

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
<b>Program Name:</b> B. Tech		<b>Assignment Type:</b> Lab	
<b>Course Coordinator Name</b>		Dr. S Vairachilai	
<b>Instructor(s) Name</b>			
<b>Course Code</b>	23CS302PC305	<b>Course Title</b>	Competitive Programming
<b>Year/Sem</b>	III/II	<b>Regulation</b>	R23
<b>Date and Day of Assignment</b>	Week I	<b>Time(s)</b>	NA
<b>Duration</b>	2 Hours each	<b>Applicable to Batches</b>	ALL batches
<b>Assignment Number:</b> Week I (Greedy + Divide and Conquer)			

Day	Question	Expected time to complete
Monday	<p><b>Assignment 1: Maximum Non-Overlapping Meetings</b></p> <p><b>Problem Statement:</b> You are given N meetings. Each meeting has a start time <math>S_i</math> and an end time <math>E_i</math>. You want to attend the maximum number of meetings. You can attend meeting j after meeting i only if the start time of meeting j is strictly greater than the end time of meeting i (<math>S_j &gt; E_i</math>).</p> <p>For each test case, output the maximum number of meetings that can be attended.</p> <p><b>Input Format</b></p> <ul style="list-style-type: none"> <li>The first line contains an integer T, the number of test cases.</li> <li>For each test case: <ul style="list-style-type: none"> <li>The first line contains an integer N.</li> <li>The next N lines each contain two integers <math>S_i</math> and <math>E_i</math>.</li> </ul> </li> </ul> <p><b>Output Format</b> For each test case, print a single integer: the maximum number of meetings that can be attended.</p> <p><b>Constraints</b></p> <ul style="list-style-type: none"> <li><math>1 \leq T \leq 20</math></li> <li><math>1 \leq N \leq 200000</math> (sum of N over all test cases <math>\leq 200000</math>)</li> <li><math>0 \leq S_i &lt; E_i \leq 10^9</math></li> </ul> <p><b>Sample Input</b></p> <pre>1 3 1 3 2 4 3 5</pre> <p><b>Expected Output</b></p> <pre>2</pre>	5 <sup>th</sup> Jan, 2026, 5:00PM

<b>Tuesday</b>	<p><b>Assignment 2: Fractional Knapsack</b></p> <p><b>Problem Statement</b></p> <p>You are given N items. Item i has value <math>V_i</math> and weight <math>W_i</math>. You have a knapsack of capacity W. You may take any fraction of an item (including the whole item). Your goal is to maximize the total value in the knapsack without exceeding the capacity.</p> <p>For each test case, output the maximum total value achievable. The result must be printed with exactly 6 digits after the decimal point.</p> <p><b>Input Format</b></p> <ul style="list-style-type: none"> <li>The first line contains an integer T, the number of test cases. For each test case:</li> <li>The first line contains two integers N and W.</li> <li>The next N lines each contain two integers <math>V_i</math> and <math>W_i</math>.</li> </ul> <p><b>Output Format</b></p> <p>For each test case, print one number: the maximum value, formatted to 6 decimal places.</p> <p><b>Constraints</b></p> <ul style="list-style-type: none"> <li><math>1 \leq T \leq 20</math></li> <li><math>1 \leq N \leq 200000</math> (sum of N over all test cases <math>\leq 200000</math>)</li> <li><math>1 \leq V_i, W_i, W \leq 10^9</math></li> </ul> <p><b>Sample Input</b></p> <pre>1 3 50 60 10 100 20 120 30</pre> <p><b>Expected Output</b></p> <pre>240.000000</pre>	6 <sup>th</sup> Jan, 2025 5:00PM
<b>Wednesday</b>	<p><b>Assignment 3: Job Sequencing with Deadlines</b></p> <p><b>Problem Statement</b></p> <p>You are given N jobs. Each job takes exactly 1 unit of time. Job i has a deadline <math>D_i</math> and a profit <math>P_i</math>. If a job is completed on or before its deadline, its profit is earned; otherwise, it cannot be counted.</p> <p>You can perform at most one job at a time. Your task is to choose and schedule jobs to maximize total profit.</p> <p>For each test case, output:</p> <ol style="list-style-type: none"> <li>the number of jobs completed</li> <li>the maximum total profit</li> </ol> <p><b>Input Format</b></p> <ul style="list-style-type: none"> <li>The first line contains an integer T, the number of test cases. For each test case:</li> <li>The first line contains an integer N.</li> <li>The next N lines each contain two integers <math>D_i</math> and <math>P_i</math>.</li> </ul>	7 <sup>th</sup> Jan, 2025 5:00PM

