문제2 : 하나 둘 셋

길이가 N 인 배열 A 가 있다. 이 배열은 1, 2, 3 세 값으로만 이루어져 있다. 우리는 이 배열에서 다음 조건을 만족하는 세 인덱스의 순서쌍 (i, j, k) 들을 최대한 많이 찾으려고 한다. 배열의 세 인덱스 i, j, k ($0 \leq i < j < k < N$)에 대해서 $A[i] = 1, A[j] = 2, A[k] = 3$ 이거나, $A[i] = 3, A[j] = 2, A[k] = 1$ 이어야 한다. 단, 한 인덱스는 최대 한 개의 순서쌍에만 들어갈 수 있다.

예를 들어 $A = \{1, 2, 3, 2, 3, 1\}$ 이 주어졌다고 하자. 조건을 만족하는 답은 $(0, 1, 4), (2, 3, 5)$ 가 된다. ($A[0] = 1, A[1] = 2, A[4] = 3$ 이고 $A[2] = 3, A[3] = 2, A[5] = 1$)

A 가 주어졌을 때, 조건을 만족하는 순서쌍을 최대한 많이 찾아서 보고하는 프로그램을 작성하라.

여러분은 다음 함수를 작성하여야 한다.

- `void maximize( vector<int> A )` : A 는 길이 N 인 vector로, 1, 2, 3 세 값으로만 이루어져 있다. `maximize`는 A 에서 문제의 조건에 맞는 순서쌍 (i, j, k) 들을 최대한 많이 찾아내고, 찾아낸 (i, j, k) 하나마다 `grader`의 `answer(int i, int j, int k)` 함수를 정확하게 한 번 호출한다. 최대 개수의 순서쌍들을 찾는 방법이 여러 가지인 경우 그 중 어떤 것을 찾아도 좋다. 또, 순서쌍들끼리의 호출 순서는 무시된다. 즉, 문제의 예에서는 `answer(0,1,4)`를 호출하고 `answer(2,3,5)`를 호출해도 되고, `answer(2,3,5)`를 호출한 후 `answer(0,1,4)`를 호출해도 된다.

구현 세부사항

여러분은 `onetwothree.cpp`라는 이름의 정확히 하나의 파일을 제출해야 한다. 이 파일에는 다음의 함수가 구현되어 있어야 한다.

- `void maximize( vector<int> A );`

이 함수는 위에서 설명한 것과 같이 동작하여야 한다. 물론 다른 함수들을 만들어서 내부적으로 사용할 수 있다. 제출한 코드는 입출력을 수행하거나 다른 파일에 접근하여서는 안된다.

grader 예시

주어지는 `grader`는 다음과 같은 형식으로 입력을 받는다.

- line 1: N
- line 2: N 개의 정수로 각각 1, 2, 3 중 하나

주어진 `grader`는 첫 줄에 `maximize` 함수가 `answer(i, j, k)` 함수를 호출한 횟수를 출력한 후, 각각의 `answer(i, j, k)` 호출마다 i, j, k 값을 한 줄에 출력한다.

제한 조건

- $3 \leq N \leq 15,000$

서브태스크 1 [14 points]

- $N \leq 18$

서브태스크 2 [29 points]

- $N \leq 100$

서브태스크 3 [53 points]

- $N \leq 3,000$,

서브태스크 4 [54 points]

- 원래 제한 조건 이외의 추가적인 제한 조건이 없음

[입력 예]

```
6
1 2 3 2 3 1
```

위의 입력에서 여러분의 코드가 동작하는 방식에 따른 실행 예를 아래에 보인다.

호출	리턴 값
<code>maximize({1, 2, 3, 2, 3, 1})</code>	2
<code>answer(0, 1, 4)</code>	0 1 4
<code>answer(2, 3, 5)</code>	2 3 5