**Screenshot of the results:**

**Fuzzy Sort:**

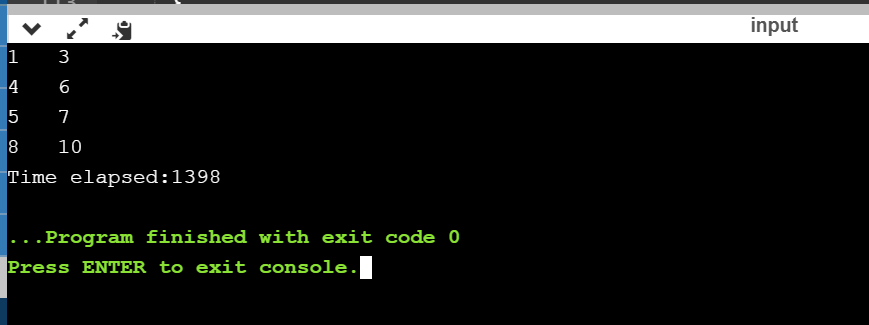
**Test case 1:**

5, 7

1, 3

4, 6

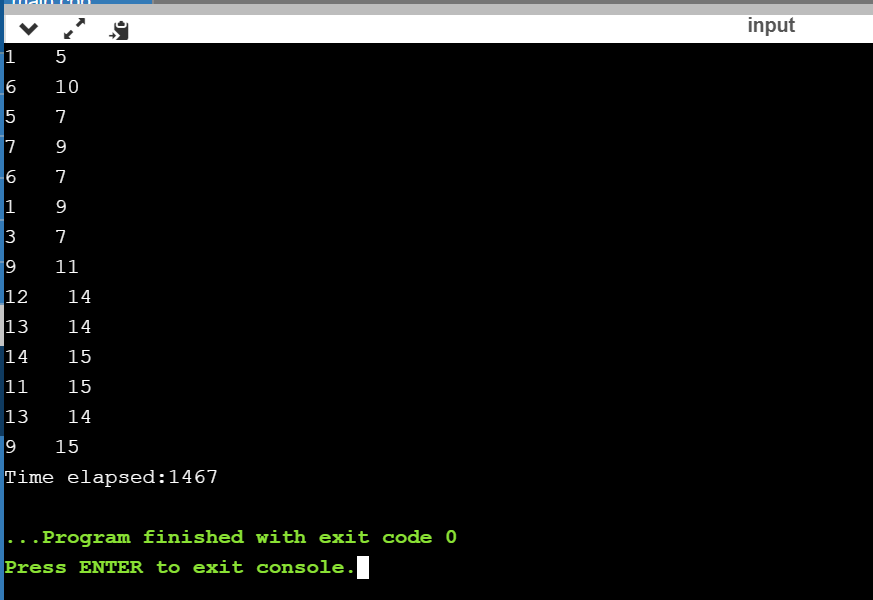
8, 10



**Test case 2:**

6, 7  
9, 11  
13, 14  
3, 7  
11, 15  
13, 14  
12, 14  
14, 15  
9, 15  
5, 7  
7, 9  
1, 5  
1, 9  
6, 10

Output:



**Randomized Quicksort:**

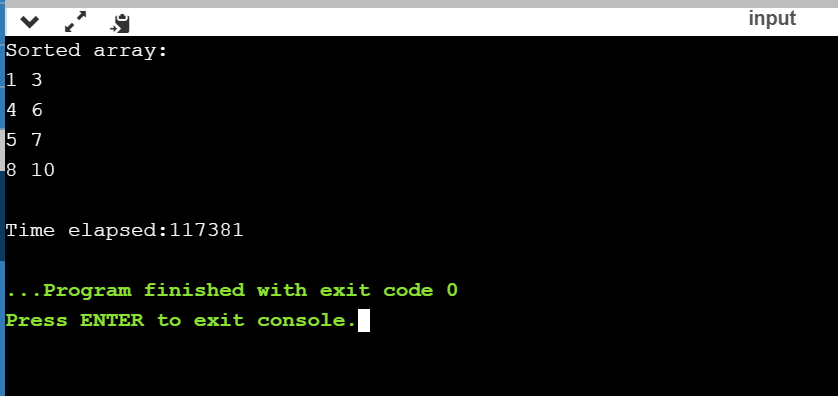
**Test case 1:**

5, 7

1, 3

4, 6

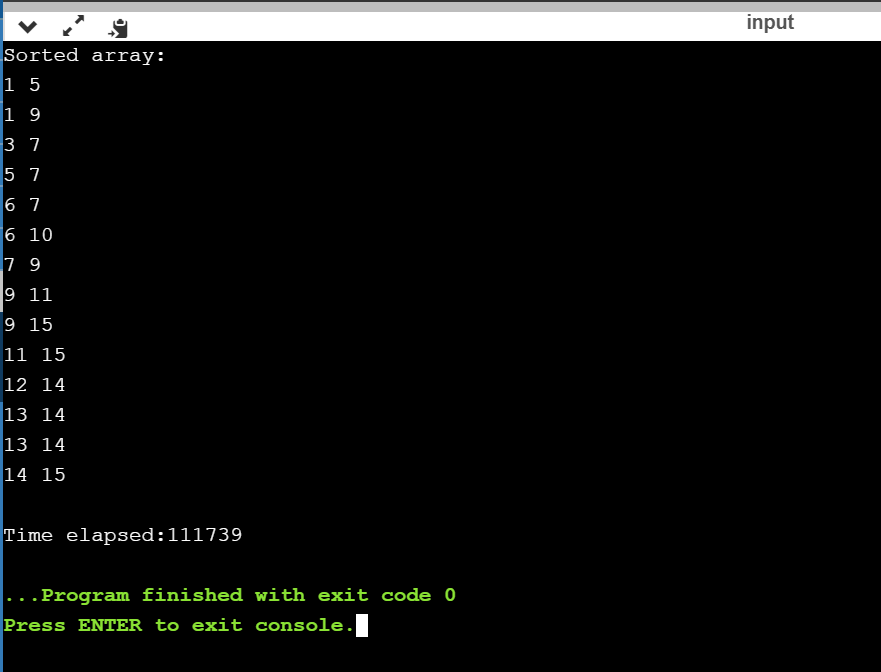
8, 10



**Test case 2:**

6, 7  
9, 11  
13, 14  
3, 7  
11, 15  
13, 14  
12, 14  
14, 15  
9, 15  
5, 7  
7, 9  
1, 5  
1, 9  
6, 10

Output:



**Conclusion:** When the execution time(in nanoseconds) was compared for fuzzy sorting and randomized quicksort, it was observed that for the same input size, fuzzy sort works faster than randomized quicksort (provided input instances are not disjoint).

**Time complexity analysis:**

**Fuzzy Sort:**

When input instances are disjoint, the worst case time complexity is O(nlogn).

When input instances overlap at one point, then the worst case time complexity reduces to less than O(nlogn) which is O(n).

**Randomized Quicksort**

Worst case time complexity of quicksort is O(nlogn).

However, if all input instances are the same numbers, then the worst case time complexity of randomized quicksort tends to be O(n^2).

**References/Citation:**

[1] For time compexity analysis

<https://stackoverflow.com/questions/45779963/randomized-quick-sort-worst-case-time-complexity>

[2] Randomized Quicksort

<https://www.geeksforgeeks.org/quicksort-using-random-pivoting/>