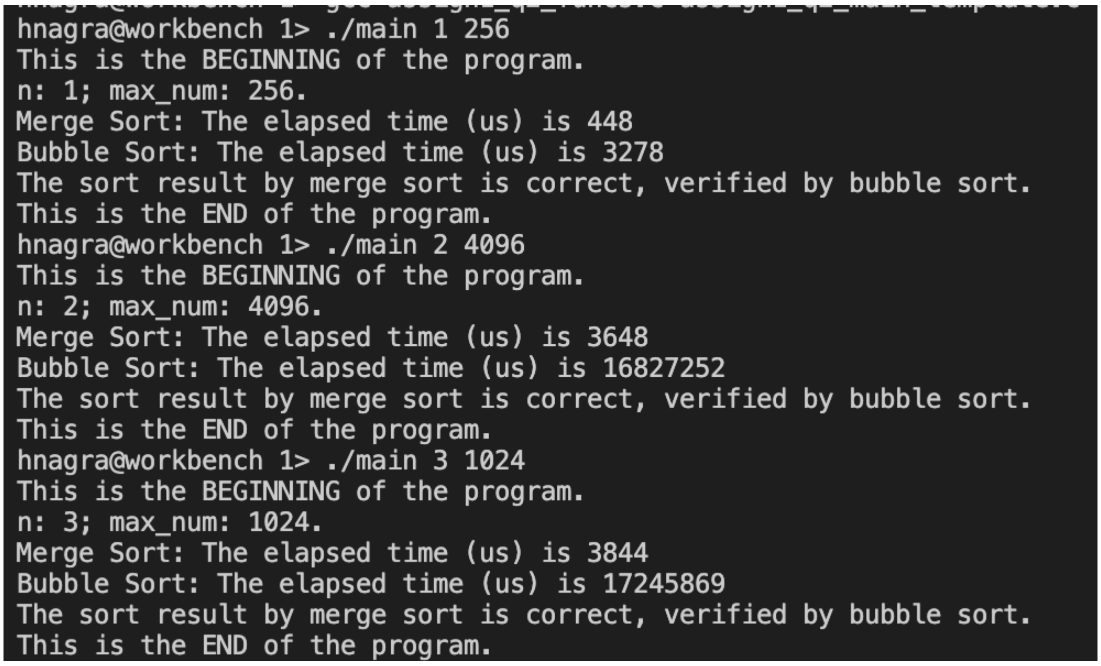
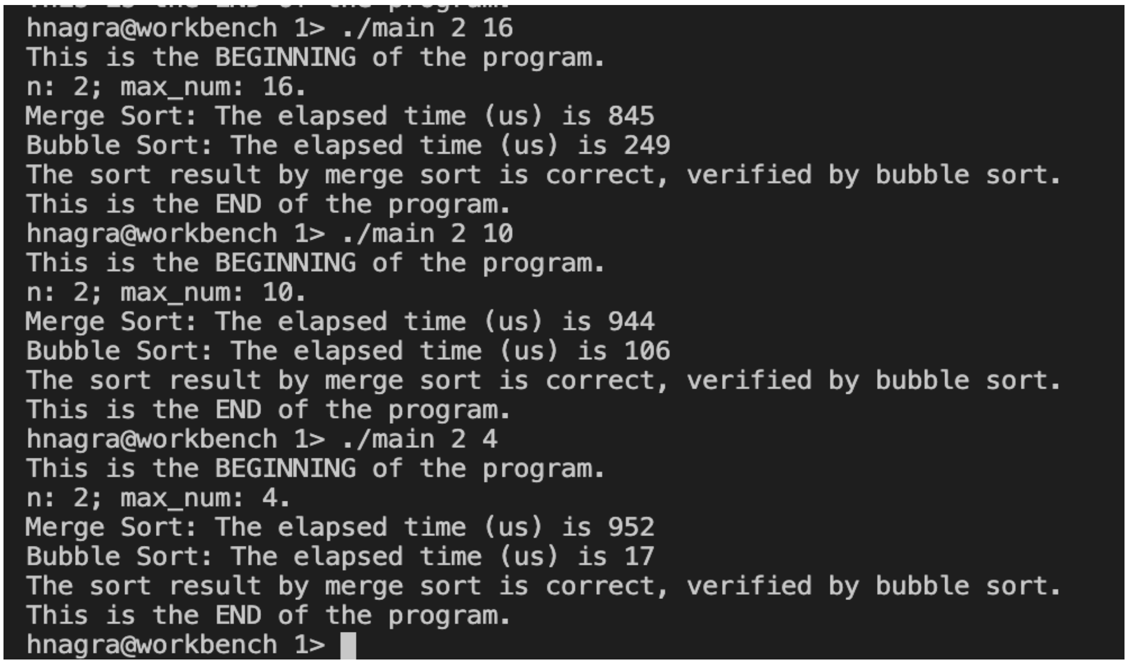
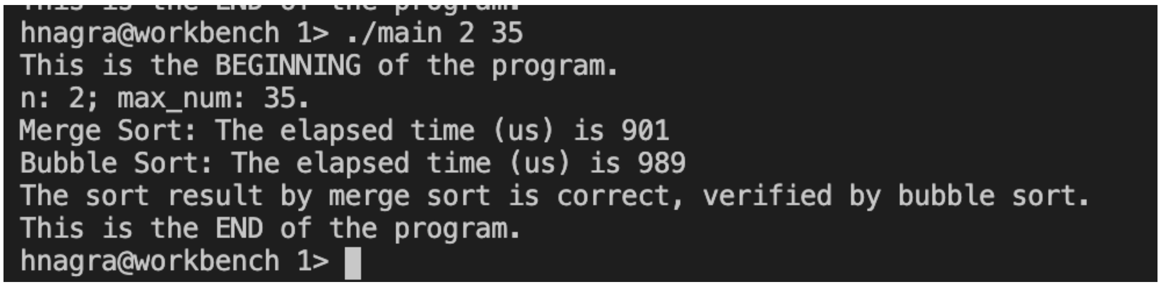


```
hnagra@workbench 1> ./main 1 256
This is the BEGINNING of the program.
n: 1; max_num: 256.
Merge Sort: The elapsed time (us) is 448
Bubble Sort: The elapsed time (us) is 3278
The sort result by merge sort is correct, verified by bubble sort.
This is the END of the program.
hnagra@workbench 1> ./main 2 4096
This is the BEGINNING of the program.
n: 2; max_num: 4096.
Merge Sort: The elapsed time (us) is 3648
Bubble Sort: The elapsed time (us) is 16827252
The sort result by merge sort is correct, verified by bubble sort.
This is the END of the program.
hnagra@workbench 1> ./main 3 1024
This is the BEGINNING of the program.
n: 3; max_num: 1024.
Merge Sort: The elapsed time (us) is 3844
Bubble Sort: The elapsed time (us) is 17245869
The sort result by merge sort is correct, verified by bubble sort.
This is the END of the program.
```

