수열 축소

n개의 정수들로 이루어진 수열 $[a_1, a_2, \ldots, a_n]$ 이 있다. 여기에 정수 i 를 주면서 '축소' 라는 명령을 하면 a_i 와 a_{i+1} 이 없어지고, 대신 두 수의 차가 들어간다. 즉, 새로 들어간 수는 $|a_i - a_{i+1}|$ 이다. 바꿔 말하면, 축소 연산 i 는

축소$([a_1, a_2, \ldots, a_n], i) =$

$[a_1, \ldots, a_{i-1}, |a_i - a_{i+1}|, a_{i+2}, \ldots, a_n]$

이다. 단, $1 \le i \le n-1$.

예를 들어, 수열 [12, 10, 4, 3, 5]에 축소연산들 2, 3, 2, 1을 차례대로 적용하면

축소([12, 10, 4, 3, 5], 2) = [12, 6, 3, 5]
축소([12, 6, 3, 5], 3) = [12, 6, 2]
축소([12, 6, 2], 2) = [12, 4]
축소([12, 4], 1) = [8] 이 되므로,
마지막에 8이 남는다.

같은 수열에 축소연산들 1, 1, 1, 1을 적용하면,

축소([12, 10, 4, 3, 5], 1) = [2, 4, 3, 5]
축소([2, 4, 3, 5], 1) = [2, 3, 5]
축소([2, 3, 5], 1) = [1, 5]
축소([1, 5], 1) = [4] 가 되므로,
마지막에 4가 남는다.

입력으로 수열과 최종값이 주어질 때, n-1 번의 축소 연산을 수행하여 최종값을 얻을 수 있는가를 알아내는 프로그램을 작성하시오. 최종값을 얻을 수 있는 축소연산들이 여러개일 경우는 그 중 하나만 출력한다.

실행파일의 이름은 SEQ.EXE 로 한다. 실행 시간은 2초를 넘으면 안되고, 부분점수는 없다.

입력 형식

입력 파일 이름은 INPUT.TXT이다. 첫줄에는 정수 n 이 주어진다 (n 은 1이상 30이하). 둘째 줄에는 최종값이 주어진다. 다음 n 줄에는 $a_1, a2, \ldots, a_n$ 에 해당하는 수가 한 줄에 하나씩 순서대로 주어진다 (각 a_i 는 1이상 30이하).

출력 형식

출력 파일 이름은 OUTPUT.TXT이다. 첫 줄부터 구한 축소연산들을 한 줄에 하나씩 순서대로 출력한다. 최종값을 구할 수 없는 경우에는 첫줄에 0을 출력한다.

입력과 출력의 예(1)

입력 (INPUT.TXT)

```
5
8
12
10
4
3
5
```

출력 (OUTPUT.TXT)

```
2
3
2
1
```

입력과 출력의 예(2)

입력 (INPUT.TXT)

```
4
7
12
10
4
5
```

출력 (OUTPUT.TXT)

```
0
```

유전자

최근에 인간의 유전자 염기서열을 밝히는 인간게놈(genome) 프로젝트가 완료되었다. 염색체내의 유전자의 염기서열을 알아내는 방법은 다음과 같다. 한 염색체의 염기서열이 너무 길면 현재의 기술로는 그 내용을 알아낼 수 없다. 따라서, 이를 PCR이라는 방식을 이용하여 여러 벌로 복제한 다음, 효소를 가하게 되면 염색체 조각들로 나뉘어진다. 이 염색체 조각들은 길이가 충분히 짧으므로 현재의 기술로 그 내용을 밝혀 낼 수 있다. 내용이 밝혀진 염색체 조각들을 다시 이어서 여러 벌의 동일한 염색체로 복구하여, 원래 염색체의 염기서열을 알아낸다. (유전자의 염기는 A, G, T, C라는 네 종류가 있으며, 이들의 조합으로 염기서열을 표시한다.)

예를 들어, 염색체의 염기서열이 있을 때, 이 염기서열을 3벌로 복제한 뒤, 효소를 가하여 11개로 나누고, 그 조각들의 내용을 밝힌 결과가 다음과 같다고 하자.

CGATGCCA
CAGGAAGCG
AGGTGCCC
CAGGA
AGGTGCCCGATGC
GGAAGCGATGGAGCTTT
ATGGAGCTTT
GATGC
CCGTGGA
AGCGATGG
TTTCGA

이 조각들을 다시 모아서 동일한 3벌의 염기서열로 재구성하면 다음과 같이 되고, 이를 통해서 전체 염색체의 염기서열을 알아낼 수 있다. 조각들은 그대로 사용할 수도 있고, 뒤집어서 사용할 수도 있다. 예를 들어 위의 마지막 조각인 TTTCGA는 AGCTTT로 뒤집어서 아래의 마지막 염기서열을 만드는데 사용하였다.

AGGTGCCCGATGC CAGGAAGCG ATGGAGCTTT
AGGTGCC CGATGCCA GGAAGCGATGGAGCTTT
AGGTGCCC GATGC CAGGA AGCGATGG AGCTTT

복제된 염색체의 벌 수와 내용이 밝혀진 염색체 조각들이 입력으로 주어져 있을 때 전체 염기서열을 구하는 프로그램을 작성하시오.

