

## 수 이어가기

다음과 같은 규칙에 따라 수들을 만들려고 한다.

**규칙 1.** 첫 번째 수로 양의 정수가 주어진다.

**규칙 2.** 두 번째 수는 양의 정수 중에서 하나를 선택한다.

**규칙 3.** 세 번째부터 이후에 나오는 모든 수는 앞의 앞의 수에서 앞의 수를 빼서 만든다. 예를 들어, 세 번째 수는 첫 번째 수에서 두 번째 수를 뺀 것이고, 네 번째 수는 두 번째 수에서 세 번째 수를 뺀 것이다.

**규칙 4.** 음의 정수가 만들어지면, 이 음의 정수를 버리고 더 이상 수를 만들지 않는다.

첫 번째 수로 100이 주어질 때, 두 번째 수로 60을 선택하여 위의 규칙으로 수들을 만들면 7개의 수들 100, 60, 40, 20, 20, 0, 20이 만들어진다. 그리고 두 번째 수로 62를 선택하여 위의 규칙으로 수들을 만들면 8개의 수들 100, 62, 38, 24, 14, 10, 4, 6이 만들어진다. 위의 예에서 알 수 있듯이, 첫 번째 수가 같더라도 두 번째 수에 따라 만들어지는 수들의 개수가 다를 수 있다.

입력으로 첫 번째 수가 주어질 때, 이 수에서 시작하여 위의 규칙으로 만들어지는 최대 개수의 수들을 구하는 프로그램을 작성하시오. 최대 개수의 수들이 여러 개일 때, 그 중 하나의 수들만 출력하면 된다.

실행파일의 이름은 SEQ.EXE로 하고, 프로그램의 실행시간은 10초를 초과할 수 없다. 여러분들이 구한 수들의 개수에 따라 부분점수가 주어질 수 있다.

## 입력 형식

입력 파일명은 INPUT.TXT로 한다. 입력 파일에 첫 번째 수가 주어진다. 이 수는 30000보다 같거나 작은 양의 정수이다.

## 출력 형식

출력 파일명은 OUTPUT.TXT로 한다. 첫째 줄에는 입력된 첫 번째 수로 시작하여 위의 규칙에 따라 만들 수 있는 수들의 최대 개수를 출력한다. 둘째 줄에 그 최대 개수의 수들을 차례대로 출력한다. 이들 수 사이에는 빈칸을 하나씩 둔다.

## 입력과 출력의 예

입력(INPUT.TXT)

100
-----

출력(OUTPUT.TXT)

8
100 62 38 24 14 10 4 6

## 스위치 상태

1부터 연속적으로 번호가 붙어있는 스위치들이 있다. 스위치는 켜져 있거나 꺼져있는 상태이다. <그림 1>에 스위치 8개의 상태가 표시되어 있다. '1'은 스위치가 켜져 있음을, '0'은 꺼져 있음을 나타낸다. 그리고 학생 몇 명을 뽑아서, 학생들에게 1 이상이고 스위치 개수 이하인 자연수를 하나씩 나누어주었다. 학생들은 자신의 성별과 받은 수에 따라 아래와 같은 방식으로 스위치를 조작하게 된다.

남학생은 스위치 번호가 자기가 받은 수의 배수이면, 그 스위치의 상태를 바꾼다. 즉, 스위치가 켜져 있으면 끄고, 꺼져 있으면 켜는다. <그림 1>과 같은 상태에서 남학생이 3을 받았다면, 이 학생은 <그림 2>와 같이 3번, 6번 스위치의 상태를 바꾼다.

여학생은 자기가 받은 수와 같은 번호가 붙은 스위치를 중심으로 좌우가 대칭이면서 가장 많은 스위치를 포함하는 구간을 찾아서, 그 구간에 속한 스위치의 상태를 모두 바꾼다. 이 때 구간에 속한 스위치 개수는 항상 홀수가 된다.

