които успее да постави три свои символа по хоризонтал, вертикал или диагонал печели играта. Ако дъската е попълнена, но няма разположени три еднакви символа, както се изисква, то няма победител.

По зададена 3 x 3 дъска, всяка клетката на която съдържа точка, хикс или нула, трябва да определите кой играч е на ход. Извеждайте **first** за първи играч и **second** за втория или някое от следните:

- **illegal** – ако дъската няма как да бъде получена при валидни ходове на двамата;
- **the first player won** – ако първия играч току-що е спечелил играта;
- **the second player won** – ако втория играч току-що е спечелил играта;
- **draw** – ако играта е приключила и няма победител;

Първия ред на входа съдържа броя на тестовите примери. Всеки от тях се състои от три реда, всеки с по три символа измежду ".", "X" или "0" (ASCII кодове 46, 88 и 48).

За всеки тест извеждайте на отделен един от шестте възможни резултата.

Вход	Изход
6	second
X0X	illegal
.0.	first
.X.	the first player won
0.X	the second player won
XX.	draw
000	
X.X	
X.0	
0.0	
XXX	
X00	
X00	
.XX	
000	
XX0	
X00	
0XX	
XX0	



Ж. Подмасиви

Подмасив $a[i...j]$ ($1 \leq i \leq j \leq n$) на масива $a = (a_1, a_2, \dots, a_n)$ е масив от последователните елементи на a , включващ елементите между i -та и j -та позиции: $a[i...j] = (a_i, a_{i+1}, \dots, a_j)$.

Интересуваме се от броя на всички подмасиви на a , в които има поне k еднакви числа.

На първия ред на всеки тест са зададени n и k ($1 \leq k \leq n \leq 4 \cdot 10^5$). На следващия ред са зададени n -те цели a_i ($1 \leq a_i \leq 10^9$) – елементите на масива. Входа завършва с нула.

Търсения брой за всеки тест да се извежда на отделен ред на стандартния изход.

Вход	Изход
4 2	3
1 2 1 2	2
5 3	6
1 2 1 1 3	
3 1	
1 1 1	
0	

Пояснение към примерите: В първия пример трите подмасива, съдържащи поне 2 еднакви елемента са $(1, 2, 1)$, $(2, 1, 2)$ и $(1, 2, 1, 2)$. Във втория пример подмасивите с поне 3 еднакви числа са $(1, 2, 1, 1, 3)$ и $(1, 2, 1, 1)$, а в третия тези с поне 1 са (1) , (1) , (1) , $(1, 1)$, $(1, 1)$ и $(1, 1, 1)$.