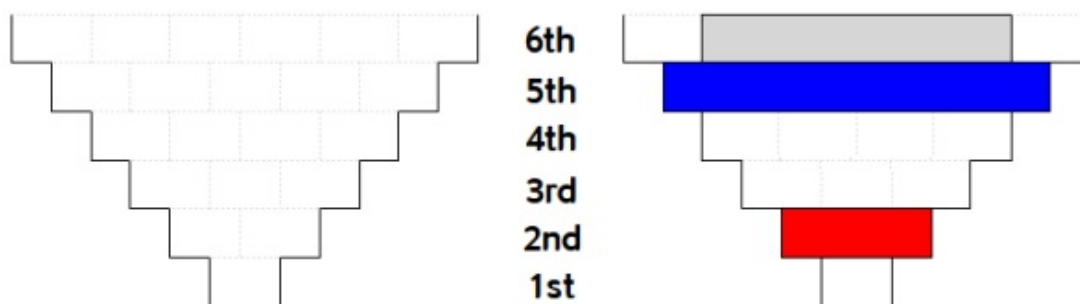


2-1. On the Grid

정국이는 아래 그림과 같은 삼각형 격자 모양 통을 갖고 있다. 통의 층수는 충분히 많아서 막대를 넣기에 충분하다. 평소에는 이 통에 길이가 $1, 2, \dots, N$ 인 막대가 차곡차곡 쌓여있지만, 오늘은 정국이의 생일이라 N 명의 친구들이 각자 하나의 막대를 가져가서 놓고 있다. 각 친구에는 0 이상 $N - 1$ 이하의 정수 번호가 붙어 있다. i 번 친구는 ($0 \leq i \leq N - 1$)는 길이가 $L[i]$ 인 막대를 가져갔다.



친구들은 정국이와 재미있는 게임을 하려고 한다. 게임은 여러 라운드로 구성되며 한 라운드의 규칙은 다음과 같다.

- 정국이가 N 명의 친구들을 임의 순서로 건드리면 그 친구는 바로 자기 막대를 통에 넣는다.
 - 막대는 격자 통 중앙으로 떨어진다.
 - 막대는 격자 통과 닿거나 다른 막대와 닿을 때까지 떨어진다.
- 정국이가 N 명의 친구들을 모두 건드리고 나면 정국이가 마지막 막대가 놓인 층수를 확인한다.
- N 명의 친구들이 자기 막대를 도로 가져간다. 한 라운드가 끝나면 격자 통에는 막대가 없다.

정국이가 게임을 진행하다 보면 N 명의 친구들이 각각 어떤 막대를 갖고 갔는지 알 수 있다. 정국이를 도와 친구들을 최소한으로 건드리어서 N 명의 친구들이 가지고 간 막대의 길이를 구하는 프로그램을 작성하라.

요구 사항

다음 함수를 구현해야 한다. 각 테스트 케이스에 대해 그레이더는 이 함수를 한 번 호출한다.

```
int[] SortDisks(int N)
```

- N : 친구의 수.
- 이 함수는 길이 N 의 배열 X 를 반환한다. $X[i]$ ($0 \leq i \leq N - 1$)의 값은 i 번 친구가 갖고 있는 막대의 길이를 의미하며 $L[i]$ 와 같아야 한다.
- 이 함수는 한 번 호출된다.

SortDisks 함수 안에서 아래 함수를 호출할 수 있다.

```
int PutDisks(int[] P)
```

- P : 길이 N 의 배열. P 는 $0, 1, \dots, N - 1$ 의 순열이어야 한다. $P[i]$ ($0 \leq i \leq N - 1$)는 정국이가 $(i + 1)$ 번째로 건드리는 친구의 번호를 의미한다.
- 이 함수는 마지막 막대가 놓인 층수를 반환한다.
- 이 함수는 최대 2000번 호출할 수 있다.

위 조건을 만족하지 않거나 SortDisks의 반환값이 올바르지 않으면, 당신의 프로그램은 Wrong Answer으로 채점된다. 나머지 경우에는 당신의 프로그램은 Accepted으로 채점된다.

제한

- $1 \leq N \leq 1000$
- $1 \leq L[i] \leq N$ ($0 \leq i \leq N - 1$)
- $L[i] \neq L[j]$ ($0 \leq i < j \leq N - 1$)

이 문제에서 그레이더는 적응적이지 않다 (NOT adaptive). 이것은 막대의 길이가 그레이더의 수행 초기에 고정되어서 당신이 요청하는 쿼리에 따라 바뀌지 않는다는 것을 의미한다.

부분문제

1. (12점) $N \leq 50$.
2. (31점) $N \leq 250$.
3. (57점) 추가 제약조건은 없다.

예제

L 의 값이 $[1, 4, 2, 3]$ 라고 하자. 다음 호출이 발생한다.

```
SortDisks(4)
```

SortDisks 함수에서 아래와 같이 PutDisks 함수를 호출했을 때 결과는 다음과 같다.

- `PutDisks([0, 2, 3, 1])`: 길이 1, 2, 3, 4인 막대가 순서대로 놓여진다. 각 막대는 1층, 2층, 3층, 4층에 떨어지므로 함수는 4를 반환한다.
- `PutDisks([2, 1, 0, 3])`: 길이 2, 4, 1, 3인 막대가 순서대로 놓여진다. 각 막대는 2층, 4층, 5층, 6층에 떨어지므로 함수는 6을 반환한다.

정답은 $[1, 4, 2, 3]$ 이다.

샘플 인터페이스

문제 페이지에서 샘플 코드를 다운로드받을 수 있다. 만약 Visual Studio나 Eclipse, Code::Blocks와 같은 IDE 툴을 사용한다면 `grid.cpp`, `grid.h`, `grader.cpp`를 한 프로젝트에 넣어서 컴파일하면 된다. 터미널에서 코드를 컴파일한다면 대회 페이지에 있는 컴파일 명령어를 이용하면 된다.

답안을 제출할 때에는 `grid.cpp`를 제출하면 된다.

Input format

- line 1: N
- line 2: $L[0] \ L[1] \ \dots \ L[N - 1]$

Output format

당신의 프로그램이 **Accepted**으로 채점되었다면, 샘플 그레이더는 첫 번째 줄에 **Correct**를 출력하고 두 번째 줄에 `PutDisks` 함수의 호출 횟수를 출력한다.

당신의 프로그램이 **Wrong Answer**으로 채점되었다면, 샘플 그레이더는 첫 번째 줄에 에러 메시지를 출력한다.