

문제	8-A
제목	온라인 정렬
내용	<p>[저작권 관련 사항: 이 문제는 Yoshihisa Itsubaki에 의해 설계된 Streams 보드 게임을 응용한 문제임. 참고 URL: <a href="http://happybaobab.com/?mod=game_detail&amp;uid=417">http://happybaobab.com/?mod=game_detail&amp;uid=417</a> ]</p> <p>숫자들을 저장할 수 있는 <math>M</math>개의 슬롯(1차원 배열)이 주어진다고 가정하자. 우리는 여기에 1부터 <math>N(\geq M)</math> 사이의 자연수 <math>M</math>개를 채워 넣고자 한다. 자연수들은 정해진 순서에 따라 주어지지만 우리는 이 순서를 미리 알지는 못하며, 같은 숫자가 주어지지 않는다고 한다. 가능하면 숫자들이 작은 숫자에서 큰 숫자의 순서로, 즉 오름차순으로 정렬되도록 슬롯을 채워 넣기를 희망하지만, 숫자들이 주어지는 순서를 미리 알지 못하므로 각 숫자가 주어질 때 적당한 슬롯에 일단 넣는 식으로 진행할 수밖에 없다고 한다. 단, 한 번 숫자를 채워 넣고 나면 다른 슬롯으로 이동은 불가능하며, <math>M</math>개의 숫자를 모두 채워 넣고 나면 오름차순으로 연결된 구간의 개수 및 각 구간의 길이에 따라 점수가 주어진다. 예를 들어, <math>M=5</math>인 경우 오름차순 구간의 길이가 1, 2, 3, 4, 5일 때 각각 1, 4, 8, 16, 32점이 주어진다고 하자. 만약 <math>N=30</math>이고, 숫자들이 1, 30, 28, 29, 15의 순서로 주어지며, 각 숫자가 주어질 때 각각 1번, 5번, 4번, 3번, 2번 슬롯에 넣었다면, 이 동작들은 각 숫자와 그 숫자가 들어간 슬롯의 번호를 순서쌍으로 구성하여 (1, 1), (30, 5), (28, 4), (29, 3), (15, 2)와 같이 나타낼 수 있다. 이 동작들을 수행하고 나면 슬롯들의 최종 상태는 (1, 15, 29, 28, 30)이 되는데, 오름차순인 구간은 (1, 15, 29)와 (28, 30)의 둘이므로, 정렬 점수는 <math>8+4 = 12</math>점이 된다. 숫자와 해당 숫자를 넣은 슬롯의 번호들로 구성된 순서쌍들이 주어질 때 정렬 점수를 계산하는 프로그램을 작성하시오.</p> <p>사용할 수 있는 언어는 C, C++로 제한한다. 프로그램의 실행 시간은 1초이다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin 입력 속도를 개선할 수 있다.</p> <pre>std::ios::sync_with_stdio(false);</pre>
입력 형식	<p>입력은 standard in으로 다음과 같이 주어진다. 첫 줄에는 테스트케이스의 수 <math>T(\leq 100)</math>가 주어지며, 둘째 줄부터는 테스트케이스들이 주어진다. 각 테스트케이스는 다음과 같이 구성된다.</p> <ol style="list-style-type: none"> <li>1. 각 테스트케이스의 첫 번째 줄에는 이 테스트케이스를 위한 슬롯의 개수 <math>M(\leq 100)</math> 및 자연수의 범위 <math>N(M \leq N \leq 1,000,000)</math>을 나타내는 자연수가 빈 칸 하나를 사이에 두고 주어진다.</li> <li>2. 각 테스트케이스의 두 번째 줄에는 오름차순 구간의 길이별 점수들 (<math>\leq 10,000</math>)이 빈 칸 하나를 사이에 두고 주어진다.</li> <li>3. 세 번째 줄부터 <math>M</math>개의 줄에는 자연수(<math>\leq N</math>)와 그 자연수를 넣을 슬롯의</li> </ol>

		번호( $\leq M$ )가 빈 칸 하나를 사이에 두고 주어진다.
출력 형식		출력은 standard out으로 표시하며, 각 테스트케이스 별로 점수들을 출력한다. (출력은 총 $T$ 줄이 된다.)
예	입력	2 5 30 // 테스트케이스 1: 30 이하의 숫자 5개가 주어질 예정임 1 4 8 16 32 // 길이 1인 구간은 1점, 길이 2인 구간은 4점, ... 1 1 30 5 28 4 29 3 15 2 6 100 // 테스트케이스 2: 100 이하의 숫자 6개가 주어질 예정임 1 2 3 100 200 300 95 6 8 1 98 5 77 4 31 3 2 2 (empty line)
	출력	12 102 (empty line)