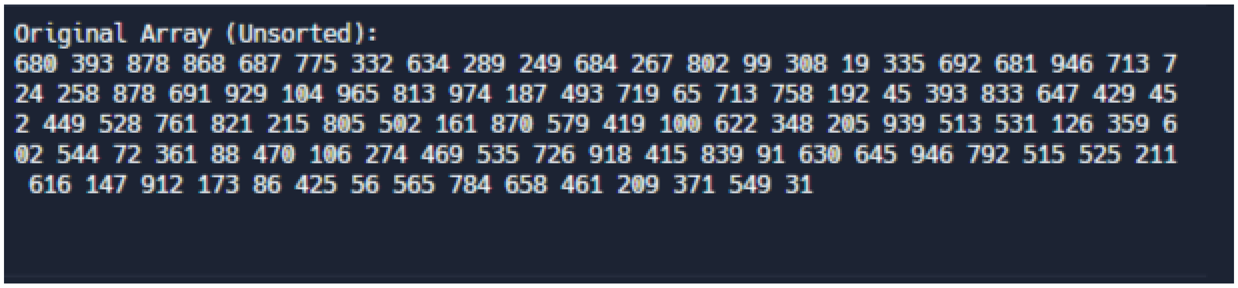
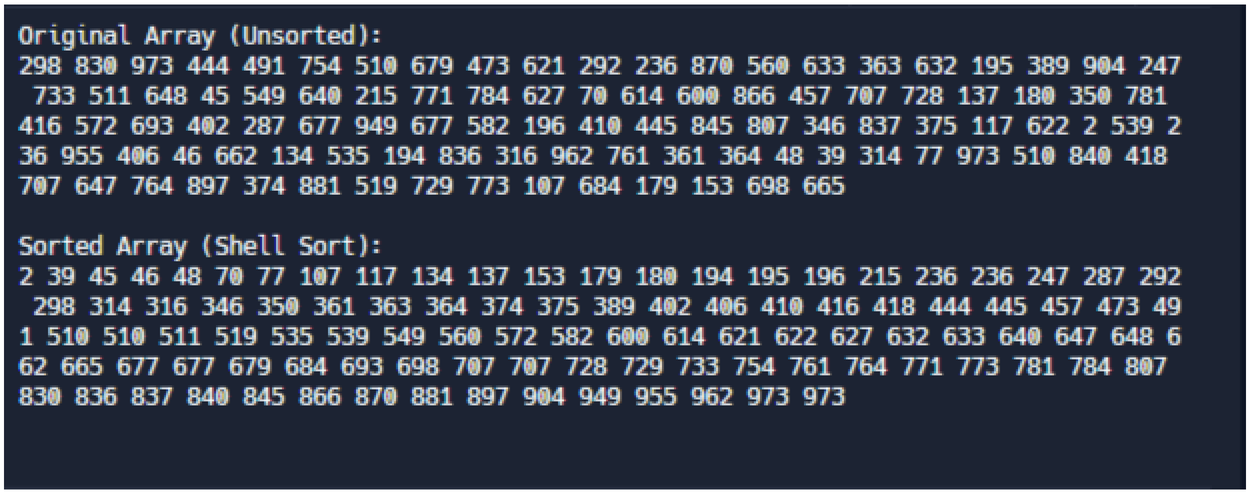
Activity No. 8	
Act 8_Sort Techniques	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed:** 10/21/24
Section: CPE21S4	**Date Submitted:** 10/21/24
Name(s): Prince Wally G. Esteban	**Instructor:** Mrs. Ma. Rizette Sayo

6. Output



```
Original Array (Unsorted):
680 393 878 868 687 775 332 634 289 249 684 267 802 99 308 19 335 692 681 946 713 7
24 258 878 691 929 104 965 813 974 187 493 719 65 713 758 192 45 393 833 647 429 45
2 449 528 761 821 215 805 502 161 870 579 419 100 622 348 205 939 513 531 126 359 6
02 544 72 361 88 470 106 274 469 535 726 918 415 839 91 630 645 946 792 515 525 211
 616 147 912 173 86 425 56 565 784 658 461 209 371 549 31
```

