R-3 (실습)

|  |
| --- |
| <설명>  초기 역량값 X와 배열 [A, B]의 집합인 TaskSet이 주어질 때, A는 작업을 시작할 수 있는 최소 역량값, B는 작업 후 얻게 되는 역량 증가값이다. 즉, X ≥ A 이면 작업이 자동 시작되고, 그로 인해 X는 X+B로 바뀌게 된다.  이 때, X=0부터 시작한다고 가정했을 때 주어진 TaskSet으로 최대 얼마까지 역량이 상승될 수 있는 지를 구하시오. 예를 들어, 배열 [11, 4], [0, 3], [3, 5]가 주어졌을 때, 시작 값 0부터 값을 최대 8까지 올릴 수 있다 (0+3+5).  <제한 사항>  가능한 작업이 존재하지 않을 때 -1를 출력.  TaskSet의 각 배열 원소 B의 값은 1 이상. |
| <입력>  - 첫째 줄에 TaskSet의 길이가 주어진다.  - 둘째 줄에 TaskSet의 원소를 공백으로 구분하여 넣습니다. |
| <출력>  - 첫째 줄에 상승된 X의 최대값이 출력됩니다. |
| <예제>   |  |  | | --- | --- | | 입력 1  4  0 1 5 4 1 2 10 2 | 출력 1  3 | | 입력 2  4  2 3 2 2 0 2 6 4 | 출력 2  11 | | 입력 3  3  14 4 4 4 1 2 | 출력 3  -1 | |

R-4 (실습)

|  |
| --- |
| <설명>  초기 평소 식욕이 왕성한 무지는 자신의 재능을 뽐내고 싶어 졌고 고민 끝에 카카오 TV 라이브로 방송을 하기로 마음먹었다. 그냥 먹방을 하면 다른 방송과 차별성이 없기 때문에 무지는 아래와 같이 독특한 방식을 생각해냈다.  회전판에 먹어야 할 N 개의 음식이 있다. 각 음식에는 1부터 N 까지 번호가 붙어있으며, 각 음식을 섭취하는데 일정 시간이 소요된다. 무지는 다음과 같은 방법으로 음식을 섭취한다.  1. 무지는 1번 음식부터 먹기 시작하며, 회전판은 번호가 증가하는 순서대로 음식을 무지 앞으로 가져다 놓는다.  2. 마지막 번호의 음식을 섭취한 후에는 회전판에 의해 다시 1번 음식이 무지 앞으로 온다.  3. 무지는 음식 하나를 1초 동안 섭취한 후 남은 음식은 그대로 두고, 다음 음식을 섭취한다.  다음 음식이란, 아직 남은 음식 중 다음으로 섭취해야 할 가장 가까운 번호의 음식을 말한다.  회전판이 다음 음식을 무지 앞으로 가져오는데 걸리는 시간은 없다고 가정한다.  4. 무지가 먹방을 시작한 지 K 초 후에 네트워크 장애로 인해 방송이 잠시 중단되었다.  무지는 네트워크 정상화 후 다시 방송을 이어갈 때, 몇 번 음식부터 섭취해야 하는지를 알고자 한다. 각 음식을 모두 먹는데 필요한 시간이 담겨있는 배열 food\_times, 네트워크 장애가 발생한 시간 K 초가 매개변수로 주어질 때 몇 번 음식부터 다시 섭취하면 되는지 return 하도록 solution 함수를 완성하라.  입출력 예 설명  입출력 예 #1  0~1초 동안에 1번 음식을 섭취한다. 남은 시간은 [2,1,2] 이다.  1~2초 동안 2번 음식을 섭취한다. 남은 시간은 [2,0,2] 이다.  2~3초 동안 3번 음식을 섭취한다. 남은 시간은 [2,0,1] 이다.  3~4초 동안 1번 음식을 섭취한다. 남은 시간은 [1,0,1] 이다.  4~5초 동안 (2번 음식은 다 먹었으므로) 3번 음식을 섭취한다. 남은 시간은 [1,0,0] 이다.  5초에서 네트워크 장애가 발생했다. 1번 음식을 섭취해야 할 때 중단되었으므로, 장애 복구 후에 1번 음식부터 다시 먹기 시작하면 된다.  제한사항  -food\_times 는 각 음식을 모두 먹는데 필요한 시간이 음식의 번호 순서대로 들어있는 배열이다.  -k 는 방송이 중단된 시간을 나타낸다.  -만약 더 섭취해야 할 음식이 없다면 -1을 반환하면 된다.  정확성 테스트 제한 사항  -food\_times 의 길이는 1 이상 2,000 이하이다.  -food\_times 의 원소는 1 이상 1,000 이하의 자연수이다.  -k는 1 이상 2,000,000 이하의 자연수이다.  효율성 테스트 제한 사항  -food\_times 의 길이는 1 이상 200,000 이하이다.  -food\_times 의 원소는 1 이상 100,000,000 이하의 자연수이다.  -k는 1 이상 2 x 10^13 이하의 자연수이다. |
| <입력>  - 첫째 줄에 k값을 입력한다.  - 둘째 줄에 food\_times을 입력합니다. |
| <출력>  - 첫째 줄에 섭취해야하는 음식의 번호를 입력하세요 |
| <예제>   |  |  | | --- | --- | | 입력 1  5  3 12 | 출력 1  1 | |

