

장난감 조립

우리는 어떤 장난감을 여러 가지 부품으로 조립하여 만들려고 한다. 이 장난감을 만드는 데는 기본 부품과 그 기본 부품들을 조립하여 만든 중간 부품이 사용된다. 기본 부품은 다른 부품을 사용하여 조립될 수 없는 부품이다. 중간 부품은 또 다른 중간 부품이나 기본 부품을 이용하여 만들어지는 부품이다.

예를 들어 보자. 기본 부품으로서 1, 2, 3, 4가 있다. 중간 부품 5는 2개의 기본 부품 1과 2개의 기본 부품 2로 만들어 진다. 그리고 중간 부품 6은 2개의 중간 부품 5, 3개의 기본 부품 3과 4개의 기본 부품 4로 만들어진다. 마지막으로 장난감 완제품 7은 2개의 중간 부품 5, 3개의 중간 부품 6과 5개의 기본 부품 4로 만들어진다. 이런 경우에 장난감 완제품 7을 만드는데 필요한 기본 부품의 개수는 1번 16개, 2번 16개, 3번 9개, 4번 17개이다. 이와 같이 어떤 장난감 완제품과 그에 필요한 부품들 사이의 관계가 주어질 때 하나의 장난감 완제품을 조립하기 위하여 필요한 기본 부품의 종류별 개수를 계산하는 프로그램을 작성하시오.

실행 파일의 이름은 TOY.EXE로 한다. 각 테스트 데이터마다 프로그램의 실행 시간은 10초를 초과할 수 없으며, 각 부품의 수를 모두 정확히 구하지 못하면 0점으로 처리된다.

입력 형식

입력 파일명은 INPUT.TXT로 한다. 입력 파일의 첫째 줄에는 정수 N ($3 \leq N \leq 100$)이 주어지는데, 1부터 $N-1$ 까지는 기본 부품이나 중간 부품의 번호를 나타내고 N 은 완제품의 번호를 나타낸다. 그리고 그 다음 줄에는 정수 M ($3 \leq M \leq 100$)이 주어지고 그 다음 M 개의 줄에는 어떤 부품을 완성하는데 필요한 부품들 간의 관계가 3개의 정수 X, Y, K 로 주어진다. 이 뜻은 중간 부품이나 완제품 X 를 만드는데 필요한 중간 부품 혹은 기본 부품 Y 가 K 개 필요하다는 뜻이다.

출력 형식

출력 파일명은 OUTPUT.TXT로 한다. 하나의 완제품을 조립하는데 필요한 기본 부품의 수를 한 줄에 하나씩 출력하되(중간 부품은 출력하지 않음), 반드시 기본 부품의 번호가 작은 것부터 큰 순서가 되도록 한다. 각 줄에는 기본 부품의 번호와 소요 개수를 출력한다.

입력과 출력의 예

입력(INPUT.TXT)

```

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

```

출력(OUTPUT.TXT)

