

카드 역배치

1부터 20까지 숫자가 하나씩 쓰인 20장의 카드가 아래 그림과 같이 오름차순으로 한 줄로 놓여있다. 각 카드의 위치는 카드 위에 적힌 숫자와 같이 1부터 20까지로 나타낸다.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
카드	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

이제 여러분은 다음과 같은 규칙으로 카드의 위치를 바꾼다: 구간 $[a, b]$ (단, $1 \leq a \leq b \leq 20$)가 주어지면 위치 a 부터 위치 b 까지의 카드를 현재의 역순으로 놓는다.

예를 들어, 현재 카드가 놓인 순서가 위의 그림과 같고 구간이 $[5, 10]$ 으로 주어진다면, 위치 5부터 위치 10까지의 카드 5, 6, 7, 8, 9, 10을 역순으로 하여 10, 9, 8, 7, 6, 5로 놓는다. 이제 전체 카드가 놓인 순서는 아래 그림과 같다.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
카드	1	2	3	4	10	9	8	7	6	5	11	12	13	14	15	16	17	18	19	20

이 상태에서 구간 $[9, 13]$ 이 다시 주어진다면, 위치 9부터 위치 13까지의 카드 6, 5, 11, 12, 13을 역순으로 하여 13, 12, 11, 5, 6으로 놓는다. 이제 전체 카드가 놓인 순서는 아래 그림과 같다.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
카드	1	2	3	4	10	9	8	7	13	12	11	5	6	14	15	16	17	18	19	20

오름차순으로 한 줄로 놓여있는 20장의 카드에 대해 10개의 구간이 주어지면, 주어진 구간의 순서대로 위의 규칙에 따라 순서를 뒤집는 작업을 연속해서 처리한 뒤 마지막 카드들의 배치를 구하는 프로그램을 작성하시오.

소스파일의 이름은 reversal.c 또는 reversal.cpp이며 수행시간은 1초를 넘을 수 없다. 사용하는 메모리는 128MB를 넘을 수 없다.

입력 형식

다음 정보가 표준 입력으로 주어진다. 총 10개의 줄에 걸쳐 한 줄에 하나씩 10개의 구간이 주어진다. i 번째 줄에는 i 번째 구간의 시작 위치 a_i 와 끝 위치 b_i 가 차례대로 주어진다. 이때 두 값의 범위는 $1 \leq a_i \leq b_i \leq 20$ 이다.

출력 형식

다음 정보를 표준 출력으로 출력한다. 1부터 20까지 오름차순으로 놓인 카드들에 대해, 입력으로 주어진 10개의 구간 순서대로 뒤집는 작업을 했을 때 마지막 카드들의 배치를 한 줄에 출력한다.

입력과 출력의 예

입력(1)

```
5 10
9 13
1 2
3 4
5 6
1 2
3 4
5 6
1 20
1 20
```

출력(1)

```
1 2 3 4 10 9 8 7 13 12 11 5
6 14 15 16 17 18 19 20
```

※ 위 출력은 한 줄에 출력된 것임

입력(2)

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

출력(2)

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20				

※ 위 출력은 한 줄에 출력된 것임

입력(3)

1	20
2	19
3	18
4	17
5	16
6	15
7	14
8	13
9	12
10	11

출력(3)

20	2	18	4	16	6	14	8	12	10	11
9	13	7	15	5	17	3	19	1		

※ 위 출력은 한 줄에 출력된 것임

컬러볼

지훈이가 최근에 즐기는 컴퓨터 게임이 있다. 이 게임은 여러 플레이어가 참여하며, 각 플레이어는 특정한 색과 크기를 가진 자기 공 하나를 조정하여 게임에 참여한다. 각 플레이어의 목표는 **자기 공보다 크기가 작고 색이 다른 공**을 사로잡아 그 공의 크기만큼의 점수를 얻는 것이다. 그리고 다른 공을 사로잡은 이후에도 본인의 공의 색과 크기는 **변하지 않는다**. 다음 예에는 네 개의 공이 있다. 편의상 색은 숫자로 표현한다.

공 번호	색	크기
1	1	10
2	3	15
3	1	3
4	4	8

이 경우, 2번 공은 다른 모든 공을 사로잡을 수 있다. 반면, 1번 공은 크기가 더 큰 2번 공과 색이 같은 3번 공은 잡을 수 없으며, 단지 4번 공만 잡을 수 있다.

공들의 색과 크기가 주어졌을 때, 각 플레이어가 사로잡을 수 있는 모든 공들의 **크기의 합**을 출력하는 프로그램을 작성하시오.

