Кръг I. Парашутизъм

Мечо Пух и неговите приятели следяха с интерес първото издание на конкурса *CodeIT*. Те също имаха желание да участват в него, ала бяха възпрепятствани от бавната безжична мрежа в гората. Всички те много съжаляваха за пропуснатата възможност да спечелят голямата награда и затова решиха сами да си организират екстремно изживяване. И тъй като горското *„Мазерати“* вече беше нает, те преминаха към план Б – скачане с парашут.

Обаче преди всяко приключение трябва първо да се помисли за безопасността! Мечо Пух вече определи мястото, където ще се приземяват животните. Това е поле, което може да се представи като матрица с размери N на M. Всяка клетка от тази матрица задава безопасността на съответстващия ѝ участък от полето. Мечо разполага и със списък на животните, записали се за скачане, общо P на брой. За всяко едно от животните са известни:

1) Формата му при приземяване. Това е таблица, която показва кои части от животното имат пряк допир с полето и съответно са изложени на риск. Всяка форма ще се състои от една компонента и няма да съдържа отцепени, несвързани части в себе си. Две клетки от формата са свързани, ако имат обща страна. Животните не могат да се завъртат при летенето. Тяхната форма на приземяване е фиксирана и не подлежи на ротации;
2) Коефициент на въздействие. След всяко приземяване настъпва поредица от сложни физични закони върху мястото на приземяването, под чието въздействие то става по-опасно за по-нататъшни приземявания. Тези сложни закони се свеждат до елементарно целочислено деление на частите от полето, засегнати от животното, на неговия коефициент на въздействие;
3) Долна граница за безопасност. При всяко приземяване е изключително важно никоя част от тялото на съответното животно да не попада на поле, което е под неговата долна граница за безопасност.

Безопасността за един скок се дефинира като сумата от безопасностите на всички клетки, засегнати от животното, което извършва скока. Не забравяйте, че след скока всички засегнати клетки ще намалеят толкова пъти, колкото е коефициентът на въздействие на животното. За повече информация, вижте примера.

Помогнете на Мечо Пух като напишете програма, която разпределя животните, така че сумарна безопасност от всички извършени скокове да е максимална. Не е задължително всяко записало се за скачане животно да извърши скок.

Вход:

От първия ред на входния файл **parash.in** се въвеждат три цели числа – N, M и P. На следващите N реда е зададено полето. На всеки от тях са дадени по M цели числа, съответно безопасността на всяка клетка. На следващите редове е дадена информация за всяко животно. За $i^{\text{-тото}}$ животно се въвеждат 4 цели числа – r_i, c_i, k_i и t_i, като r_i и c_i са съответно броят редове и броят колони на таблицата, която определя формата му при приземяване, k_i е коефициентът на въздействие и t_i е неговата долна граница за безопасност. На следващите r_i реда има по c_i символа. Всеки символ е 1, ако съответната клетка би имала допир с полето при приземяване, или 0 в противен случай.

Изход:

На първия ред от изходния файл **parash.out** програмата трябва да изведе едно цяло число B – броят на животните, които ще скачат според разписанието. На следващите B реда програмата трябва да изведе по 3 цели числа, зададени в този ред – номерът на поредното животно, което ще скача, както и реда и колоната на клетката, която съвпада с горния ляв край на таблицата, описваща животното. Животните са номерирани от 1 до P спрямо реда, зададен във входния файл. Редовете и колоните са индексирани от 1.

Ограничения:

$2 \le \mathbf{N}, \mathbf{M} \le 50$
$1 \le \mathbf{P} \le 100$
$1 \le \mathbf{r_i} \le \min(N, 10)$
$1 \le \mathbf{c_i} \le \min(M, 10)$
$2 \le \mathbf{k_i} \le 1000$
$1 \le \mathbf{t_i} \le 1000$
1 ≤ Безопасността на всяка клетка от полето ≤ 100 000

Таблицата, дефинираща формата на приземяване на дадено животно, няма да съдържа редове или колони състоящи се единствено от нули.