T-1 (과제)

|  |  |  |
| --- | --- | --- |
| <설명>  동빈이네 전자 매장에는 부품이 N개 있다. 각 부품은 정수 형태의 고유한 번호가 있다. 어느 날 손님이 M개의 종류의 부품을 대량으로 구매하겠다며 당일 날 견적서를 요청했다. 동빈이는 때를 놓치지 않고 손님이 문의한 부품 M개 종류를 모두 확인해서 견적서를 작성해야 한다. 이때 가게 안에 부품이 모두 있는지 확인하는 프로그램을 작성해보자.  예를 들어 가게의 부품이 총 5개일 때 부품 번호가 다음과 같다고 하자.   |  | | --- | | N=5  [8, 3, 7, 9, 2] |   손님은 총 3개의 부품이 있는지 확인 요청했는데 부품 번호는 다음과 같다.   |  | | --- | | M=3  [5, 7, 9] |   이때 손님이 요청한 부품 번호의 순서대로 부품을 확인해 부품이 있으면 yes를, 없으면 no를 출력한다. 구분은 공백으로 한다. |
| <입력>  - 첫째 줄에 정수 N이 주어진다. (1<=N<=1,000,000)  - 둘째 줄에는 공백으로 구분하여 N개의 정수가 주어진다. 이때 정수는 1보다 크고 1,000,000 이하이다.  - 셋째 줄에는 정수 M이 주어진다. (1<=M<=100,000)  - 넷째 줄에는 공백으로 구분하여 M개의 정수가 주어진다. 이때 정수는 1보다 크고 10억 이하이다. |
| <출력>  - 첫째 줄에 공백으로 구분하여 각 부품이 존재하면 yes를, 없으면 no를 출력한다. |
| <예제>   |  |  | | --- | --- | | 입력 1  5  8 3 7 9 2  3  5 7 9 | 출력 1  no yes yes | |

T-2 (과제)

|  |
| --- |
| <설명>  N개의 원소를 포함하고 있는 수열이 오름차순으로 정렬되어 있습니다. 이때 이 수열에서 x가 등장하는 횟수를 계산하세요. 예를 들어 수열 {1, 1, 2, 2, 2, 2, 3}이 있을 때 x = 2라면, 현재 수열에서 값이 2인 원소가 4개이므로 4를 출력합니다.  단, 이 문제는 시간 복잡도 O(log N)으로 알고리즘을 설계하지 않으면 '시간 초과' 판정을 받습니다. |
| <입력>  - 줄에 N과 x가 정수 형태로 공백으로 구분되어 입력됩니다. (1 ≤ N ≤ 1,000,000), (-10⁹ ≤ x ≤ 10⁹)  - 둘째 줄에 N개의 원소가 정수 형태로 공백으로 구분되어 입력됩니다. (-10⁹ ≤ 각 원소의 값 ≤ 10⁹) |
| <출력>  - 수열의 원소 중에서 값이 x인 원소의 개수를 출력합니다. 단, 값이 x인 원소가 하나도 없다면  -1을 출력합니다. |
| <예제>   |  |  | | --- | --- | | 입력 1  7 3  1 1 2 2 2 2 3 | 출력 1  4 | | 예제 입력 2  7 4  1 1 2 2 2 2 3 | 출력 2  -1 | |

T-3 (과제)

|  |
| --- |
| <설명>  오늘 동빈이는 여행 가신 부모님을 대신해서 떡집 일을 하기로 했다. 오늘은 떡볶이 떡을 만드는 날이다. 동빈이네 떡볶이 떡은 재밌게도 떡볶이 떡의 길이가 일정하지 않다. 대신에 한 봉지 안에 들어가는 떡의 총 길이는 절단기로 잘라서 맞춰준다.  절단기의 높이(H)를 지정하면 줄지어진 떡을 한 번에 절단한다. 높이가 H보다 긴 떡은 H 위의 부분이 잘릴 것이고, 낮은 떡은 잘리지 않는다. 이걸 처리 안 해줘서 시간을 허비했다.  예를 들어 높이가 19, 14, 10, 17cm인 떡이 나란히 있고 절단기 높이를 15cm로 지정하면 자른 뒤 떡의 높이는 15, 14, 10, 15cm가 될 것이다. 잘린 떡의 길이는 차례대로 4, 0, 0, 2cm이다. 손님은 6cm만큼의 길이를 가져간다.  손님이 왔을 때 요청한 총 길이가 M일 때 적어도 M만큼의 떡을 얻기 위해 절단기에 설정할 수 있는 높이의 최댓값을 구하는 프로그램을 작성하시오. |
| <입력>  - 첫째 줄에 떡의 개수 N과 요청한 떡의 길이 M이 주어진다. (1<=N<=1,000,000, 1<=M<=2,000,000,000)  - 둘째 줄에는 떡의 개별 높이가 주어진다. 떡 높이의 총합은 항상 M 이상이므로, 손님은 필요한 양만큼 떡을 사갈 수 있다. 높이는 10억보다 작거나 같은 양의 정수 또는 0이다. |
| <출력>  - 적어도 M만큼의 떡을 집에 가져가기 위해 절단기에 설정할 수 있는 높이의 최댓값을 출력한다. |
| <예제>   |  |  | | --- | --- | | 입력 1  4 6  19 15 10 17 | 출력 1  15 | |