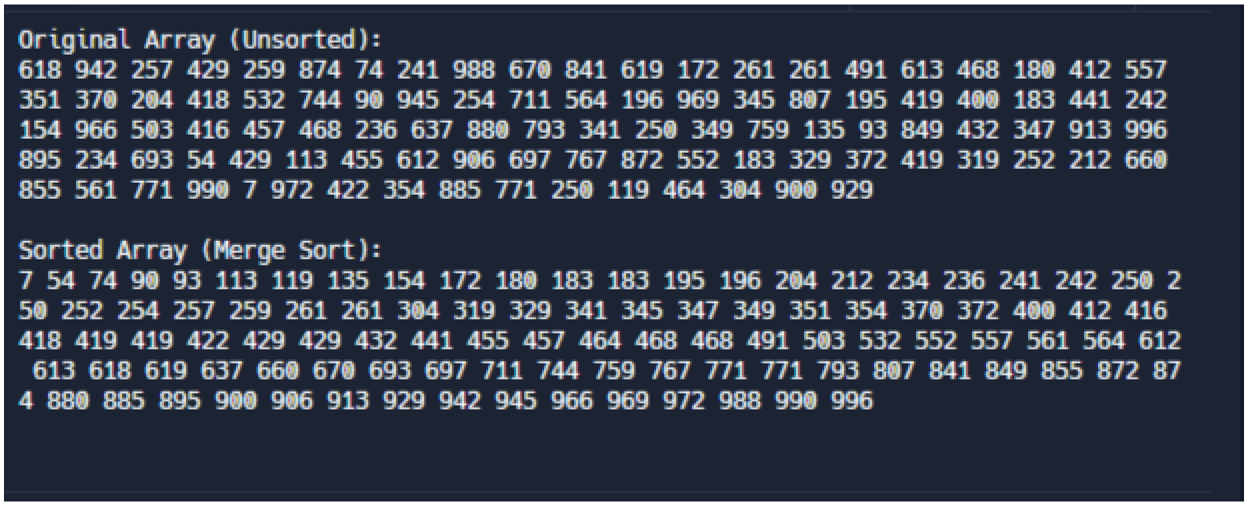
TABLE 8.1



```
Original Array (Unsorted):
298 830 973 444 491 754 510 679 473 621 292 236 870 560 633 363 632 195 389 904 247
 733 511 648 45 549 640 215 771 784 627 70 614 600 866 457 707 728 137 180 350 781
416 572 693 402 287 677 949 677 582 196 410 445 845 807 346 837 375 117 622 2 539 2
36 955 406 46 662 134 535 194 836 316 962 761 361 364 48 39 314 77 973 510 840 418
707 647 764 897 374 881 519 729 773 107 684 179 153 698 665

Sorted Array (Shell Sort):
2 39 45 46 48 70 77 107 117 134 137 153 179 180 194 195 196 215 236 236 247 287 292
 298 314 316 346 350 361 363 364 374 375 389 402 406 410 416 418 444 445 457 473 49
1 510 510 511 519 535 539 549 560 572 582 600 614 621 622 627 632 633 640 647 648 6
62 665 677 677 679 684 693 698 707 707 728 729 733 754 761 764 771 773 781 784 807
830 836 837 840 845 866 870 881 897 904 949 955 962 973 973
```