Оценяване:

Оценяването ще стане по формулата $\left(\frac{\mathbf{yours}+1}{\mathbf{max}+1}\right)^2 \times \frac{\mathbf{100}}{\mathbf{T}}$, където **max** е максималната сумарна безопасност на скоковете, получена от някого от участниците, **yours** е сумарната безопасност, получена от вашата програмата и **T** е броят тестове. Даден участник ще получи 0 точки за съответния тест независимо от сумарната безопасност, ако програмата му не се съобразява с долната граница на безопасност на някое животно или го приземява извън полето.

Времето за изпълнение на програмата за един тест е 2 секунди.

Пример:

parash.in:	parash.out:	Обяснение:
5 5 2 1 9 4 4 7 1 6 7 7 5 6 2 1 6 6 6 7 8 2 5 1 4 2 2 3 3 3 3 2 010 110 111 3 2 2 4 01 11 01	2 1 2 1 2 1 4	Дадено е едно възможно решение. Първото животно ще скача клетките с горен ляв край (2, 1). Безопасността на скока е 6+2+6+6+7+8 = 35. След скока всички тези клетки намалят 3 пъти. Второто животно ще скача върху клетките с горен ляв край (1, 4), чиято сумарна безопасност е 7+7+5+6 = 25. След скока използваните клетки намалят 2 пъти. Сумарната безопасност за двата скока е 60.

Тестване:

Следната таблица дава допълнителна информация за тестовете, които ще бъдат използвани при определяне на финалното класиране на кръга:

% от официалните тестове	Вид 1 (20 %)	Вид 2 (30 %)	Вид 3 (50 %)
$N \leq$	10	25	50
$M \leq$	10	25	50
$P \leq$	20	50	100
$k_i \leq$	10	100	1000
$t_i \leq$	10	100	1000
Максимална безопасност на клетка от полето	1000	10 000	100 000

Генератор:

За допълнително удобство, състезателите ще имат достъп до генератор на тестове за задачата. **С този генератор ще бъдат изготвени и финалните тестове.** Генераторът ще бъде качен на сайта на конкурса.

Генераторът има няколко режима на изпълнение. Всеки от тях е лесно достъпен от командния ред (*command prompt* под *Windows*; *terminal* или *konsole* под *Linux*):

1) Първи режим (без допълнителни параметри):

Операционна система	Linux	Windows
Начин на извикване (в команден ред)	./parash.gen	parash.gen

По подразбиране, SEED = 1 (вижте втори режим за повече подробности) и се използват ограниченията, зададени в секция „Ограничения“ на задачата.

2) Втори режим (със зададен SEED):

Операционна система	Linux	Windows
Начин на извикване (в команден ред)	./parash.gen 257	parash.gen 257

По този начин, Вие ще можете да генерирате тест със зададен от Вас SEED. На базата на този SEED ще бъдат случайно генерирани всички стойности в теста. В примера, SEED = 257. Изходът на генератора при SEED = 257 винаги ще бъде един и същ.

По подразбиране се използват ограниченията, зададени в секция „Ограничения“ на задачата.

3) Трети режим (със зададен SEED и файл, задаващ ограниченията в задачата):

Операционна система	Linux	Windows
Начин на извикване (в команден ред)	./parash.gen 257 constr.txt	parash.gen 257 constr.txt

Отново, както при втория режим, задаваме SEED, с който да бъде генериран теста. Разликата е, че в този режим чрез отделен файл задаваме и необходимите ни ограничения в задачата. В примера, файлът се намира в текущата ни работна директория. Ако това не е така, то трябва да се зададе местонахождението на файла.

Въпросният файл трябва да съдържа **точно** 6 цели числа – максимално допустимите стойности съответно за N, M, P, k_i, t_i и за безопасност на клетка от полето. Минималните стойности на тези числа се подразбира да са същите, като зададените им в секция „Ограничения“ на задачата.