

첨부 5

제 16 회  
한 국 정 보 올 림 피 아 드

경 시 문 제

## 검은점과 하얀점 연결

$2n$ 개의 점이 x축의 좌표  $1, 2, \dots, 2n$ 에 놓여 있다. 그 중  $n$ 개는 검은 점이고 나머지  $n$ 개는 하얀 점이다. 하나의 검은 점과 하나의 하얀 점을 연결하여 한 쌍을 만들면, 모두  $n$ 개의 쌍이 만들어진다. 한 쌍의 점을 연결할 때는, 왼쪽 점에서 출발하여 수직으로 올라가고, 거기서 수평으로 오른쪽으로 간 후, 다시 수직으로 내려가서 연결하면 하나의 길이 생긴다. 이렇게 생긴  $n$ 개의 길들은 서로 겹쳐서는 안되고, 서로 교차해서도 안된다. 모든 길의 거리의 합을 가장 작게 하도록

$n$ 개의 길은 만드는 프로그램을 작성하시오. 단, 거리의 단위는 수직, 수평 모두 1이다.

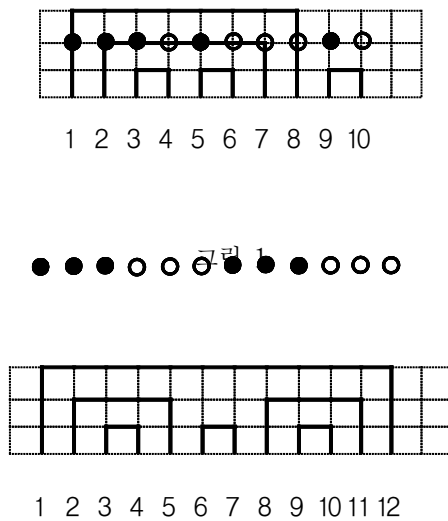


그림 2

그림 1의 경우 다른 방법으로 연결할 수도 있지만, 위 방법이 최소 연결 방법이고 거리의 합이 31이다. 그림 2의 경우도 다른 방법이 있지만, 위 방법이 최소 연결이며 거리의 합은 40이다.

실행 파일의 이름은 POINT.EXE로 한다. 실행시간은 10초를 넘으면 안되며, 정답이 아닌 경우라도 조건을 만족하면 부분 점수를 줄 수 있다.

## 입력 형식

입력 파일명은 INPUT.TXT이다. 첫째 줄에는 점의 개수를 나타내는 정수  $2n$ 이 주어진다.

$2n$ 은 100이하의 정수이다. 그 다음 줄에는  $n$ 개의 0과  $n$ 개의 1로 이루어진 문자열이 주어진다. 0은 하얀 점이고, 1은 검은 점이다. 왼쪽부터 차례로 좌표  $1, 2, \dots, 2n$ 에 해당한다.

## 출력 형식

출력 파일의 이름은 OUTPUT.TXT이다. 첫째 줄에는 길의 거리의 합을 출력한다. 다음  $n$ 개의 줄의 각 줄에는 연결되는 한 쌍의 점들의 좌표를 나타내는 두 정수를 출력한다. 두 정수 사이에는 빈칸이 하나 있다. 앞의 정수가 뒤 정수보다 작아야하고,  $n$ 개의 줄은 앞 정수가 커지는 순서로 출력한다.

## 입력과 출력의 예(그림1)

입력(INPUT.TXT)

```
10
1110100010
```

출력(OUTPUT.TXT)

```
31
1 8
2 7
3 4
5 6
9 10
```

## 입력과 출력의 예(그림2)

입력(INPUT.TXT)

12 111000111000
--------------------

출력(OUTPUT.TXT)

