

문제 1. 저울대

입력 파일: balance.in
출력 파일: balance.out
시간 제한: 2 seconds
메모리 제한: 256 megabytes

새 헛간의 마룻간을 지을 돈을 모으기 위하여, 소 베시는 저울대에서 앞뒤로 움직이면서 균형을 잡는 놀라운 서커스를 하기 시작했다.

베시가 버는 돈의 양은 최종적으로 저울대에서 점프하는 위치와 고나련 되어있다. 저울대의 위치는 왼쪽에서 오른쪽으로 $0, 1, \dots, N+1$ 의 번호가 붙어있다. 만약 베시가 0 혹은 $N+1$ 에 도달하면 저울대에서 떨어져서 돈을 받을 수 없다.

만약 베시가 k 번 위치에 있다면, 다음 중 하나를 할 수 있다.

1. 동전을 던진다. 만약 뒷면이 나오면 $k-1$ 번 위치로 간다. 앞면이 나오면 $k+1$ 번 위치로 간다. (앞뒷면이 나올 확률은 각각 $\frac{1}{2}$ 이다.)
2. 저울대에서 점프를 하여 $f(k)$ 만큼의 돈을 번다. ($0 \leq f(k) \leq 10^9$)

베시의 움직임은 무작위 동전에 영향을 받기 때문에 어떤 결과가 나올지 보장할 수 없다는 것을 알고 있다. 하지만, 시작하는 위치에 따라서 최적의 결정들을 할 때 벌 수 있는 돈의 기댓값을 구하고 싶다. (“최적의”라는 뜻은, 돈의 기댓값을 최대로 하는 것이다.) 예를 들어, 전략이 $1/2$ 의 확률로 10 , $1/4$ 의 확률로 8 , $1/4$ 의 확률로 0 만큼의 돈을 번다면, 벌 수 있는 돈의 기댓값은 가중평균인 $10(1/2) + 8(1/4) + 0(1/4) = 7$ 이다.

입력 형식

첫째 줄에는 정수 N 이 주어진다. 다음 N 개의 줄에는 $f(1) \dots f(N)$ 이 주어진다.

출력 형식

N 개의 줄을 출력하여라. i 번째 줄에는 i 번 위치에서 시작해서, 최적의 전략을 사용하였을 때 벌 수 있는 돈의 기댓값의 10^5 배를 가장 가까운 정수로 반올림하여 출력하여라.

예제

balance.in	balance.out
2	150000
1	300000
3	

문제 2. 정리하자

입력 파일: itout.in
출력 파일: itout.out
시간 제한: 2 seconds
메모리 제한: 512 megabytes

농부 존은 $1 \cdots N$ 의 번호가 붙은 일렬로 서 있는 N 마리의 소를 가지고 있다. ($1 \leq N \leq 10^5$)

농부 존은 소가 증가하는 순서대로 정렬되기를 원하지만, 현재는 정리가 되어있지 않다. 예전에 농부 존은 놀라운 “버블정렬” 알고리즘을 사용하여 소를 정렬했지만, 오늘은 귀찮다고 생각한다. 한번에 하나씩 특정한 소를 불러 정렬 할 것이다. 어떤 소가 불렸을 때, 그 소는 (자신의 관점에서) 정리가 되도록 움직일 것이다. 그 소의 오른쪽에 있는 소의 번호가 더 작아질 때 까지, 오른쪽에 있는 소와 자리를 바꿀 것이다. 그 후, 왼쪽에 있는 소의 번호가 더 커질 때 까지, 왼쪽에 있는 소와 자리를 바꿀 것이다. 최종적으로, 소는 정렬되었다. 그 소의 입장에서 왼쪽에 있는 소가 작은 번호를, 오른쪽에 있는 소가 큰 번호를 가진다.

농부 존은 어떤 소의 부분집합을 골라서, 이 부분집합을 차례대로 보면서 (번호가 증가하는 순서로) 소를 불러 N 마리의 소가 정렬될 때 까지 반복할 것이다. 예를 들어, 번호 {2, 4, 5}번 소를 부분집합으로 고르면, 2번 소를 부르고, 4번 소를 부르고, 5번 소를 부를 것이다. N 마리의 소가 정렬되지 않았다면, 같은 소에게 계속 계속 반복해서 부를 것이다.

