

첨부 5

제 16 회
한 국 정 보 올 림 피 아 드

경 시 문 제

촌수 계산

우리 나라는 가족 혹은 친척들 사이의 관계를 촌수라는 단위로 표현하는 독특한 문화를 가지고 있다. 이러한 촌수는 다음과 같은 방식으로 계산된다. 기본적으로 부모와 자식 사이를 1촌으로 정의하고 이로부터 사람들 간의 촌수를 계산한다. 예를 들면 나와 아버지, 아버지와 할아버지는 각각 1촌이므로 나와 할아버지는 2촌이 되고, 아버지 형제들과 할아버지는 1촌이므로 나와 아버지 형제들과는 3촌이 된다. 여러 사람들에 대한 부모 자식들 간의 관계가 주어졌을 때, 주어진 두 사람의 촌수를 계산하는 프로그램을 작성하시오.

실행 파일의 이름은 FAMILY.EXE로 하고, 프로그램의 실행시간은 10초를 초과할 수 없다. 정확한 답에 대하여서는 만점으로 채점하고, 틀린 답에 대하여서는 0점으로 채점한다.

입력 형식

입력 파일명은 INPUT.TXT로 한다. 사람들은 $1, 2, 3, \dots, n$ ($1 \leq n \leq 100$)의 연속된 번호로 각각 표시된다. 입력 파일의 첫째 줄에는 전체 사람의 수 n 이 주어지고, 둘째 줄에는 촌수를 계산해야 하는 서로 다른 두 사람의 번호가 주어진다. 그리고 셋째 줄에는 부모 자식들 간의 관계의 개수 m 이 주어진다. 넷째 줄부터는 부모 자식간의 관계를 나타내는 두 번호 $x y$ 가 각 줄에 나온다. 이때 앞에 나오는 번호 x 는 뒤에 나오는 정수 y 의 부모 번호를 나타낸다.

출력 형식

출력 파일명은 OUTPUT.TXT로 한다. 입력 파일에서 요구한 두 사람의 촌수를 나타내는 정수를 출력한다. 어떤 경우에는 두 사람간의 친척 관

계가 전혀 없어 촌수를 계산할 수 없을 때가 있다. 이 때에는 -1을 출력해야한다.

입력과 출력의 예(1)

다음은 9명의 사람으로 구성된 촌수 관계에서 7번과 3번의 촌수를 구하는 입력 데이터의 예이다.

입력(INPUT.TXT)

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

출력(OUTPUT.TXT)

```
3
```

입력과 출력의 예(2)

입력(INPUT.TXT)

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

출력(OUTPUT.TXT)

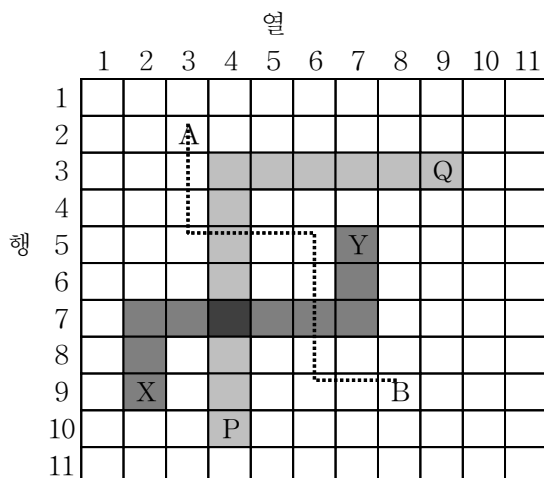
```
-1
```

회로 배치

회로를 $n \times n$ 의 격자 판에 배치하려고 한다. 여기서 각 격자(정사각형 칸)는 가장자리에 있는 격자를 제외하고 상, 하, 좌, 우 4개의 이웃 격자를 갖는다. 회로는 시작과 끝이 있는 연속된 이웃 격자들의 길(path)이다. 아래 그림에서는 X와 Y, P와 Q를 연결한 두 개의 회로가 있다. 이미 회로들이 배치되어 있는 격자 판에 새로 배치할 회로의 양 끝 격자가 주어질 때, 이들 두 격자를 잇는 새로운 회로를 배치하려고 한다.

새로 배치될 회로는 이미 회로가 배치된 격자 위에 배치될 수도 있다. 이 회로의 배치 비용은 이 회로가 지나가는 격자에 따라 다음과 같이 결정된다. 회로가 배치되지 않은 빈 격자를 지나가는 비용은 1이고, 이미 회로가 놓여있는 격자를 지날 때는 비용이 k ($k \geq 2$)이다. 주어진 문제는 최소의 비용이 소요되는 새로운 회로를 찾는 것이다.

예를 들어 아래의 그림에서 k 가 4로 주어진다면, 점선을 따라 격자 A와 B를 잇는 회로의 비용은 19이지만, 비용이 16인 최소비용 회로가 존재한다. (이 비용에는 격자 A, B의 비용도 포함된다.)



실행 파일의 이름은 CIRCUIT.EXE로 하고, 프로그램의 실행 시간은 10초를 초과할 수 없다. 최소 비용을 구하지 못하는 경우는 0점으로 처리된다.

