Родословно дърво

Родствени връзки на планетата ФМИ са доста объркани. Те се събират в групи, така че фмитяните могат да имат както един, така и десет родителя и никои не е учуден от стотина деца. Фмитяните са свикнали с техния начин на живот, и го намират за много естествен, но в планетарния парламент това объркано родословно дърво води до объркване. Там се събират най-достойните фмитяни и затова, за да не се засегне някои по време на дискусии е решено първо да се дава думата на най-възрастните фмитяни, след това на по-младите и най-накрая на най-младите и бездетни фмитяни. Разбира се спазването на това решение не е тривиална задача. Не винаги фмитяните познават всичките си родители. Ако по грешка, някой говори преди родителите си или прародителите си, тогава става голям скандал. Задачата е да се напише програма която, да определи реда на изказване във фмитянския парламент, и да прекратят скандалите веднъж за винаги.

Входа съдържа множество тестови примери. Първия ред на всеки от тях съдържа числото N (1 <= N <= 10 000) – броя на членовете на фмитянския парламент. Според вековните традиции членовете на парламента са номерирани с естествените числя от 1 до N. Следват точно N реда, като i-тия ред съдържа списък с децата на i-тия член на парламента. Списъка с децата с поредица от номерата на децата в произволен ред разделени с интервали. Списъкът може да е празен. Списъкът, дори и празен, завършва с 0. Ако N е извън обявените граници това означава край на входа.

За всики от тестовите примери изведете по една единствена линия, списък с членовете на фмитянския парламент по реда на изказване, разделени с интервали. Ако задачата има няколко решения, отпечатайте кое да е от тях. Поне едно решение винаги съществува.

Пример

Вход	**Изход**
4 0 4 1 0 1 0 3 0 -45	2 4 3 1

ПРИЯТЕЛИ

В един град живеят N жители, за някои двойки от които се знае, че са приятели. От известната максима “Приятелите на моите приятели са и мои приятели” следва, че ако A и B са приятели, и B и C са приятели, то A и C също са приятели. Напишете програма, която намира броя на гражданите в най-голямата група от приятели.

На първия ред на стандартния вход е зададен броят на тестовете. Всеки тест започва с ред, на който са зададени числата N и M, където N е броят на жителите ($10 \leq N \leq 10000$), а M е броят на двойките, за които се знае, че са приятели ($0 < M \leq 50000$). На всеки от следващите M реда има по две числа – номерата на двойка приятели A, B ($1 \leq A \leq N$, $1 \leq B \leq N$, $A \neq B$), като между дадените двойки може да има и повтарящи се.

За всеки тестов пример програмата трябва да изведе на стандартния изход едно число – броя на гражданите в най-многобройната група от приятели.

Пример

Вход	**Изход**
1 10 12 1 2 3 1 3 4 5 4 3 5 4 6 5 2 2 1 7 10 1 2 9 10 8 9	6

ИГРА С ПЛОЧКИ

Върху част от клетките на правоъгълна мрежа от квадрати с N реда и M стълба са поставени плочки. Напишете програма, която да провери дали е възможно плочка да бъде преместена от зададено място на дъската до друго зададено място на дъската. Плочката може да бъде местена от квадрата, в който се намира, в някой от съседните **празни** квадрати отляво, отдясно, отгоре или отдолу и не може да напуска мрежата.

На първия ред на стандартния вход е зададен броят на тестовете. Всеки тест започва с ред, на който са зададени целите числа N и M (3 < N, M ≤ 500). Всеки от следващите N реда съдържа по един низ с дължина M, описващ поредния ред от дъската – знакът '*', означава, че в съответния квадрат има плочка, а знакът '.' (точка) – че квадратът е празен. Последният ред съдържа четири цели положителни числа – координатите на квадрата с плочката, която искаме да преместим, и координатите на празен квадрат, в който искаме да я преместим.

За всеки тест, на отделен ред на стандартния изход програмата трябва да изведе YES, ако преместването е възможно или NO, ако не е възможно.

Пример

Вход	**Изход**
2 4 5 ***** *...* ***.* ...*. 3 1 4 3 4 5 ***** *...* ***.* ...*. 1 2 4 5	YES NO

Забележка. Редовете са номерирани от 1 до N в реда по който са зададени на входа, а стълбовете от 1 до M – отляво надясно.

Дънна платка

Освен велик математик, Станчо е и млад техник. Едно от любимите му занимания за времето, в което се води, че трябва да прави нещо смислено е да модифицира домашния си компютър. При последното разглобяване Станчо реши да оптимизира тотално дънната си платка, тъй като не беше доволен от производителността ѝ, а не му се даваха пари за нова. За разликата от повечето компютри, при Станчовият дънната платка представлява матрица N на M от пинове, редовете и колоните на която са на разстояние 1. Всеки пин е или положителен или отрицателен. За да работи дънната платка, всеки положителен пин трябва да се свърже свързан към един отрицателен и всеки отрицателен към един положителен. Поради технически ограничения, връзките между пиновете са еднопосочни и трябва да са изградени от отсечки които са успоредни на редовете или колоните на матрицата. За сметка на това обаче, няма никакви проблеми тези връзки да се пресичат. Напишете програма, която по дадена дънна платка да определя най-малката възможна сумарна дължина на връзки които да се изградят, така че дънната платка да заработи.

На стандартния вход са дадени няколко описания на дънни платки. Всяко от тях започва с ред с двойка числа N и M (1 ≤ N, M ≤ 1000). Следват N реда с по M знака ‘+’ или ‘-’ описващи дали съответния пин е положителен или отрицателен.

За всяка дънна платка вашата програма трябва да изведе на стандартния изход ред с число — минималната обща дължина на връзки които да се изградят. В случай, че не е възможно платката да се свърже изведете 0.

Вход	Изход	Пояснение
2 2 +- -+ 2 3 +-- --- 1 3 +++ 3 6 +++-++ ++---+ +-----	4 10 0 28	При първия пример всеки пин може да се свърже към съседния. При втория трябва да се изградят връзки с дължина 1, 1, 2, 1, 2 и 3 Третата платка не може да се свърже, тъй като няма отрицателни пинове.