예를 들어 <그림 2>에서 여학생이 3을 받았다면, 3번 스위치를 중심으로 2번, 4번 스위치의 상태가 같고 1번, 5번 스위치의 상태가 같으므로, <그림 3>과 같이 1번부터 5번까지 스위치의 상태를 모두 바꾼다. 만약 <그림 2>에서 여학생이 4를 받았다면, 3번, 5번 스위치의 상태가 서로 다르므로 4번 스위치의 상태만 바꾼다.

스위치 번호	①	②	③	④	⑤	⑥	⑦	⑧
스위치 상태	0	1	0	1	0	0	0	1

<그림 1>

스위치 번호	①	②	③	④	⑤	⑥	⑦	⑧
스위치 상태	0	1	1	1	0	1	0	1

<그림 2>

스위치 번호	①	②	③	④	⑤	⑥	⑦	⑧
스위치 상태	1	0	0	0	1	1	0	1

<그림 3>

입력으로 스위치들의 처음 상태가 주어지고, 각 학생의 성별과 받은 수가 주어진다. **학생들은 입력되는 순서대로** 자기의 성별과 받은 수에 따라 스위치의 상태를 바꾸었을 때, 스위치들의 마지막 상태를 출력하는 프로그램을 작성하시오.

실행파일의 이름은 SWITCH.EXE로 하고, 프로그램의 실행시간은 10초를 초과할 수 없다.

## 입력 형식

입력 파일명은 INPUT.TXT로 한다. 첫째 줄에는 스위치 개수가 주어진다. 스위치 개수는 100 이하인 양

의 정수이다. 둘째 줄에는 각 스위치의 상태가 주어진다. 켜져 있으면 1, 꺼져있으면 0이라고 표시하고 사이에 빈칸이 하나씩 있다. 셋째 줄에는 학생수가 주어진다. 학생수는 100 이하인 양의 정수이다.

## 제17회 한국정보올림피아드 (2000.7.15)

## 초 등 부 문 제 2

넷째 줄부터 마지막 줄까지 한 줄에 한 학생의 성별, 학생이 받은 수가 주어진다. 남학생은 1로, 여학생은 2로 표시하고, 학생이 받은 수는 스위치 개수 이하인 양의 정수이다. 학생의 성별과 받은 수 사이에 빈칸이 하나씩 있다.

### 출력 형식

출력 파일의 이름은 OUTPUT.TXT이다. 스위치의 상태를 1번 스위치에서 시작하여 마지막 스위치까지 한 줄에 20개씩 출력한다. 예를 들어 21번 스위치가 있다면 이 스위치의 상태는 둘째 줄 맨 앞에 출력한다. 켜진 스위치는 1, 꺼진 스위치는 0으로 표시하고, 스위치 상태 사이에 빈칸을 하나씩 둔다.

### 입력과 출력의 예

입력(INPUT.TXT)

```
8
0 1 0 1 0 0 0 1
2
1 3
2 3
```

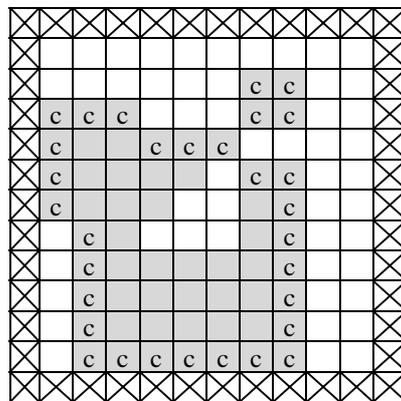
출력(OUTPUT.TXT)

```
1 0 0 0 1 1 0 1
```

치즈

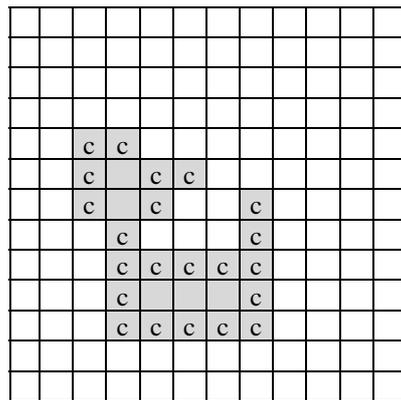
아래 <그림 1>과 같이 정사각형 칸들로 이루어진 사각형 모양의 판이 있고, 그 위에 얇은 치즈(회색으로 표시된 부분)가 놓여 있다. 판의 가장자리(<그림 1>에서      부분)에는 치즈가 놓여 있지 않으며 치즈에는 하나 이상의 구멍이 있을 수 있다.

