**1. Ламана**

**Максимальна оцінка: 100 балів**

**Обмеження на час: 1 сек.**

**Обмеження на пам’ять: 256 MБ**

**Вхідний файл: polyline.in**

**Вихідний файл: polyline.out**

**Програма: polyline.\***

**Завдання**

На площині позначено кілька точок. Якщо це можливо, побудуйте замкнену ламану без самоперетинів, що проходить через усі ці точки і не має відмінних від них вершин.

**Вхідні дані**

У першому рядку вхідного файлу задано кількість точок *n*, 3 ≤ *n* < 2017. У *n* наступних рядках вказано по дві координати кожної точки: абсцису та ординату відповідно. Точок, що повторюються, немає. Всі координати — цілі числа, що за модулем не перевищують 10 000.

**Вихідні дані**

Якщо побудувати потрібну замкнену ламану неможливо, слід вивести 0. В іншому разі треба вивести порядок, у якому задані точки розташовані на ламаній, тобто порядок обходу всіх цих точок, починаючи з довільного місця і в довільному напрямку (або за, або проти годинникової стрілки). Нумерація точок починається з одиниці. Ламаних, що задовольняють умову, може бути кілька. Достатньо знайти хоча б одну.

**Приклад**

| **polyline.in** | **polyline.out** |
| --- | --- |
| 71 0-1 -11 1-1 10 01 -10 -1 | 2 4 3 1 6 5 7 |

**2. Печери**

### Максимальна оцінка: 100 балів

### Обмеження на час: 7 сек.

### Обмеження на пам’ять: 128 MБ

### Вхідний файл: caves.in

### Вихідний файл: caves.out

### Програма: caves.\*

Пер Ґюнт, герой п’єси норвезького письменника Генріка Ібсена, потрапив у полон короля тролів. Щоб визволитися, Пер Ґюнт повинен перемогти у грі, яку король пропонує своїм бранцям. Правила гри такі:

* є10 печер , деякі з яких сполучені між собою;
* бранець грає протягом *m* днів:
	+ у перший день він може вибрати собі довільну печеру;
	+ кожного наступного дня він може або залишитися у печері, у якій перебуває, або перейти у будь-яку іншу печеру, до якої є хід з поточної печери;
* кожного дня король винагороджує бранця певною наперед обумовленою кількістю золотих монет, яка залежить і від печери, і від номера дня;
* якщо за всю гру бранець накопичить кількість монет, виражену одним з *l* «королівських чисел», то бранця навіки залишають у полоні — він програв. Інакше його відпускають на волю з виграшем — усією накопиченою за час гри кількістю монет.

**Завдання**

Визначити, *N*w, *p*max, *p*min — відповідно кількість різних виграшів, найбільший і найменший виграші.

|  |
| --- |
| **Приклад** |
| **caves.in** | **caves.out** |
| 2 21 0 1 0 0 0 0 0 0 00 1 0 0 0 0 0 0 0 01 0 1 0 0 0 0 0 0 00 0 0 1 0 0 0 0 0 00 0 0 0 1 0 0 0 0 00 0 0 0 0 1 0 0 0 00 0 0 0 0 0 1 0 0 00 0 0 0 0 0 0 1 0 00 0 0 0 0 0 0 0 1 00 0 0 0 0 0 0 0 0 11 2 3 0 0 0 0 0 0 04 5 6 0 0 0 0 0 0 07 0 | 2 9 5 |

**Вхідні дані**

Перший рядок файлу містить десяткові записи натуральних чисел *m, l,* де *m ≤*60.

Кожний з наступних 10 рядків містить по 10 цілих чисел — нулів або одиниць. При 1 ≤ *j, k* ≤ 10: якщо *k* число(*j*+ 1) рядка файлу дорівнює 1, то з *j* печери можна потрапити безпосередньо до *k* печери і навпаки. Інакше це зробити неможливо.

Кожний з наступних *m* рядків містить по 10 невід’ємних цілих чисел. При 1 ≤ *j* ≤ *m* і1 ≤ *k* ≤ 10*: k* число(*j* + 11) рядка — кількість монет, які отримує гравець як винагороду за перебування у *j* день у *k* печері.

Останній (*m* + 12) рядок містить у порядку спадання *l* різних цілих невід’ємних «королівських чисел», кожне з яких менше від 264. Останнє число у цьому рядку — нуль.

**Вихідні дані**

Єдиний рядок файлу містить десяткові записи трьох чисел: *N*w, *p*max, *p*min. Якщо *N*w = 0, то *p*max = *p*min = 0. Відомо, що *p*max  < 264, *l* + *N*w *≤* 106.

##