입력 형식

입력 파일명은 INPUT.TXT로 한다. 입력 파일의 첫째 줄에는 격자 판의 크기를 나타내는 정수 ($2 \leq n \leq 50$)가 주어진다. 둘째 줄에는 새로 배치할 회로의 시작 격자, 마지막 격자의 위치를 나타내는 4개의 정수가 주어진다. 한 격자의 위치는 위 그림에서 주어진 행과 열의 번호 순서로 주어진다. (시작 격자와 마지막 격자의 위치는 같을 수 없다.) 셋째 줄에는 회로가 배치된 격자를 지나는데 드는 비용인 정수 k 가 주어진다. 넷째 줄에는 이미 배치된 회로의 개수가 주어진다. 다섯째 줄부터는 한 줄에 한 회로의 배치 정보가 다음과 같이 주어진다. 첫째 정수는 회로의 시작 격자, 90°로 꺾이는 방향 전환 격자들, 그리고 마지막 격자의 총 개수이고, 그 다음부터는 이들 격자의 위치가 시작 격자부터 마지막 격자까지 행과 열의 순서대로 주어진다.

출력 형식

출력 파일명은 OUTPUT.TXT로 한다. 첫째 줄에는 회로의 최소 비용을 출력한다. 둘째 줄에는 최소비용 회로의 정보를 다음과 같이 출력한다. (입력 형식과 동일함.) 처음에 회로의 시작 격자, 90°로 꺾이는 방향 전환 격자들, 그리고 마지막 격자의 총 개수를 출력하고, 그 다음부터는 이들 격자의 위치를 시작 격자부터 마지막 격자까지 행과 열의 순서대로 한 개씩 공백을 두고 출력한다.

입력과 출력의 예

앞 그림에 해당되는 입력과 출력은 다음과 같다.

입력(INPUT.TXT)

```
11
2 3 9 8
4
2
3 3 9 3 4 10 4
4 9 2 7 2 7 7 5 7
```

출력(OUTPUT.TXT)

```
16
3 2 3 2 8 9 8
```

같은 길이 막대기 만들기

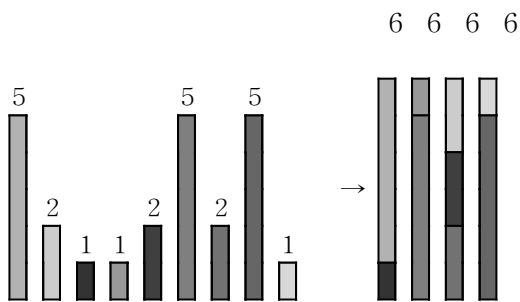
승연이는 길이가 정수인 동일한 막대기를 여러 개 가지고 있는데, 이 막대기들을 각각 정수의 길이를 갖는 여러 개의 토막으로 아무렇게나 잘랐다. 단, 어떤 막대기는 자르지 않을 수도 있다.

승연이는 이렇게 잘려진 토막들을 다시 붙여서 원래 상태의 막대기로 만들려고 하는데 원래 자기가 가지고 있던 막대기의 수와 막대기의 길이를 잊어버렸다. 그래서 승연이는 길이가 모두 같은 가장 짧은 막대기들로 복원하기로 하였다.

문제는 잘려진 모든 토막으로부터 구성될 수 있는 같은 길이의 막대기 중에서 가장 짧은 길이를 계산하는 프로그램을 작성하는 것이다.

아래 <그림>의 예에서 보듯이 같은 막대기 몇 개를 잘라서 길이가 { 5, 2, 1, 1, 2, 5, 2, 5, 1 }인 토막으로 만들었다. 문제는 이 토막들로부터 구할 수 있는 같은 길이의 막대기 중에서 가장 짧은 것을 구하는 것이다.

아래 그림의 예에서 길이가 12인 긴 막대기 2개를 만들 수도 있으나 가장 짧은 동일한 막대기들의 길이는 6이 된다.



<그림>

실행 파일의 이름은 STICK.EXE로 하고, 프로그램의 실행시간은 10초를 초과할 수 없다. 조건을 만족하지 못하는 답은 0점 처리되며, 조건을

만족하는 답의 점수는 가장 좋은 답과 본인의 답과의 차이에 따라 적절히 주어진다.

입력 형식

입력 파일명은 INPUT.TXT로 한다. 입력은 두 줄로 구성된다. 입력 파일의 첫째 줄에는 잘려진 토막들의 총 수를 나타낸다. 이 수는 최대 50이다. 두 번째 줄에는 작게 잘라진 토막들의 각 길이를 나타내는 정수가 나열된다. 각 토막의 길이를 나타내는 숫자 사이에는 하나씩의 공백이 있다. 각 토막의 길이 l 은 $2 \leq l \leq 1000$ 인 정수이다.

출력 형식

출력 파일명은 OUTPUT.TXT로 한다. 첫째 줄에 복구된 막대기들의 길이를 나타내는 정수를 출력한다. 그 다음 줄에는 각 토막들이 연결된 상태를 아래 출력 파일의 예와 같이 각 줄에 임의의 순서로 표시한다.

입력과 출력의 예

<그림>에 해당되는 입력과 출력은 다음과 같다.

입력(INPUT.TXT)

```
9
5 2 1 1 2 5 2 5 1
```

출력(OUTPUT.TXT)

```
6
5 1
1 5
2 2 2
1 5
```