

ЧЕТВЪРТО КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

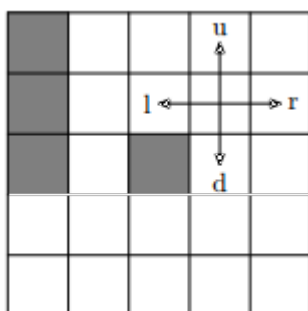
ПЛОВДИВ, 10 юни, 2018 г.

Група А, 11-12 клас

Задача АК1. Хамелеон

Жълтото пате изучава новия графичен редактор „Хамелеон“. Този редактор е необикновено прост. Той поддържа точно два цвята - черен и бял, и един курсор - хамелеон.

Работното поле на редактора е квадрат с $N \times N$ клетки. В една от клетките на полето се намира курсорът-хамелеон. Курсорът може да се премества в рамките на полето в четири направления — нагоре, надолу, наляво или надясно точно по една клетка. Цветът на курсора винаги трябва да съвпада с цвета на клетката, в която се намира. За целта, когато той се премества в клетка от другия цвят, трябва да се случи едно от две събития: или курсорът променя цвета си на цвета на тази клетка, или обратно — клетката променя своя цвят на цвета на курсора. Например, ако курсорът се премества от черна клетка в бяла, трябва или той да се оцвети в бял цвят, или бялата клетка, в която той се намира, трябва да стане черна. Ако клетката и курсорът имат еднакъв цвят, то техният цвят не се променя.



Първоначално курсорът има черен цвят и се намира в горната лява клетка на полето. Тази клетка също е черна. Всички останали клетки на полето са бели.

Напишете програма, определяща последователност от действия на курсора-хамелеон, след изпълнение на които на полето ще се получи картинка, зададена във входните данни.

Вход

От първия ред на стандартния вход се въвежда числото N ($5 \leq N \leq 100$) — размерът на полето.

На следващите N реда се описва картинката, която трябва да се получи. Всеки ред от описанието на картинката има дължина N и се състои от знаци 'W', ако съответната клетка е бяла, и 'B', ако е черна.

От последния ред се въвежда номерът на теста.

Изход

На стандартния изход се извежда един ред съдържащ описанието на търсената последователност от действия.

За означаване на преместване вляво, нагоре, вдясно или надолу **с изменение на цвета на курсора** се използват буквите 'l', 'u', 'r' или 'd'. За означаване на преместване вляво, нагоре, вдясно или надолу **с изменение на цвета на клетката** се използват буквите 'L', 'U', 'R' или 'D' съответно. Ако курсорът се премества в клетка със своя цвят, може да се използва както главна, така и малка буква. Няма интервали между символите, означаващи преместванията на курсора.

Тестове и оценяване

В тази задача тестовите данни са достъпни за участниците. [Може](#) да ги свалите от системата. Тестовите са номерирани в съответствие с наименованията на файловете от 1 до 20. Всеки тест се оценява с максимум 5 точки. Тестовите се оценяват независимо.

Тестовите се приемат за успешно преминати, ако изведената последователност съдържа не повече от 5 000 000 действия и води до правилен резултат.

ЧЕТВЪРТО КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

ПЛОВДИВ, 10 юни, 2018 г.

Група А, 11-12 клас

Първите 10 теста се оценяват с 5 точки, ако тестът е успешно преминал.

Останалите 10 теста се оценяват по следния начин. Ако тестът е успешно преминал,

то:

- с 5 точки, ако отговорът съдържа не повече от $3N^2$ действия;
- с 4 точки, ако отговорът съдържа не повече от $5N^2$ действия;
- с 3 точки, ако отговорът съдържа не повече от $10N^2$ действия;
- с 2 точки, ако отговорът съдържа не повече от $2.5N^3$ действия;
- с 1 точка, ако отговорът съдържа не повече от 5 000 000 действия.

Пример

Вход	Изход
5 BWWW BWWW BWBWW WWWW W WWWW W 1	DDRRdlU

Визуализатор

За да можете по-добре да си представите вида на картинката, която трябва да получите или да проследите как се изпълнява последователността от действия, които е генерирала Вашата програма, журито Ви предоставя визуализатор. Той е реализиран във файл Visualizer.jar. Свалете този файл от системата и го запишете в някоя папка. Тези, които работят под Windows, просто трябва да стартират този файл. Работещите под Ubuntu трябва отворят Terminal, да отидат в папката, в която са записали Visualizer.jar и да дадат команда `java -jar Visualizer.jar`.