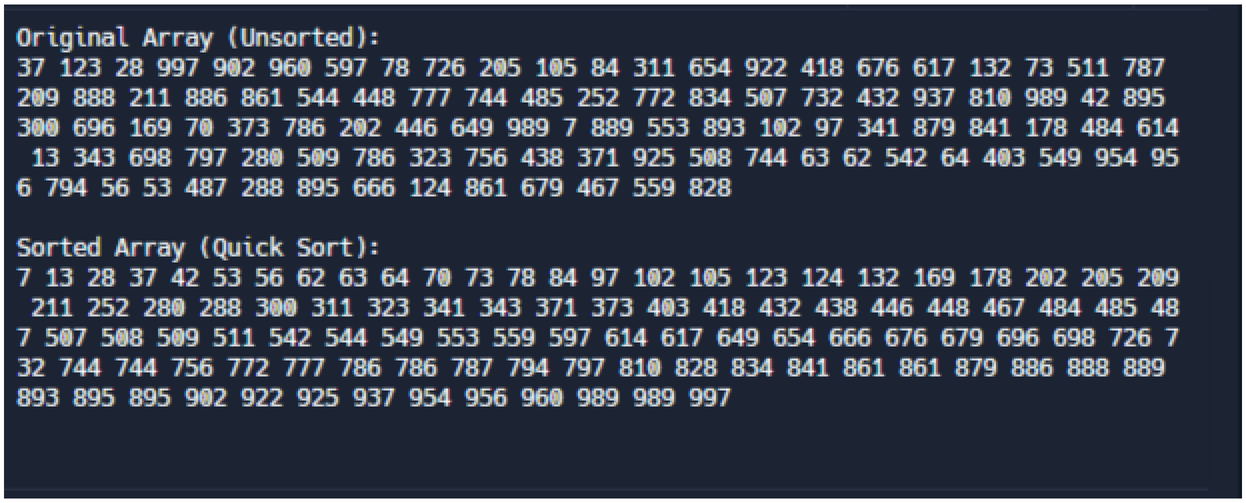
TABLE 8.2



```
Original Array (Unsorted):
618 942 257 429 259 874 74 241 988 670 841 619 172 261 261 491 613 468 180 412 557
351 370 204 418 532 744 90 945 254 711 564 196 969 345 807 195 419 400 183 441 242
154 966 503 416 457 468 236 637 880 793 341 250 349 759 135 93 849 432 347 913 996
895 234 693 54 429 113 455 612 906 697 767 872 552 183 329 372 419 319 252 212 660
855 561 771 990 7 972 422 354 885 771 250 119 464 304 900 929

Sorted Array (Merge Sort):
7 54 74 90 93 113 119 135 154 172 180 183 183 195 196 204 212 234 236 241 242 250 2
50 252 254 257 259 261 261 304 319 329 341 345 347 349 351 354 370 372 400 412 416
418 419 419 422 429 429 432 441 455 457 464 468 468 491 503 532 552 557 561 564 612
 613 618 619 637 660 670 693 697 711 744 759 767 771 771 793 807 841 849 855 872 87
4 880 885 895 900 906 913 929 942 945 966 969 972 988 990 996
```

