|  |
| --- |
| **Problem Statement:**  You have to implement the concept of **Hashmap** in Java. The final task is to merge two hashmaps in such a way that no data is lost. If the maps contain duplicate keys with different values, then while merging add the value of the duplicate key with a new key by concatenating “new” with the key.  You are given a class *Tester* having mergeMaps and main method. You’ve been provided with a starter code which contains sub-tasks as defined below:  **Sub task 1:** Create two hashmaps inside main method to store names and values  **Sub task 2:** Store some names as key and different values in both hashmaps  **Sub task 3:** Call mergeMaps method by passing both hashmaps created in main method.  **Sub task 4:** Implement a logic to copy both maps into one and override if both key and values are same otherwise only if keys are same and value is different append key with new keyword.  **Sub task 5:** Return the resultant map and display  **Note:**  1. Do not remove the predefined code else your code may not execute as expected.  2. You’ve to solve the problem using **Hashmap** ONLY. Solving through any alternate method other than the Hashmap may lead to disqualification. |
| **Input Format:**  First line contains N inputs for map one  Pass key and value for map for N lines  After passing all key values for N lines input M for map two  Pass key and value for map for M lines  **Output Format:**  Print the merge output of two maps |
| **Input Format :**  3  Kelly  10  Micheal  20  Ryan  30  2  Jim  15  Andy  45  **Output Format :**  {“Kelly”=10,”Micheal”=20,”Ryan”=30,”Jim”=15,”Andy”=45} |
| **Test Case 1 :**  **Input Format:**  3  Toby  10  Micheal  20  Angela  30  Toby  10  Andy  45  **Output Format:**  {“Toby”=10,”Micheal”=20,”Angela”=30, “Andy”=45}  **Test Case 2 :**  **Input Format:**  3  Kelly  10  Micheal  20  Ryan  30  Kelly  15  Andy  45  **Output Format:**  {“Kelly”=10,”Micheal”=20,”Ryan”=30,”KellyNew”=15,”Andy”=45} |