실행 파일 이름은 GENOME.EXE로 하고 실행시간은 5초 이내로 한다. 부분 점수는 없다.

입력 형식

입력 파일 이름은 INPUT.TXT로 한다. 입력 파일의 첫 줄에는 복제한 염기서열의 벌 수 k (k 는 2이상 20이하)가 주어진다. 그 다음 줄에는 염색체 조각 수 n 이 주어진다(n 은 200이하). 다음 n 개의 줄에 염색체 조각들이 한 줄에 하나씩 주어진다(각 염색체 조각의 길이는 5이상 70이하). 주어진 염색체의 조각에 열거된 순서대로 1번부터 n 번까지의 번호를 부여한다. 즉, 앞의 예에서 염색체 조각 CAGGA의 번호는 4이다. 전체 염기서열의 길이는 30이상 700이하이다.

출력 형식

출력 파일 이름은 OUTPUT.TXT로 한다. 출력 파일에는 알아낸 k 벌의 염기서열을 구성하는 순서대로 그 조각 번호들을 한 줄에 한 벌씩 출력한다. 만일 어떤 염색체 조각이 뒤집어서 사용된 경우는 그 조각 번호의 음의 값을 출력한다. 조각 번호 사이에는 빈 칸을 하나 씩 둔다. 알아낸 염기서열의 시작은 염색체 양 끝 어느 쪽에서 시작해도 무방하며 경우에 따라서 답은 하나 이상일 수가 있는데 이 경우에는 하나만 출력하면 된다. 그리고 출력되는 k 벌의 염기서열들의 순서는 상관없다. 단, 출력된 조각번호 순서대로 구성된 k 벌의 염기서열들은 앞 뒤 순서가 모두 같아야 한다.

입력과 출력의 예

입력(INPUT.TXT)

```
3
11
CGATGCCA
CAGGAAGCG
AGGTGCCC
CAGGA
AGGTGCCCGATGC
GGAAGCGATGGAGCTTT
ATGGAGCTTT
GATGC
CCGTGGA
AGCGATGG
TTTCGA
```

출력(OUTPUT.TXT)

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

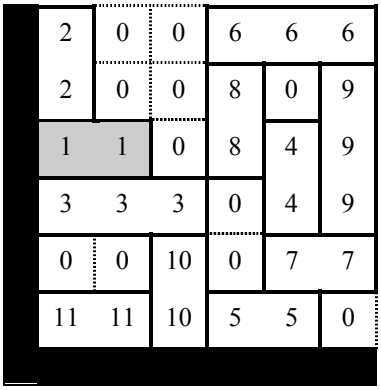
주차장

<그림 1>과 같이 정사각형 *N*×*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 않는한 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.

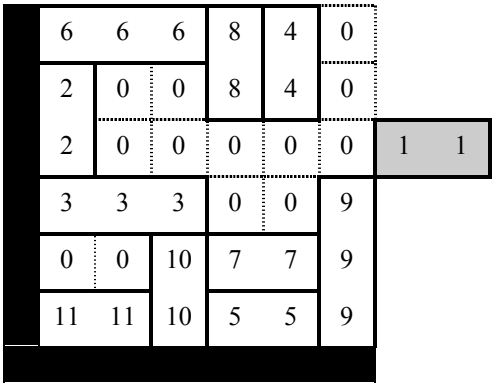
가정

(1) 모든 차의 폭은 1이고, 길이는 2 또는 3 이다.
(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) 우리 차가 주차장을 빠져나갈 수 없는 경우도 있을 수 있다.

실행 파일 이름은 PARKING.EXE로 한다.



<그림 1> 초기 주차장의 모습



<그림 2> 우리 차를 빼낸 후의 주차장의 모습

입력 형식

입력 파일의 이름은 INPUT.TXT 이다. 첫째 줄에는 주차장의 크기를 나타내는 정수 N ($3 \le N \le 15$), 다음 N개의 줄에는 각각 N개의 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.TXT 이다. 첫째 줄에는 총 작업의 횟수를 나타내는 정수 K를 출력한다. 다음 K줄은 진행된 작업을 순서대로 출력한다. 한줄이 한 작업을 나타내며 두 개의 정수로 이루어진다. 첫째 정수는 움직인 차의 번호, 둘째 정수는 차가 움직인 거리를 나타낸다. 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.

만약 차가 주차장에서 빠져나갈 수 없으면 첫 줄에 0을 출력한다.

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
```

평가방법

(1) 각 테스트 데이터마다 실행시간이 20초가 넘는 경우는 0점이다.

(2) 우리 차 이외에 다른 차가 작업 도중 주차장을 조금이라도 벗어나거나 작업 도중 차들끼리 서로 겹치는 경우가 발생하면 0점이다.

(3) (1)과 (2)의 경우가 아니라면 여러분들이 구한 작업 횟수에 따라 부분 점수가 주어질 수 있다.