ЧЕТВЪРТО КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

ПЛОВДИВ, 10 юни, 2018 г.

Група А, 11-12 клас

Задача АК2. БЕНЗИНОСТАНЦИИ

Лятото приближава и Лора ежедневно мечтае за безгрижните дни на плажа. Понеже не ѝ се работи, тя прекарва дните си като планува пътувания до морето. Тъй като все пак неработенето не е особено доходоносно, тя иска да планува максимално евтини пътувания.

За простота нека си представим, че пътят, по който Лора ще пътува, е разположен на числовата права и е с обща дължина L , започвайки от 0 и завършвайки в L . Колата на Лора има резервоар, който има капацитет от K литра гориво, и е първоначално празен. По пътя има N бензиностанции, разположени на **различни** позиции. Бензиностанция номер i се намира на целочислена координата X_i и там може да се зарежда гориво на цена C_i на литър.

Лора изминава път с дължина 1, използвайки точно 1 литър гориво. Ако в даден момент тя е на едно и също място с бензиностанция, то може да зареди произволно *цяло* число литри гориво без да надхвърля капацитета на резервоара си.

Сега Лора иска да напишете програма, която да изготви план за това къде и колко да зарежда тя, така че да успее да стигне до морето на максимално ниска цена. За жалост обаче лятото не е толкова близо и до тогава могат да настъпят много промени. Затова програмата ви трябва да обработва Q на брой заявки, като всяка заявка се състои в отваряне на нова бензиностанция някъде на пътя. Гарантирано е, че винаги съществува начин Лора да измине разстояние L и в нито един момент няма да има две бензиностанции на една и съща позиция.

Първоначално, както и след всяка заявка, програмата трябва да пресметне цената на най-евтиното възможно пътуване.

Задача

Напишете функции *init()* и *addStation()*, които ще се компилират заедно с програмата на журито и, получавайки информация за първоначалния набор от бензиностанции, както и за всяка новооткрита бензиностанция, определят минималната цена, за която Лора може да стигне до морето – при първоначалния набор от бензиностанции и след добавянето на всяка бензиностанция.

Детайли по реализацията

Вие трябва да напишете и предадете към системата програмен файл **stations.cpp**, който съдържа функциите:

```
long long init(int L, int K, int N, int stationPositions[], int stationOilPrices[]) и  
long long addStation(int newStationPosition, int newStationOilPrice)
```

Функцията *init* се извиква само веднъж в началото на програмата и нейните параметри имат следния смисъл:

L – общата дължина на пътя до морето;

K – Капацитет на резервоара на колата на Лора в литри;

N – Брой на бензиностанциите в първоначалния набор;

stationPositions – масив с координатите на бензиностанциите от първоначалния набор. Индексирането започва от 0.

stationOilPrices – масив с цените на бензина в бензиностанциите от първоначалния набор. Индексирането започва от 0.

При извикването *init* трябва да върне минималната цена, за която Лора може да стигне до морето при първоначалния набор от бензиностанции.

Бележка: Не е задължително бензиностанциите да са подредени по нарастващ ред на координатите им.

ЧЕТВЪРТО КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

ПЛОВДИВ, 10 юни, 2018 г.

Група А, 11-12 клас

Функцията *addStation* се извиква Q пъти – при всяка заявка за добавяне на бензиностанция. Нейните параметри имат следния смисъл:

newStationPosition – координата на новата бензиностанция, която се добавя;

newStationOilPrice – цена за литър гориво в новата бензиностанция.

При всяко извикване *addStation* трябва да връща минималната цена, за която Лора може да стигне до морето при новия набор от бензиностанции, който се получава.

Файлт **stations.cpp** НЕ трябва да съдържа функция *main()*, но може да съдържа декларации и други функции, които са необходими за работата на *init* и *addStation*. В началото си този файл трябва да съдържа `#include "stations.h"`.

Ограничения

$$0 \leq C_i \leq 10^9$$

$$0 \leq X_i \leq L$$

Гарантирано е, че във всеки момент съществува решение.