TABLE 8.3



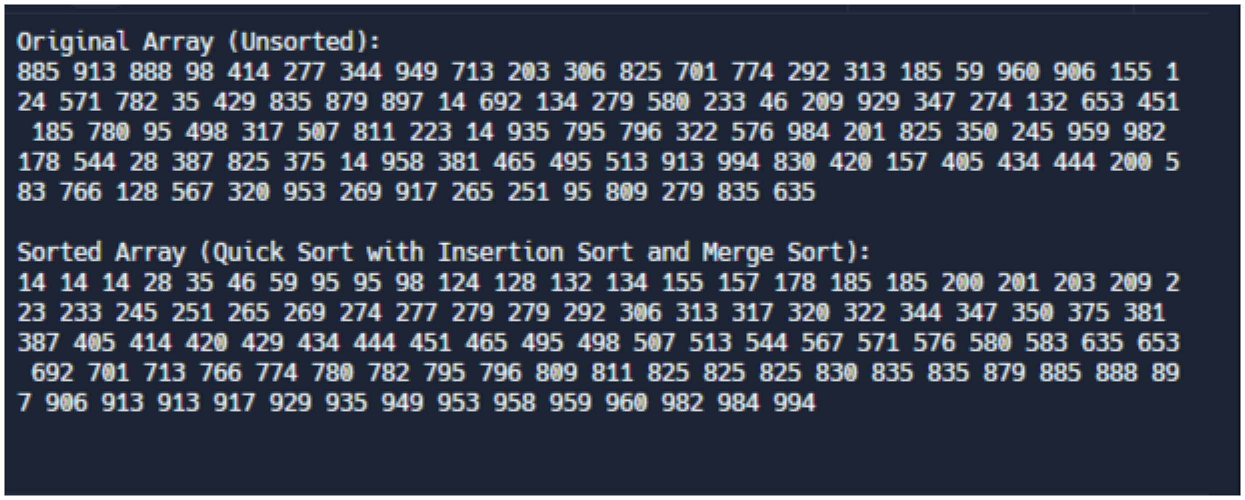
```
Original Array (Unsorted):
37 123 28 997 902 960 597 78 726 205 105 84 311 654 922 418 676 617 132 73 511 787
209 888 211 886 861 544 448 777 744 485 252 772 834 507 732 432 937 810 989 42 895
300 696 169 70 373 786 202 446 649 989 7 889 553 893 102 97 341 879 841 178 484 614
 13 343 698 797 280 509 786 323 756 438 371 925 508 744 63 62 542 64 403 549 954 95
6 794 56 53 487 288 895 666 124 861 679 467 559 828

Sorted Array (Quick Sort):
7 13 28 37 42 53 56 62 63 64 70 73 78 84 97 102 105 123 124 132 169 178 202 205 209
 211 252 280 288 300 311 323 341 343 371 373 403 418 432 438 446 448 467 484 485 48
7 507 508 509 511 542 544 549 553 559 597 614 617 649 654 666 676 679 696 698 726 7
32 744 744 756 772 777 786 786 787 794 797 810 828 834 841 861 861 879 886 888 889
893 895 895 902 922 925 937 954 956 960 989 989 997
```

TABLE 8.4

7. Supplementary Activity

PROBLEM 1:



```
Original Array (Unsorted):
885 913 888 98 414 277 344 949 713 203 306 825 701 774 292 313 185 59 960 906 155 1
24 571 782 35 429 835 879 897 14 692 134 279 580 233 46 209 929 347 274 132 653 451
 185 780 95 498 317 507 811 223 14 935 795 796 322 576 984 201 825 350 245 959 982
178 544 28 387 825 375 14 958 381 465 495 513 913 994 830 420 157 405 434 444 200 5
83 766 128 567 320 953 269 917 265 251 95 809 279 835 635

Sorted Array (Quick Sort with Insertion Sort and Merge Sort):
14 14 14 28 35 46 59 95 95 98 124 128 132 134 155 157 178 185 185 200 201 203 209 2
23 233 245 251 265 269 274 277 279 279 292 306 313 317 320 322 344 347 350 375 381
387 405 414 420 429 434 444 451 465 495 498 507 513 544 567 571 576 580 583 635 653
 692 701 713 766 774 780 782 795 796 809 811 825 825 825 830 835 835 879 885 888 89
7 906 913 913 917 929 935 949 953 958 959 960 982 984 994
```

Yes, we can use other sorting algorithms (Insertion Sort and Merge Sort) to sort the left and right sub-lists after partitioning in Quick Sort. This hybrid approach combines the strengths of different algorithms to optimize sorting performance.

PROBLEM 2:

Quick Sort is generally faster in practice, but Merge Sort's performance is more consistent across different data distributions.

8. Conclusion
In this activity, I learned about the strengths and weaknesses of various sorting algorithms, particularly Quick Sort and Merge Sort, and how to effectively combine them for optimal performance. Implementing the divide-and-conquer strategy clarified the sorting process, emphasizing the importance of careful pivot selection in Quick Sort to avoid poor performance. The supplementary activity of sorting sub-lists with different algorithms reinforced the concept of leveraging multiple sorting methods to enhance efficiency based on the data characteristics. Overall, I believe I did well in analyzing sorting algorithms, but I recognize the need to improve my understanding of algorithm theory and to better select the most suitable algorithms for different scenarios. This experience has been a valuable opportunity for growth in my knowledge of sorting techniques.
9. Assessment Rubric