농부 존은 어떤 소들이 집중하는지 모르기 때문에, 이 부분집합의 크기를 최소로 하고 싶다. 그리고 농부 존은 K 가 행운의 숫자라고 생각 하고 있다. 소가 최종적으로 모두 정렬되어있게 하는 크기가 최소인 부분집합 중 사전순으로 K 번째 집합을 찾아라.

집합 S 가 T 보다 작기 위해서는, S 와 T 의 원소들을 오름차순으로 정렬 했을 때, 정렬된 두 수열에 대해서, S 를 정렬한 수열이 사전순으로 작다는 의미이다. 예를 들어, {1, 3, 6} 은 {1, 4, 5} 보다 사전순으로 작다.

배점

3/16의 점수에 해당하는 데이터에서는 $N \leq 6$ 이고 $K = 1$ 이다.

추가되는 5/16의 점수에 해당하는 데이터에서는 $K = 1$ 이다.

추가되는 8/16의 점수에 해당하는 데이터에서는 추가 제한조건이 없다.

입력 형식

첫째 줄에는 정수 N 이 주어진다. 둘째 줄에는 정수 K ($1 \leq K \leq 10^{18}$) 이 주어진다. 셋째 줄에는 N 개의 공백으로 구분된 정수가 주어지고, 왼쪽부터 오른쪽 까지 소의 번호가 주어진다.

적어도 K 개의 올바른 부분집합이 존재함이 보장된다.

출력 형식

첫째 줄에는 최소 크기 부분집합의 크기를 출력해야 한다. 다음 부터 사전순으로 K 번째 최소 집합을 번호가 증가하는 순서대로 한 줄에 하나씩 출력해야 한다.

제한

itout.in	itout.out
4 1	2
4 2 1 3	1
	4

참고 사항

배열 4 2 1 3으로 시작한다. 농부 존이 1번 소를 부른 이후로 배열은 1 4 2 3이 된다. 4번 소를 부른 이후로 배열은 1 2 3 4 가 되고, 정렬되었다.

문제 3. 소 모임

입력 파일: gathering.in
출력 파일: gathering.out
시간 제한: 2 seconds
메모리 제한: 256 megabytes

소들은 거대한 모임을 위해서 세계 곳곳에서 모였다. N 마리의 소가 있고, $N - 1$ 쌍의 소가 서로를 안다. 모든 소는 몇몇 서로 아는 관계를 거쳐서 모두 알 수 있다.

매우 재밌는 시간을 보냈지만, 이제 하나하나씩 떠날 시간이 왔다. 그들은 떠날 때 소가 두 마리 이상 있을 때는, 남아있는 모든 소에게 적어도 한 마리의 친구가 있도록 떠나고 싶다. 또한, 수하물 공간 문제 때문에, M 쌍의 소 (a_i, b_i) 가 있어서, a_i 번 소가 b_i 번 소보다 먼저 떠나야 한다. a_i 번 소와 b_i 번 소는 친구일 수도, 아닐 수도 있다.

각 소에 대해서, 해당하는 소가 떠나는 마지막 소가 될 수 있는지를 구하여라. 위 조건을 만족하면서 떠날 수 있는 방법이 존재하지 않을 수도 있다.

입력 형식

첫째 줄에는 공백으로 구분된 두 정수 N, M 이 주어진다.

둘째 줄부터 N 번째 줄까지의 i 번째 줄에는 x_i 와 y_i 가 주어지고, $1 \leq x_i, y_i \leq N$, $x_i \neq y_i$ 이며, x_i 번 소와 y_i 번 소가 친구라는 것을 의미한다.

$N + 1$ 번째 줄부터 $N + M$ 번째 줄까지의 i 번째 줄에는 a_i 와 b_i 가 주어지고, $1 \leq a_i, b_i \leq N$, $a_i \neq b_i$ 이며, a_i 번 소가 b_i 번 소가 떠나기 전에 떠나야 한다는 것을 의미한다.

$1 \leq N, M \leq 10^5$ 임이 보장된다. 20% 점수에 해당하는 테스트 케이스에 대해서는 $N, M \leq 3000$ 임이 보장된다.

출력 형식

N 개의 각 줄에 d_i 를 출력하여라. i 번째 소가 마지막으로 떠날 수 있으면 $d_i = 1$ 이고, 아니면 $d_i = 0$ 이다.

예제

gathering.in		gathering.out	
5	1	0	
1	2	0	
2	3	1	
3	4	1	
4	5	1	
2	4		