Подзадачи

Задачата е разделена на подзадачи. За да вземете точките предвидени за дадена подзадача, трябва да преминете всички тестове в нея.

Подзадача	Точки	N	Q	L, K
1	10	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	$1 \leq L, K \leq 100$
2	11	$1 \leq N \leq 1\ 000$	$1 \leq Q \leq 100$	$1 \leq L, K \leq 10^9$
3	16	$1 \leq N \leq 2\ 000$	$1 \leq Q \leq 2\ 000$	$1 \leq L, K \leq 10^9$
4	12	$1 \leq N \leq 5\ 000$	$1 \leq Q \leq 5\ 000$	$1 \leq L, K \leq 10^9$
5	23	$1 \leq N \leq 80\ 000$	$1 \leq Q \leq 80\ 000$	$1 \leq L, K \leq 10^9$
6	28	$1 \leq N \leq 300\ 000$	$1 \leq Q \leq 300\ 000$	$1 \leq L, K \leq 10^9$

Пример

Функция, която се вика	L	K	N	stationPositions или newStationPosition	stationOilPrices или newStationOilPrfice	Функцията връща
init	100	42	5	82 0 35 68 40	216 210 215 220 212	21188
addStation				90	209	21118

Обяснение на примера: При първоначалното разположение на бензиностанциите, задавано от функция *init*, за Лора е най-изгодно да направи следните зареждания: на бензиностанция в точка 0 – 42 л, на бензиностанция в точка 40 – 40 л, на бензиностанция в точка 82 – 18 л.

След добавянето на бензиностанция в точка 90, в която бензинът е по-евтин, за Лора става по-изгодно да направи следните зареждания: на бензиностанция в точка 0 – 42 л, на бензиностанция в точка 40 – 40 л, на бензиностанция в точка 82 – 8 л, на бензиностанция в точка 90 – 10 л.

Локално тестване

За да можете да тествате Вашите функции *init()* и *addStation()* на локалния си компютър, Ви се предоставят файлове *Lgrader.cpp* и *stations.h*. Компилирайте ги заедно с вашия файл

ЧЕТВЪРТО КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

ПЛОВДИВ, 10 юни, 2018 г.

Група А, 11-12 клас

stations.cpp и ще получите програма, с която да тествате функциите си. Програмата изисква от стандартния вход да се въведе следната последователност от данни:

От първия ред се въвеждат три цели, положителни числа L , K и N – дължина на пътя до морето, капацитет на резервоара и брой на бензиностанциите в първоначалния набор.

От втория ред се въвеждат N цели, неотрицателни числа, разделени с интервали – координатите на бензиностанциите в първоначалния набор.

От третия ред се въвеждат N цели, неотрицателни числа, разделени с интервали – цените на литър гориво в бензиностанциите в първоначалния набор.

От следващия ред се въвежда цяло, неотрицателно число Q – брой на заявките за добавяне на нови бензиностанции.

Следват Q реда, всеки от които съдържа информация за една заявка за нова бензиностанция. Всеки ред съдържа по две цели неотрицателни числа, разделени с интервал – координатата на новата бензиностанция и цената на литър гориво в нея.

На изход програмата ще изведе $Q+1$ реда, всеки от които съдържа по едно цяло число – минималната цена, за която Лора може да стигне до морето при първоначалния набор от бензиностанции и след добавянето на всяка нова бензиностанция.

ЧЕТВЪРТО КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

ПЛОВДИВ, 10 юни, 2018 г.

Група А, 11-12 клас

Задача АК3. Road Signs

Ели живее на безкрайно дълга права улица, движеща се от запад на изток. По нея има разположени N пътни знака с числа от двете си страни. i -тият знак се намира на P_i метра на изток от къщата на Ели (като P_i може да бъде и отрицателно, в случай, че знакът се намира на запад от къщата на момичето). Всеки знак има изписани числата W_i на страната, виждана от пътуващите от изток на запад, и E_i на страната, виждана от пътуващите от запад на изток.

Ели не знае за какво са тези знаци. Момичето подозира, че числата обозначават разстоянието (в метри) от всеки от знаците до някакви две много важни точки. Разбира се, различните знаци могат да сочат към различни точки.