	<p><b>Output Format</b> For each test case, print two integers: jobs_done total_profit</p> <p><b>Constraints</b></p> <ul style="list-style-type: none"> <li>• <math>1 \leq T \leq 20</math></li> <li>• <math>1 \leq N \leq 200000</math> (sum of N over all test cases <math>\leq 200000</math>)</li> <li>• <math>1 \leq D_i \leq 100000</math></li> <li>• <math>1 \leq P_i \leq 10^9</math></li> </ul> <p><b>Sample Input</b></p> <pre>1 5 2 100 1 19 2 27 1 25 3 15</pre> <p><b>Expected Output</b></p> <pre>3 142</pre>	
	<p><b>Assignment 4: Count Inversions (Divide and Conquer)</b></p> <p><b>Problem Statement</b> Given an array A of size N, an inversion is a pair of indices <math>(i, j)</math> such that <math>i &lt; j</math> and <math>A[i] &gt; A[j]</math>. Your task is to count the total number of inversions in the array.</p> <p>This must be solved efficiently for large N (a naive <math>O(N^2)</math> approach will not pass).</p> <p><b>Input Format</b></p> <ul style="list-style-type: none"> <li>• The first line contains an integer T, the number of test cases. For each test case:</li> <li>• The first line contains an integer N.</li> <li>• The second line contains N integers representing the array.</li> </ul>	
Thursday	<p><b>Output Format</b> For each test case, print a single integer: the inversion count.</p> <p><b>Constraints</b></p> <ul style="list-style-type: none"> <li>• <math>1 \leq T \leq 20</math></li> <li>• <math>1 \leq N \leq 200000</math> (sum of N over all test cases <math>\leq 200000</math>)</li> <li>• <math>-10^9 \leq A[i] \leq 10^9</math></li> </ul> <p><b>Sample Input</b></p> <pre>2 5 2 4 1 3 5 4 4 3 2 1</pre> <p><b>Expected Output</b></p> <pre>3</pre>	8 <sup>th</sup> Jan, 2025 5:00PM

	6	
	<b>Assignment 5: Kth Smallest Element (Quickselect Divide and Conquer)</b> <p><b>Problem Statement</b>            You are given an array of N integers and an integer K (1-indexed). You must find the Kth smallest element in the array.</p> <p><b>Example:</b>            Array: 7 10 4 3 20 15            Sorted: 3 4 7 10 15 20            K = 3, answer = 7</p> <p>For each test case, output the Kth smallest element.</p> <p><b>Input Format</b></p> <ul style="list-style-type: none"> <li>The first line contains an integer T, the number of test cases.            For each test case:             <ul style="list-style-type: none"> <li>The first line contains two integers N and K.</li> <li>The second line contains N integers.</li> </ul> </li> </ul> <p><b>Output Format</b>            For each test case, print a single integer: the Kth smallest element.</p> <p><b>Constraints</b></p> <ul style="list-style-type: none"> <li><math>1 \leq T \leq 20</math></li> <li><math>1 \leq N \leq 200000</math> (sum of N over all test cases <math>\leq 200000</math>)</li> <li><math>1 \leq K \leq N</math></li> <li><math>-10^9 \leq A[i] \leq 10^9</math></li> </ul> <p><b>Sample Input</b>            2            6 3            7 10 4 3 20 15            7 4            12 3 5 7 19 0 2</p> <p><b>Expected Output</b>            7            5</p>	
Friday		9 <sup>th</sup> Jan, 2025 5:00PM