1 16

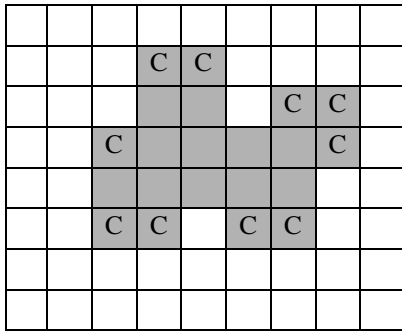
2 16

3 9

4 17

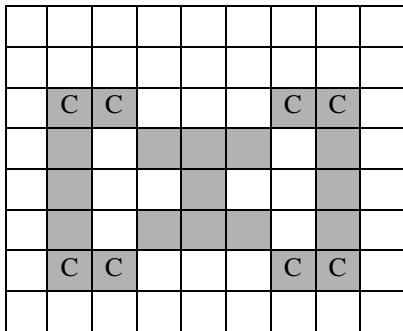
치즈

$N \times M$ ($5 \leq N, M \leq 100$)의 모눈종이 위에 아주 얇은 치즈가 <그림 1>과 같이 표시되어 있다. 단, N 은 세로 격자의 수이고, M 은 가로 격자의 수이다. 이 치즈는 냉동 보관을 해야만 하는데 실내온도에 내어 놓으면 공기와 접촉하여 천천히 녹는다. 그런데 이러한 모눈종이 모양의 치즈에서 각 치즈 격자(작은 정사각형 모양)의 4변 중에서 적어도 2변 이상이 실내온도의 공기와 접촉한 것은 정확히 한시간 만에 녹아 없어져 버린다. 따라서 아래 <그림 1> 모양과 같은 치즈(회색으로 표시된 부분)라면 C 로 표시된 모든 치즈 격자는 한 시간 후에 사라진다.

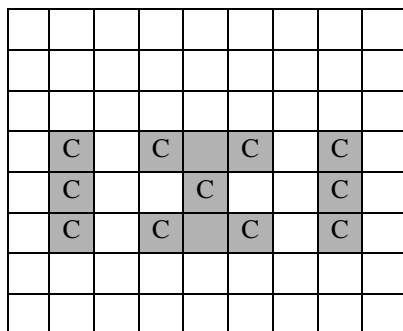


<그림 1>

<그림 2>와 같이 치즈 내부에 있는 공간은 치즈 외부 공기와 접촉하지 않는 것으로 가정한다. 그러므로 이 공간에 접촉한 치즈 격자는 녹지않고 C 로 표시된 치즈 격자만 사라진다. 그러나 한 시간 후, 이 공간으로 외부공기가 유입되면 <그림 3>에서와 같이 C 로 표시된 치즈 격자들이 사라지게 된다.



<그림 2>



<그림 3>

모눈종이의 맨 가장자리에는 치즈가 놓이지 않는 것으로 가정한다.

입력으로 주어진 치즈가 모두 녹아 없어지는데 걸리는 정확한 시간을 구하는 프로그램을 작성하시오.

실행 파일의 이름은 CHEESE.EXE로 한다. 각 테스트 데이터마다 프로그램의 실행 시간은 10초를 초과할 수 없으며, 정확한 시간을 구하지 못하면 0점 처리된다.

입력 형식

입력 파일명은 INPUT.TXT로 한다. 첫째 줄에는 모눈종이의 크기를 나타내는 두 개의 정수 N, M ($5 \leq N, M \leq 100$)이 주어진다. 그 다음 M 줄에는 모눈종이 위의 격자에 치즈가 있는 부분은 1로 표시되고, 치즈가 없는 부분은 0으로 표시된다. 또한, 각 0과 1은 하나의 공백으로 분리되어 있다.

출력 형식

출력 파일명은 OUTPUT.TXT로 한다. 출력으로는 주어진 치즈가 모두 녹아 없어지는데 걸리는 정확한 시간을 정수로 첫줄에 출력한다.

입력과 출력의 예

다음은 위의 <그림 1>에 해당하는 입력 및 출력 데이터의 예이다.

입력(INPUT.TXT)

```
8 9
0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0 0
0 0 0 1 1 0 1 1 0
0 0 1 1 1 1 1 1 0
0 0 1 1 1 1 1 0 0
0 0 1 1 0 1 1 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

출력(OUTPUT.TXT)

