

1번

유형	선행 탐색	소요 시간	10분								
문제	<p>[문제 설명] 정수 n이 주어질 때, n보다 작거나 같은 수들 중 가장 큰 세제곱수를 구하는 프로그램을 작성하세요.</p> <p>[제한 사항] - $1 \leq n \leq 100,000,000$</p> <p>[입력 형식] 입력의 첫 번째 줄에는 정수 n이 입력됩니다.</p> <p>[출력 형식] n 이하의 자연수 중 가장 큰 세제곱수를 출력합니다.</p> <p>[입출력 예시 #1]</p> <table border="1"><thead><tr><th>n</th><th>출력</th></tr></thead><tbody><tr><td>15</td><td>8</td></tr></tbody></table> <p>[입출력 예시 #2]</p> <table border="1"><thead><tr><th>n</th><th>출력</th></tr></thead><tbody><tr><td>125</td><td>125</td></tr></tbody></table>	n	출력	15	8	n	출력	125	125		
n	출력										
15	8										
n	출력										
125	125										

2번

유형	정렬	소요 시간	10분				
문제	<p>[문제 설명] 0~9의 수가 최대 100,000개 개 입력됩니다. 이때 가장 많이 들어온 숫자 순서대로 출력하세요. 단, 들어온 횟수가 동일한 경우에는 작은 수를 먼저 출력합니다.</p> <p>[제한 사항] 입력된 문자열의 길이 $\leq 100,000$</p> <p>[입력 형식] 0~9로 이루어진 길이 100,000 이하의 문자열이 들어옵니다.</p> <p>[출력 형식] 가장 많이 들어온 숫자 순서대로 공백으로 구분하여 출력하세요.</p> <p>[입출력 예시 #1]</p> <table border="1"><thead><tr><th>str</th><th>출력</th></tr></thead><tbody><tr><td>221123</td><td>1 2 3 0 4 5 6 7 8 9</td></tr></tbody></table>	str	출력	221123	1 2 3 0 4 5 6 7 8 9		
str	출력						
221123	1 2 3 0 4 5 6 7 8 9						

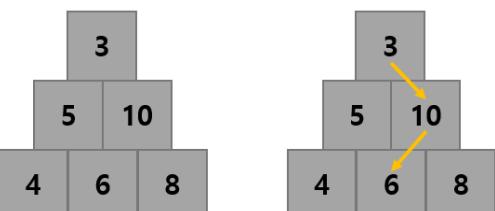
	[입출력 예시 #2]	
	str	출력
	1235670089006427894100	0 1 2 4 6 7 8 9 3 5
초기코드	<pre>/* * @param str {string} * * @return {numbers} */</pre>	

3번

유형	Dynamic Programming	예상 소요 시간	40분									
문제	<p>[문제 설명] $h \times w$개의 격자 형태의 게임판이 있습니다. 이 중 원쪽 위에 시작점이 있고, 오른쪽 아래에 종료 지점이 있습니다. 게임 방법은 시작 지점에 말을 놓고, 말을 종료 지점까지 이동하면 이기는 게임이며, 말은 오른쪽이나 아래로만 이동할 수 있습니다. 이때, 게임을 이기는 방법의 수를 출력하는 함수, solution을 완성해주세요.</p> <p>예를 들어, h가 2이고, w가 3인 게임판이 주어질 때, 게임을 이기는 방법의 수는 다음과 같이 총 3가지 방법이 있습니다.</p> <ul style="list-style-type: none"> > 오른쪽 -> 오른쪽 -> 아래 > 오른쪽 -> 아래 -> 오른쪽 > 아래 -> 오른쪽 -> 오른쪽 <p>[제한 사항] - 말은 오른쪽이나 아래로만 이동할 수 있습니다.</p> <p>[입력 형식] - 두 자연수 h와 w는 1 이상 25 이하의 자연수입니다.</p> <p>[출력 형식] 게임을 이기는 방법의 수를 출력합니다.</p> <p>[입출력 예시]</p> <table border="1"> <thead> <tr> <th colspan="2">입력</th> <th>출력</th> </tr> <tr> <th>h</th> <th>w</th> <th></th> </tr> </thead> <tbody> <tr> <td>2</td> <td>3</td> <td>3</td> </tr> </tbody> </table>	입력		출력	h	w		2	3	3		
입력		출력										
h	w											
2	3	3										

