

## Завдання 1. Монополія

Дені дуже подобається грати в монополію. Але одна гра триває дуже довго – від 5 до 7 годин. Тож, Дені задумалася про зміну класичних правил. Гравці в монополію зазвичай рухаються в одному напрямку з позиції на позицію і через деякий час повертаються на початок, починають знову і так далі. У новій версії рух знову буде відбуватися з позиції на позицію, але з поточної позиції може бути кілька можливостей для наступного руху. Дені хоче знайти такі спрямовані з'єднання між позиціями в грі, щоб гравець ніколи не міг повернутися в позицію, де він був, як би він не рухався (звичайно, дотримуючись правил). Таким чином гра буде меншою за тривалістю.

Вона вже почала виготовляти нову дошку – вона вибрала кількість позицій  $N$  (позиції пронумеровані від 1 до  $N$ ) і склала список з  $M$  з'єднань (кожне з'єднання має напрямок, і немає з'єднання, яке з'єднує позицію із самою собою). Якщо позиція  $i$  пов'язана з позицією  $j$ , тоді немає прямого з'єднання в протилежному напрямку, тобто від позиції  $j$  до позиції  $i$ , а також немає інших прямих з'єднань із позиції  $i$  в позицію  $j$ . Дені подумала, що її гра з новою дошкою готова, але раптом помітила, що умова, яку вона хоче додати (коли ви переміщуєтесь з позиції в позицію, використовуючи з'єднання, ви не можете повернутися до раніше відвіданої позиції), не відповідає її списку з'єднань. Спочатку вона подумала видалити деякі з прямих з'єднань, але це призведе до переписування списку, який може бути дуже довгим. Ось чому Дені вирішила змінити напрямок деяких спрямованих з'єднань.

### Завдання

Ви регулярно граєте в монополію з Дені. Ось чому ви хочете допомогти їй, написавши програму **monopoly**, яка має вказати їй, які з'єднання слід змінити, щоб зазначена умова виконувалася. Програма повинна містити функцію *find\_reverse*, яка буде скопійована разом з програмою журі.

### Деталі реалізації

Функція *find\_reverse* повинна мати наступний прототип:

```
std::string find_reverse (int N, int M, int connections[][2]);
```

Вона викликається лише один раз програмою журі з трьома параметрами:  $N$  – кількість позицій на новій дошці,  $M$  – кількість спрямованих з'єднань у списку Дені та масив *connections* який має  $M$  рядків, кожен складається з двох чисел  $x$  та  $y$  – початкова та кінцева позиція для спрямованого з'єднання. Ця функція повинна повертати двійковий рядок довжини  $M$  – в порядку масиву *connections*, для кожного з'єднання вам потрібно встановити '1' в рядку, якщо з'єднання потрібно перевернути і '0', інакше. Якщо існує декілька рішень, ви можете повернути будь-яке з них.

Ви повинні надіслати в систему файл **monopoly.cpp**, який містить реалізацію функції *find\_reverse*. Файл може містити інші функції та код, необхідні для вашої програми, але він **не повинен містити** головну функцію – *main*. Крім того, ви не повинні намагатися читати зі стандартного введення чи записувати на стандартний вивід!

### Обмеження

- ♣  $3 \leq N \leq 5 \times 10^5$
- ♣  $3 \leq M \leq 1.5 \times 10^6$

### Підзавдання

Підзавдання	Бали	$N$	$M$	Додаткові обмеження
1	0	–	–	Приклад
2	15	$\leq 7$	$\leq 21$	–
3	40	$\leq 10^3$	$\leq 5 \times 10^3$	–
4	25	$\leq 10^5$	$\leq 5 \times 10^5$	–
5	20	$\leq 5 \times 10^5$	$\leq 1.5 \times 10^6$	–

Щоб отримати бали за дану підзадачу, ваше рішення має пройти всі тести для підзадачі.

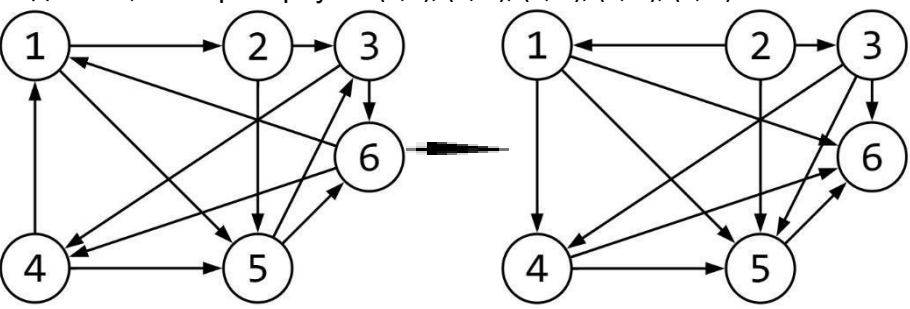
### Локальне тестування

Для локального тестування вам надається файл **Lgrader.cpp**. Ви повинні помістити його в ту саму папку, де і ваша програма **monopoly.cpp** і ви повинні зібрати разом файли **Lgrader.cpp** та **monopoly.cpp**. Таким чином ви створите програму для перевірки правильності вашої функції. Програмі знадобиться наступна послідовність даних зі стандартного введення:

- на першому рядку: два цілих числа – кількість позицій  $N$  і кількість з'єднань  $M$  нової дошки.
- на останніх  $M$  рядках: у кожному рядку має бути два цілих числа  $x$  та  $y$  – початкова і кінцева позиція для кожного спрямованого з'єднання.

Результатом програми буде знайдений вами двійковий рядок.

### Приклад локального тестування

Вхід	Вивід	Пояснення
6 12 1 2 1 5 2 3 2 5 3 4 3 6 4 1 4 5 5 3 5 6 6 1 6 4	100000101011	<p>З'єднання, які перевернули: (1,2), (4, 1), (5, 3), (6, 1), (6, 4).</p>  <p>На малюнку вище показано позиції, зліва, з спрямованими зв'язками зі списку Дені, праворуч, після зміни напрямку з'єднань з відповідною '1' у виводі. Зрозуміло, що на початковій дошці використовуючи спрямовані з'єднання (1, 2), (2, 3), (3, 4) та (4, 1) ми можемо почати з позиції 1 і рухаючись вздовж цих спрямованих з'єднань, ми повернемося в цю ж позицію. На отриманій дошці видно, що умова Дені зберігається. Зверніть увагу, що є й інші дійсні рішення: (4, 1), (5, 3), (6, 1), (6, 4) (1, 5), (2, 3), (2, 5), (3, 4), (3, 6), (4, 5), (5, 6)</p>