```
4
```

주차장

<그림 1>과 같이 정사각형 $N \times N$ 크기의 주차장이 있다. 이 주차장에는 차들이 많이 있는데, 우리 차를 주차장 밖으로 빼내어야 한다. 그런데 차를 움직일 수 있는 주차관리원은 한 명 뿐이다. 이 주차관리원이 하나의 차를 타고 그 차를 움직이는 것을 하나의 “작업”이라 부르기로 한다. 하나의 작업에 차는 앞뒤로 다른 차와 겹치지 않는 한 얼마든지 움직일 수 있다. 다음의 가정 하에 작업의 횟수를 가장 적게하면서 우리 차를 주차장 밖으로 빼내기 위해서 움직여야 하는 모든 차들의 순서를 계산하는 프로그램을 작성하시오. 답이 여러 가지가 나올 수 있는 경우에는 그 중 하나만 출력한다.

가정

- (1) 모든 차의 폭은 1이고, 길이는 2 또는 3이다.
- (2) 차는 주차되어 있는 방향에 따라 수직차(예: <그림 1>의 4번차), 수평차(예: <그림 1>의 3번차)로 구분한다. 수직차는 상하로만, 수평차는 좌우로만 움직일 수 있고 회전은 할 수 없다.
- (3) 같은 차에 대해서 여러 번 작업할 수 있다.
- (4) 우리 차 외에 다른 차는 작업 도중 조금이라도 주차장 밖으로 나갈 수 없으며, 우리 차가 완전히 주차장을 나가는 순간에 전체 작업은 끝난다(<그림 2>).
- (5) 우리 차가 수평차라면 오른쪽으로만 주차장을 나갈 수 있고, 수직차이면 위쪽으로만 나갈 수 있다.
- (6) 입력 데이터에서 우리 차는 반드시 주차장을 빠져 나갈 수 있도록 되어 있다.

실행 파일의 이름은 PARKING.EXE로 한다.

평가 방법

- (1) 각 테스트 데이터마다 실행시간이 20초가 넘는 경우는 0점이다.
- (2) 우리 차 이외에 다른 차가 작업 도중 주차장을 조금이라도 벗어나거나 작업 도중 차들이 서로 겹치는 경우가 발생하면 0점이다.
- (3) (1)과 (2)의 경우가 아니라면 여러분들이 구한 작업 횟수에 따라 부분 점수가 주어진다.

2	0	0	6	6	6
2	0	0	8	0	9
1	1	0	8	4	9
3	3	3	0	4	9
0	0	10	0	7	7
11	11	10	5	5	0

6	6	6	8	4	0		
2	0	0	8	4	0		
2	0	0	0	0	0	1	1
3	3	3	0	0	9		
0	0	10	7	7	9		
11	11	10	5	5	9		

<그림 2> 우리 차를 빼낸 후의 주차장의 모습

입력 형식

입력 파일의 이름은 INPUT.TXT이다. 첫째 줄에는 주차장의 크기를 나타내는 정수 N ($3 \leq N \leq 15$), 다음 N 개의 줄에는 각각 N 개의 숫자가 한칸씩 띄어서 나타나는데 0은 빈 공간을 뜻하고 1은 우리 차를 뜻하며, 나머지 차에는 2부터 연속된 정수가 각각 부여된다.

출력 형식

출력 파일의 이름은 OUTPUT.TXT이다. 첫째 줄에는 총 작업의 횟수를 나타내는 정수 K 를 출력한다. 다음 K 줄은 진행된 작업을 순서대로 출력한다. 한줄이 한 작업을 나타내며 두 개의 정수로 이루어진다.

첫째 정수는 움직인 차의 번호, 둘째 정수는 차가 움직인 거리를 나타낸다. 단, 거리는 격자의 개수를 의미한다. 수평차인 경우 오른쪽 방향으로 움직인 거리를 양의 정수, 왼쪽 방향으로 움직인 거리를 음의 정수로 나타낸다. 수직차이면 위쪽 방향으로 움직인 거리를 양의 정수, 아래쪽 방향으로 움직인 거리를 음의 정수로 나타낸다.

입력과 출력의 예

입력(INPUT.TXT)

6
2 0 0 6 6 6
2 0 0 8 0 9
1 1 0 8 4 9
3 3 3 0 4 9
0 0 10 0 7 7
11 11 10 5 5 0

출력(OUTPUT.TXT)

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