매개변수 & 리턴타입	<table border="1"><thead><tr><th colspan="2">매개변수</th><th>Return</th></tr></thead><tbody><tr><td>number</td><td>h</td><td></td></tr><tr><td>number</td><td>w</td><td>number</td></tr></tbody></table>	매개변수		Return	number	h		number	w	number
매개변수		Return								
number	h									
number	w	number								
초기 코드	<pre>/** * @param h {number} * @param w {number} * @return {number} */ function solution(h, w) { var answer = 0; return answer; }</pre>									

4번

유형	Dynamic Programming	예상 소요 시간	40분
문제	<p>[문제 설명]</p> <p>n 층의 피라미드의 꼭대기에서 1층으로 내려와 탈출하려고 합니다. 피라미드에는 방이 있으며, n 층에는 1개의 방이 존재하고, $n-1$ 층에는 2개의 방, $n-2$ 층에는 3개의 방, 1층에는 n개의 방이 존재합니다.</p> <p>각 방은 왼쪽 아래와 오른쪽 아래의 방으로 이동하는 계단이 있고, 각 방에는 j개의 보석이 존재합니다.</p> <p>n 층의 피라미드 각 방에 존재하는 보석의 정보 <code>arr</code>가 주어질 때, 꼭대기 층에서 1층까지 내려오면서 모을 수 있는 보석의 최대 수를 구하는 함수, <code>solution</code>을 완성해주세요.</p> <p>예를 들어, 3층의 피라미드 그림 (a) 인 각 방에 존재하는 보석의 정보 <code>arr</code>가 <code>[[3], [5, 10], [4, 8, 6]]</code>이 주어질 때, 피라미드를 탈출하여 모을 수 있는 보석의 최대 수는 그림 (b) 인 다음과 같습니다. $> 3 + 10 + 8 = 21$</p>  <p>(a)</p> <p>(b)</p> <p>[제한 사항]</p> <ul style="list-style-type: none"> - 계단은 올라갈 수 없고, 내려갈 수만 있습니다. <p>[입력 형식]</p> <ul style="list-style-type: none"> - 피라미드의 층수 n은 1 이상 100 이하의 자연수입니다. 		

	<p>- 각 방의 보석 수 j는 1 이상 1000 이하의 자연수입니다.</p> <p>[출력 형식] 피라미드를 탈출하며 모을 수 있는 보석의 최대 수를 출력합니다.</p> <p>[입출력 예시]</p> <table border="1"> <thead> <tr> <th style="background-color: #d9e1f2;">입력</th><th style="background-color: #d9e1f2;">출력</th></tr> </thead> <tbody> <tr> <td>arr [[3], [5, 10], [4, 8, 6]]</td><td>21</td></tr> </tbody> </table>	입력	출력	arr [[3], [5, 10], [4, 8, 6]]	21
입력	출력				
arr [[3], [5, 10], [4, 8, 6]]	21				
매개변수 & 리턴타입	<table border="1"> <thead> <tr> <th style="background-color: #d9e1f2;">매개변수</th><th style="background-color: #d9e1f2;">Return</th></tr> </thead> <tbody> <tr> <td>number[][], arr</td><td>number</td></tr> </tbody> </table>	매개변수	Return	number[][], arr	number
매개변수	Return				
number[][], arr	number				
초기 코드	<pre>/** * @param arr {number[][]} * @return {number} */ function solution (arr) { var answer = 0; return answer; }</pre>				