이 치즈를 공기 중에 놓으면 녹게 되는데 공기와 접촉된 칸은 한 시간이 지나면 녹아 없어진다. 치즈의 구멍 속에는 공기가 없지만 구멍을 둘러싼 치즈가 녹아서 구멍이 열리면 구멍 속으로 공기가 들어가게 된다. <그림 1>의 경우, 치즈의 구멍을 둘러싼 치즈는 녹지 않고 'c'로 표시된 부분만 한 시간 후에 녹아 없어져서 <그림 2>와 같이 된다.

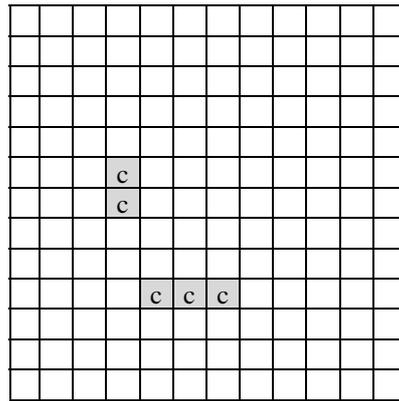


<그림 1> 원래 치즈 모양

다시 한 시간 후에는 <그림 2>에서 'c'로 표시된 부분이 녹아 없어져서 <그림 3>과 같이 된다.



<그림 2> 한 시간 후의 치즈 모양



<그림 3> 두 시간 후의 치즈  
모양

<그림 3>은 원래 치즈의 두 시간 후 모양을 나타내고 있으며, 남은 조각들은 한 시간이 더 지나면 모두 녹아 없어진다. 그러므로 처음 치즈가 모두 녹아 없어지는 데는 세 시간이 걸린다. <그림 3>과 같이 치즈가 녹는 과정에서 여러 조각으로 나누어 질 수도 있다.

입력으로 사각형 모양의 판의 크기와 한 조각의 치즈가 판 위에 주어진 때, 공기 중에서 치즈가 모두 녹아 없어지는 데 걸리는 시간과 모두 녹기 한 시간 전에 남아있는 치즈조각이 놓여 있는 칸의 개수를 구하는 프로그램을 작성하시오.

실행파일의 이름은 **CHEESE.EXE**로 하고, 프로그램의 실행시간은 10초를 초과할 수 없다.

### 입력 형식

입력 파일명은 **INPUT.TXT**로 한다. 입력 파일의 첫째 줄에는 사각형 모양 판의 세로와 가로 길이가 양의 정수로 주어진다. 세로와 가로 길이는 최대 100이다. 판의 각 가로줄의 모양이 첫 줄부터 차례로 입력 파일의 둘째 줄부터 마지막 줄까지 주어진다. 치즈가 없는 칸은 0, 치즈가 있는 칸은 1로 주어지며 각 숫자 사이에는 빈칸이 하나씩 있다.

### 출력 형식

출력 파일명은 **OUTPUT.TXT**로 한다. 첫째 줄에는 치즈가 모두 녹아서 없어지는 데 걸리는 시간을 출력하고, 둘째 줄에는 모두 녹기 한 시간 전에 남아있는 치즈조각이 놓여 있는 칸의 개수를 출력한다.

입력과 출력의 예

입력(INPUT.TXT)

```
13 12
000000000000
000000000000
000000011000
011100011000
011111100000
011111011000
011110011000
001100011000
001111111000
001111111000
001111111000
001111111000
000000000000
```

출력(OUTPUT.TXT)

```
3
5
```