40 1 12 2 5 3 4 6 7 8 11 9 10
---

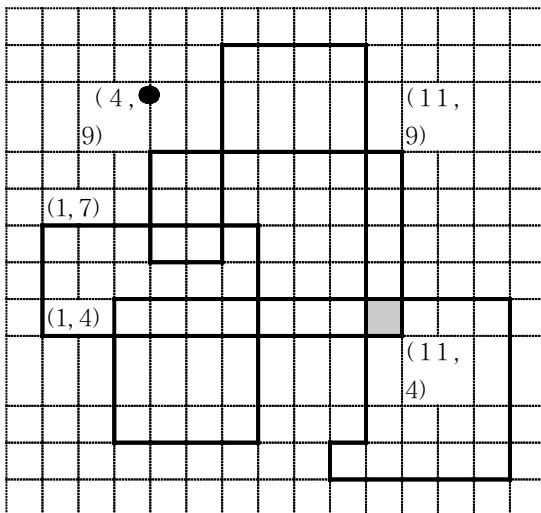


## 단순 사각형

평면상에서 어떤 로봇이 수평 또는 수직으로 선을 그으면서 움직인다. 이 로봇은 처음 정해진 위치에서 시작하여 움직인 뒤에 다시 처음 위치로 되돌아 온다.

로봇이 움직이는 방향은 위(Up), 아래(Down), 오른쪽(Right), 왼쪽(Left) 4가지뿐이며, 이는 각각 U, D, R, L로 표시된다. 또 움직인 거리는 양의 정수이다. 움직임은 방향과 거리로써 표현되는데 예를 들어 'R 3' 이라고 하면 오른쪽으로 3만큼 움직인 것이다.

최종적인 궤적을 살펴보면 다양한 다각형이 생겨난다. 이 다각형 중에서 그 내부에 다른 점이나 선분을 포함하지 않은 사각형을 단순 사각형이라고 한다. 문제는 주어진 궤적으로 만들어진 단순 사각형 중에서 가장 작은 면적의 단순 사각형을 구하는 것이다.



위의 그림에서 좌표 (4, 9)는 출발점이고 로봇의 움직임은 R 7, D 5, L 10, U 3, ...으로 진행된다. 위 그림에서 회색으로 표시한 부분이 가장 작은 면적의 단순 사각형이므로 이것이 답이 된다.

두 선분이 만나는 경우는 수직선분과 수평선분이 한 점에서 만나는 경우 뿐이며 수평선분과 수평선분, 수직선분과 수직선분이 서로 만나거나 겹치는 경우는 없다.

실행 파일의 이름은 RECT.EXE로 한다. 실행시간은 10초를 넘으면 안된다. 정답이 아니면 0점으로 처리한다.

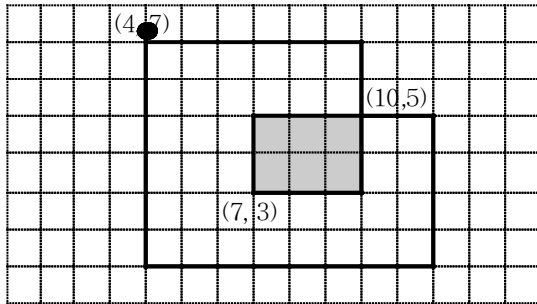
## 입력 형식

입력 파일명은 INPUT.TXT로 한다. 입력 파일의 첫째 줄에는 시작점의 x좌표와 y좌표가 하나의 빈칸을 사이에 두고 주어진다. 그 다음 줄에는 움직인 동작의 횟수  $n$ 이 주어진다.  $n$ 은 100 이하의 정수이다. 그 다음  $n$ 개의 줄에는 움직임의 방향(U, D, R, L)과 거리가 하나의 빈칸을 사이에 두고 주어진다. 로봇의 움직임은 x, y 모두 1 이상 100 이하인 정수 좌표 내에서 이루어진다.

## 출력 형식

출력 파일명은 OUTPUT.TXT로 한다. 가장 작은 면적을 갖는 단순 사각형의 왼쪽 아래 꼭지점의 좌표를 첫째줄에, 오른쪽 위 꼭지점의 좌표를 둘째줄에 출력하면 된다. 좌표의 출력은 x좌표, y좌표 순으로 하며 하나의 빈칸을 사이에 둔다. 가장 작은 면적의 단순 사각형이 여러 개인 경우에는 그 중에서 하나만 출력하면 된다. 만일 단순 사각형이 존재하지 않으면 첫째줄에 0을 출력한다.

### 입력과 출력의 예



위의 그림에 대한 입력과 출력은 아래와 같다.