소스파일의 이름은 ball.c 또는 ball.cpp이며 수행 시간은 1초를 넘을 수 없다. 사용하는 메모리는 128MB를 넘을 수 없다.

입력 형식

표준 입력의 첫 줄에는 공의 개수를 나타내는 자연수 N 이 주어진다($1 \leq N \leq 200,000$). 다음 N 개의 줄 중 i 번째 줄에는 i 번째 공의 색을 나타내는 자연수 C_i 와 그 크기를 나타내는 자연수 S_i 가 주어진다($1 \leq C_i \leq N$, $1 \leq S_i \leq 2,000$).

서로 같은 크기 혹은 같은 색의 공들이 있을 수 있다.

출력 형식

표준 출력에 N 개의 줄을 출력한다. N 개의 줄 중 i 번째 줄에는 i 번째 공을 가진 플레이어가 잡을 수 있는 모든 공들의 크기 합을 출력한다.

입력과 출력의 예

입력(1)

```
4
1 10
3 15
1 3
4 8
```

출력(1)

```
8
21
0
3
```

입력(2)

```
3
2 3
2 5
2 4
```

출력(2)

```
0
0
0
```

L 모양의 종이 자르기

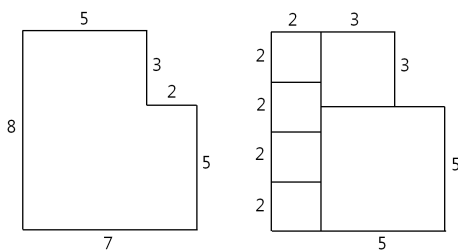
각 변의 길이가 양의 정수이고 영문자 L 모양인 종이가 주어져 있다. 이 종이를 칼로 여러 번 잘라서 모든 조각이 한 변의 길이가 양의 정수인 정사각형이 되도록 하고자 한다.

칼로 종이를 자르는 규칙은 다음과 같다.

- ① 자르는 방향은 수직 또는 수평만 허용된다. 즉, 사선으로는 자를 수 없다.
- ② 자르는 도중 칼의 방향을 바꿀 수 없다.
- ③ 자르는 도중에 칼을 멈출 수 없다. 즉, 일단 어떤 조각을 자르기 시작하면 그 조각이 반드시 둘로 분리될 때 까지 자른다.
- ④ 잘려진 조각의 각 변의 길이는 양의 정수이어야 한다.

위의 규칙에 따라 주어진 L 모양의 종이를 잘라 각 조각이 정사각형이 되도록 하되, 잘려진 조각 개수가 최소가 되도록 하고자 한다.

예를 들어, 각 변의 길이가 아래 왼쪽 그림에서 보인 것과 같은 종이가 주어질 때, 최소 개수의 정사각형 조각을 얻도록 자른 결과를 아래 오른쪽 그림에서 보였다.

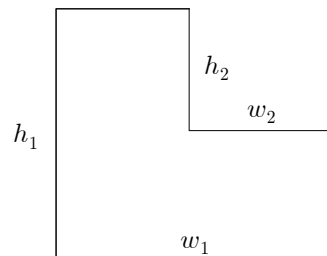


L 모양의 종이를 제시한 규칙에 따라 잘랐을 때, 잘려진 조각의 개수가 최소가 되도록 하는 프로그램을 작성하시오.

소스파일의 이름은 cut.c 또는 cut.cpp이며 수행 시간은 2초를 넘을 수 없다. 사용하는 메모리는 128MB를 넘을 수 없다.

입력 형식

표준입력에는 L 모양 종이의 각 변의 길이에 관한 정보를 나타내는 네 정수 h_1 , w_1 ($2 \leq h_1, w_1 \leq 50$), h_2 ($1 \leq h_2 < h_1$), 그리고 w_2 ($1 \leq w_2 < w_1$)가 차례대로 주어진다. 각 정수에 대응하는 변은 아래 그림에서 보인 것과 같다.



출력 형식

입력에서 주어진 변의 길이를 갖는 L 모양의 종이를 제시한 규칙에 따라 잘랐을 때 생긴 조각의 최소 개수를 표준출력 한 줄에 출력한다.

