

# XV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Общински кръг

25.01.2009 г.

Група А (11.-12. клас)

## Задача А1. ОЦВЕТЯВАНЕ

Разглеждаме точките с целочислени координати  $x$  върху правата ( $0 < x < 500001$ ). Първоначално тези точки са оцветени с бял цвят. Прилагаме  $N$  операции от следния вид: при  $i$ -тата операция променяме цвета на тези от разглежданите точки, които са с координати в интервала от  $a_i$  до  $b_i$  (включително и на точките в двата края на интервала) – ако точката е била оцветена в бяло, оцветяваме я в черно, а ако е била оцветена в черно, оцветяваме я в бяло.

Напишете програма **color**, която извежда броя на точките от най-дългата последователност от съседни точки, оцветени в черно, която се е получила след завършване на последната от  $N$ -те операции.

### Вход

От първия ред на стандартния вход се въвежда едно цяло число  $N$ . От следващите  $N$  реда се въвеждат по две цели числа  $a_i$  и  $b_i$ , разделени с интервал.

### Изход

На един ред на стандартния изход програмата трябва да изведе едно цяло число, равно на търсения брой.

### Ограничения

$0 < N < 5001$   
 $0 < a_i < 500001,$   
 $0 < b_i < 500001,$   
 $a_i \leq b_i, i = 1, \dots, N$

## ПРИМЕР

### Вход

2  
3 7  
5 10

### Изход

3

## Задача А2. ШОСЕТА

В държавата  $X$  има  $N$  на брой населени места, като някои двойки от тях са свързани с преки шосета, а цялата мрежа от шосета е така изградена, че между всеки две населени места съществува път (макар и не пряк). В една прекрасна

година държавата  $X$  била приета за член на Европейския Съюз и тъй като нейните шосета не отговаряли на стандартите на съюза, те трябвало да бъдат разрушени и построени отново. Правителството се събрало и по принципа „като ще е гарга, рошава да е“ решило изцяло да промени системата от пътища, а именно да се ликвидират старите шосета и да се прекарат преки шосета между всички двойки населени места, между които не е имало преки шосета при старата система. Някой от съветниците на премиера плахо се обадил, че при такава промяна може да се случи така, че да има двойки населени места, между които да не съществува път (макар и не пряк). Това накарало господа министрите сериозно да се замислят и те възложили на група от експерти да пресметнат при изграждане на новата шосейна мрежа какъв минимален брой от старите преки шосета да се запазят и ремонтират, така че да няма двойка населени места, между които да липсва път.

Напишете програма **roads**, която да помогне на експертите да се справят с поставената им задача.

### Вход

От първия ред на стандартния вход се въвеждат две цели числа  $N$  и  $M$ , разделени с интервал:

$N$  – брой на населените места

$M$  – брой на преките шосета

От следващите  $M$  реда се въвеждат по две цели числа  $a$  и  $b$ , разделени с интервал, задаващи номерата на двете населени места, между които съществува пряко шосе (всички населени места са номерирани от 1 до  $N$ ).

### Изход

На един ред на стандартния изход програмата трябва да изведе едно число, равно на минималния брой стари преки шосета, които трябва да се запазят и ремонтират.

### Ограничения

$N \leq 1000$

$M \leq 500000$

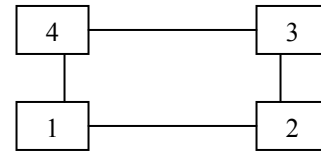
$1 \leq a, b \leq N$

### ПРИМЕР

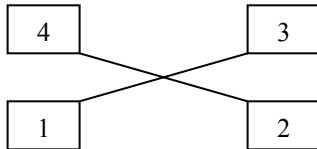
Вход	Изход
4 4	1
1 2	
2 3	
3 4	
4 1	

### Обяснение на примера:

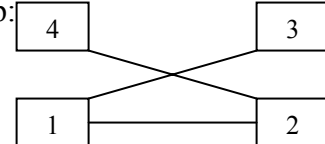
Конфигурацията, съответстваща на примерния вход е:



Като се прекарат нови пътища и се унищожат старите се получава



За да има път между всеки две населени места и при новата пътна мрежа, трябва да се запази минимум едно от старите преки шосета, например:



### Задача А3. САНСУ

Върху квадратна плочка 3x3 са записани цифрите от 1 до 9 в някакъв ред

. Ще казваме, че това е „сансу”, ако всички трицифрени числа, образувани от цифрите на всеки ред, на всяка колонка и по двата диагонала, са кратни на 3. Показаната плочка не е сансу: числата 468 и 492 се делят на 3, но нито едно от останалите (197, 352, 413, 695, 872 и 893) не се дели на 3.

4	6	8
1	9	7
3	5	2

За две цифри на плочката ще казваме, че са съседни, ако са написани в квадратчета, които имат обща страна (в примера 4 и 6, 9 и 5, и т. н., но не 4 и 9, още по-малко 1 и 2!).

Единственият позволен ход е да сменим местата на две съседни цифри. Определете колкото е възможно по-малък брой ходове, чрез които зададената плочка се превръща в сансу. Напишете програма **sansu**, която определя този брой и показва получения резултат.

#### Вход:

На три реда от стандартния вход са зададени цифрите от 1 до 9 (евентуално разбъркани, но всяка по веднъж) по три на ред, разделени с интервал: това са редовете на зададената плочка.

#### Изход:

Изведете на първия ред на стандартния изход едно цяло число – най-малкия намерен брой ходове, необходими за превръщането на зададената плочка в сансу. На трите следващи реда изведете полученото сансу: цифрите по редове от горе надолу, разделени с интервал.

## ПРИМЕР

### Вход

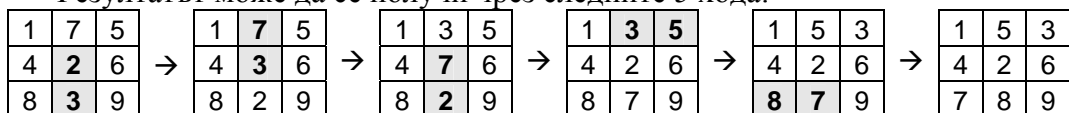
1 7 5  
4 2 6  
8 3 9

### Изход

5  
1 5 3  
4 2 6  
7 8 9

## Обяснение на примера:

Резултатът може да се получи чрез следните 5 хода:



С други пет хода може да се получи и друго сансу, например:

6 5 7  
1 3 2  
8 4 9

Но не може да се получи сансу с по-малко от 5 хода.

## Оценяване:

Вярно решение за тест получава от 0 до 10 точки, в зависимост от близостта си до оптималното решение.