입력(INPUT.TXT)

```
4 7
8
R 6
D 4
L 3
U 2
R 5
D 4
L 8
U 6
```

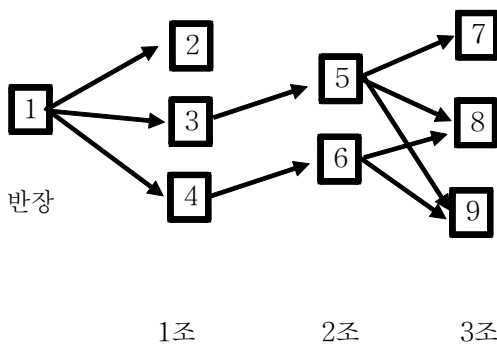
출력(OUTPUT.TXT)

```
7 3
10 5
```

## 비상 연락망

방학 중 비상 연락을 위하여 전체 학생에게 연락할 수 있는 비상 연락망을 구성하였다. 이 연락망의 구성은 반 전체 학생들을 반장을 제외하고  $k$ 개의 조로 나누고, 반장은 1조의 학생들의 전화번호를 모두 가지고 있고, 1조의 각 학생은 2조 학생들 중 일부의 전화 번호를, ...,  $i$ 조의 각 학생은  $(i+1)$ 조 학생들 중 일부의 전화번호를 가지고 있다. 이 연락망을 이용하여 가장 신속한 비상 연락계획을 결정하려고 한다.

비상 연락은 반장으로부터 시작하며 연락을 받은 학생은 자기가 전화번호를 가지고 있는 학생에게 전화할 수 있다. 모든 학생은 정확히 1명으로부터만 전화를 받는다. 단, 전화는 한 통화에 1분이 걸리고, 한 사람이 여러 학생에게 전화할 경우 한 명씩 순차적으로 한다.



위의 그림은 비상 연락망의 예이다. 반장은 1번으로 표시하고, 나머지 학생들은 2부터 일련번호로 표시한다. 그림에서 번호가 주어진 사각형은 해당 학생들을 나타내는 노드들이고 학생  $a$ 가 학생  $b$ 의 전화번호를 가지고 있으면 노드  $a$ 에서 노드  $b$ 로 화살표가 주어진다.

위의 그림과 같이 비상 연락망이 구성되어 있는 경우 아래의 표는 가능한 비상 연락계획 가운데 두 가지를 보여주고 있다. 연락계획 A는 4분만에 연락을 완료하고, 연락계획 B는 5분이 걸린다.

연락계획 A

송신자	수신자	통화시각
1	4	1
1	3	2
1	2	3
3	5	3
4	6	2
5	7	4
6	9	3
6	8	4

연락계획 B

송신자	수신자	통화시각
1	2	1
1	3	2
1	4	3
3	5	3
4	6	4
5	7	4
5	8	5
6	9	5

주어진 연락망을 이용하여 모든 학생들에게 연락할 수 있는 가장 신속한 비상 연락계획을 세우는 프로그램을 작성하시오.

실행 파일의 이름은 PHONE.EXE로 한다. 실행 시간은 10초를 넘으면 안되며, 가장 좋은 답이 아닌 경우라도 조건을 만족하면 부분 점수를 줄 수 있다.

### 입력 형식

입력 파일명은 INPUT.TXT로 한다. 첫째 줄에는 전체 반의 학생 수  $n$ 이 주어진다. 이때  $n$ 은 100이하의 정수이다. 그 다음  $n$ 개의 줄에는 각 줄마다 학생의 번호와 그 학생이 전화번호를 가지고 있는 학생들의 번호가 하나의 빈칸을 사이에 두고 주어진다.

### 출력 형식

출력 파일명은 OUTPUT.TXT로 한다. 첫째 줄에 비상 연락에 걸리는 총 시간을 분 단위로 출력하고 다음 줄부터 순서대로 매분마다 연락이 이루어지는 송신자 번호와 수신자 번호의 순서쌍들을 한 줄에 출력한다.

예를 들어, 연락계획 A에서 통화시각 2에 연락이 이루어지는 쌍은 (1,3)과 (4,6)이다. 이에 대한 출력은 아래와 같이 첫 칸에서 시작하여 하나

의 빈칸을 사이에 두고

1 3 4 6 또는

4 6 1 3

과 같이 출력한다.

### 입력과 출력의 예

앞의 그림에 대한 입력과 연락계획 A에 대한 출력의 예이다.

입력(INPUT.TXT)

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

출력(OUTPUT.TXT)

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