입력과 출력의 예

입력(1)

8 7 3 2

출력(1)

6

입력(2)

5 3 1 1

출력(2)

3

입력(3)

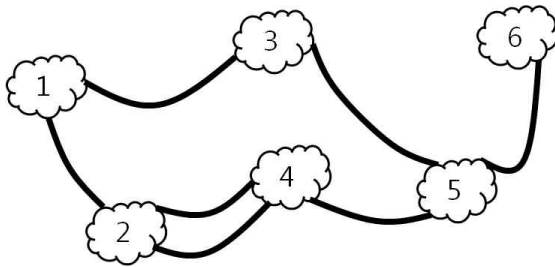
5 5 1 2

출력(3)

6

공중도시

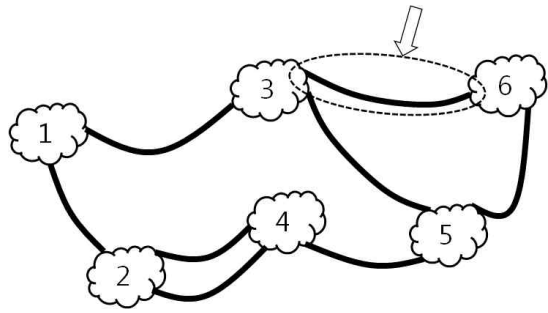
서기 4,000년 현재, 지구의 황폐화로 인하여 사람들은 공중에 섬을 띄워 공중도시를 만들어 살고 있다. 각 공중도시는 부양무게의 한계로 작은 크기의 섬으로 만들고 대신 다리로 연결하여 모든 도시 사이에 이동이 가능하다. 아래 그림은 도시 1부터 도시 6까지 모두 6개의 공중도시가 서로 다리로 연결된 모습을 보여준다.



두 도시는 서로 다른 두 개 이상의 다리로 직접 연결될 수도 있다. 위 그림에서 도시 2와 도시 4는 서로 다른 두 개의 다리로 연결되어 있다.

그런데, 간혹 발생하는 천재지변 때문에 다리가 끊어질 가능성이 있다. 위 그림에서 도시 5와 도시 6을 잇는 다리가 하나 끊어진다면 도시 6에서는 다른 도시로 이동할 수가 없지만, 도시 1과 도시 3을 잇는 다리가 하나 끊어지더라도 여전히 모든 도시 사이에 이동이 가능하다.

그래서 하나의 다리가 끊어지더라도 여전히 모든 두 도시 사이에 이동이 가능하도록 다리를 추가로 건설하려고 한다. 위 그림의 예에서는 다음 그림과 같이 도시 3과 도시 6을 잇는 다리를 하나 추가로 건설하면 임의의 다리가 하나 끊어지더라도 여전히 모든 도시 사이에 이동이 가능하다. 물론 도시 3 대신 다른 도시와 도시 6을 잇는 다리를 하나 추가해도 가능하다.



공중도시와 현재 상태의 다리가 주어져 있을 때, 임의의 다리가 하나 끊어지더라도 여전히 모든 도시 사이에 이동이 가능할 수 있도록 다리의 길이에 상관없이 추가로 건설해야 할 다리의 최소 개수와 그 위치를 찾는 프로그램을 작성하시오.

소스파일의 이름은 city.c 또는 city.cpp이며 수행 시간은 1초를 넘을 수 없다. 사용하는 메모리는 128MB를 넘을 수 없다.

입력 형식

다음 정보가 표준 입력으로 주어진다. 첫 줄에는 도시의 개수 N 과 다리의 개수 M 이 주어진다. 두 값의 범위는 $3 \leq N \leq 100,000$, $N-1 \leq M \leq 200,000$ 이다. 그 다음 M 개의 줄에 걸쳐 각 줄에는 다리로 직접 연결된 두 도시 C_1 과 C_2 가 차례대로 주어진다. 이때 $1 \leq C_1, C_2 \leq N$ 이다. 주어진 다리를 이용하여 모든 도시 사이에 이동이 가능하다. 다리의 개수가 도시의 개수보다 하나 작은 경우만 풀어도 30%의 점수를 받을 수 있다.

출력 형식

다음 정보를 표준 출력으로 출력한다. 첫 줄에는 추가로 건설할 다리의 최소 개수 R 을 출력한다. 그 다음 R 개의 줄에 걸쳐 각 줄에는 추가로 건설할 다리로 직접 연결될 두 도시 D_1 과 D_2 를 차례대로 출력한다. 이때 $1 \leq D_1, D_2 \leq N$ 이다. 이

때 다리의 순서는 상관이 없으며, 답이 여러 개인 경우에는 그 중 한 가지만 출력하면 된다.

입력과 출력의 예

입력(1)

```
6 7
1 2
1 3
2 4
2 4
4 5
3 5
5 6
```

출력(1)

```
1
3 6
```

입력(2)

```
3 3
1 2
1 3
2 3
```

출력(2)

```
0
```

입력(3)

```
9 10
1 2
2 3
2 3
3 4
4 5
4 6
3 6
6 7
6 8
8 9
```

출력(3)

```
2
1 5
7 9
```