Сега Ели се чуди кои са най-важните двойки точки на пътя. Момичето определя "важността" на двойка точки X и Y като броя знаци, които сочат поне една от двете. По-формално, важността за двойка точки с координати X и Y се определя като броя на всички знаци, за които поне едно от следните условия е изпълнено:

$P_i - W_i = X$	$P_i - W_i = Y$	$P_i + E_i = X$	$P_i + E_i = Y$
-----------------	-----------------	-----------------	-----------------

Точките X и Y могат да са с произволни координати, включително отрицателни такива - както казахме, улицата е безкрайно дълга и в двете посоки.

Възможно е да има повече от една двойка "най-важни" точки: например, ако K знака сочат точките X_1 и Y_1 , но също така K (потенциално различни) знака сочат X_2 и Y_2 . Сега Ели иска да намери колко важни са най-важните точки, и колко двойки такива точки има.

Вход

На първия ред на стандартния вход ще бъде зададено едно цяло число N – броят знаци. На всеки от следващите N реда ще бъдат зададени по три цели числа P_i , W_i , и E_i – съответно позицията на знака; числото, виждано от пътуващите от изток на запад; и числото, виждано от пътуващите от запад на изток. Знаците ще са зададени в нарастващ ред на позицията си P_i , като няма да има два знака с една и съща позиция. Също така няма да има точка, която е сочена от всички знаци.

Изход

На единствен ред на стандартния изход изведете две цели числа – колко е важността на най-важните двойки точки, и броя различни двойки такива точки.

Ограничения

- ❖ $2 \leq N \leq 100,000$
- ❖ $1 \leq E_i, W_i \leq 1,000,000$
- ❖ $-1,000,000 \leq P_i \neq P_j \leq 1,000,000$
- ❖ В тестове, даващи 20% от точките за задачата, $2 \leq N \leq 20$
- ❖ В тестове, даващи 40% от точките за задачата, $2 \leq N \leq 500$
- ❖ В тестове, даващи 60% от точките за задачата, $2 \leq N \leq 10,000$

Примери

Примерен Вход	Примерен Изход
11 -17 10 19 -8 2 50 -2 11 3 0 13 42 3 2 6 7 33 15 11 10 31 12 25 4 13 10 7 14 10 10 17 16 25	6 3

Важността на най-важните двойки точки е 6. Съществуват три двойки такива точки:
 $X_1 = -13, Y_1 = 1$ (със сочещи ги знаци $\{3, 4, 5, 7, 8, 11\}$),

**ЧЕТВЪРТО КОНТРОЛНО СЪСТЕЗАНИЕ НА
РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР**

ПЛОВДИВ, 10 юни, 2018 г.

Група А, 11-12 клас

$X_2 = -13, Y_2 = 42$ (със сочещи ги знаци {2, 3, 4, 7, 8, 11}),

$X_3 = 1, Y_3 = 42$ (със сочещи ги знаци {2, 3, 4, 5, 7, 11}).

<i>Примерен Вход</i>	<i>Примерен Изход</i>
24	4 42
-10 9 1	
-9 20 10	
-8 1 11	
-7 12 17	
-6 20 16	
-5 22 13	
-4 30 6	
-3 20 7	
-2 25 16	
-1 3 21	
0 30 7	
1 17 22	
2 9 5	
3 25 18	
4 21 13	
5 16 17	
6 4 5	
7 15 8	
8 11 24	
9 26 3	
10 25 14	
11 2 21	
12 22 5	
13 24 17	

Всички най-важни двойки точки са:

*(-27, -19), (-27, -17), (-27, -11), (-27, -9), (-27, 2), (-27, 7), (-27, 10), (-27, 17), (-27, 32), (-19, -17),
(-19, -11), (-19, 2), (-19, 7), (-19, 17), (-19, 32), (-17, -11), (-17, -9), (-17, 2), (-17, 7), (-17, 10),
(-17, 32), (-11, -9), (-11, 2), (-11, 7), (-11, 10), (-11, 17), (-11, 32), (-9, 2), (-9, 7), (-9, 10), (-9, 17),
(-9, 32), (2, 7), (2, 10), (2, 17), (2, 32), (7, 10), (7, 17), (7, 32), (10, 17), (10, 32), (17, 32)}*