Merge Sort as seen above performs sorting much quicker than Bubble Sort as expected but the same does not apply when we are dealing with arrays of shorter length (approximate < size around 600).



```
hnagra@workbench 1> ./main 2 16
This is the BEGINNING of the program.
n: 2; max_num: 16.
Merge Sort: The elapsed time (us) is 845
Bubble Sort: The elapsed time (us) is 249
The sort result by merge sort is correct, verified by bubble sort.
This is the END of the program.
hnagra@workbench 1> ./main 2 10
This is the BEGINNING of the program.
n: 2; max_num: 10.
Merge Sort: The elapsed time (us) is 944
Bubble Sort: The elapsed time (us) is 106
The sort result by merge sort is correct, verified by bubble sort.
This is the END of the program.
hnagra@workbench 1> ./main 2 4
This is the BEGINNING of the program.
n: 2; max_num: 4.
Merge Sort: The elapsed time (us) is 952
Bubble Sort: The elapsed time (us) is 17
The sort result by merge sort is correct, verified by bubble sort.
This is the END of the program.
hnagra@workbench 1>
```



```
hnagra@workbench 1> ./main 2 35
This is the BEGINNING of the program.
n: 2; max_num: 35.
Merge Sort: The elapsed time (us) is 901
Bubble Sort: The elapsed time (us) is 989
The sort result by merge sort is correct, verified by bubble sort.
This is the END of the program.
hnagra@workbench 1>
```

1. n = 2, max_num = 4 → Speedup = 0.018
2. n = 2, max_num = 10 → Speedup = 0.112
3. n = 2, max_num = 16 → Speedup = 0.295
4. n = 2, max_num = 35 → Speedup = 1.09
5. n = 1, max_num = 256 → Speedup = 7.317
6. n = 2, max_num = 4096 → Speedup = 4,612.73
7. n = 3, max_num = 1024 → Speedup = 4,486.44

As shown above the Speedup increases exponentially with the size of the array. The larger the array, merge sort gets faster and bubble sort gets slower.

The reason behind bad performance of merge sort on a smaller data size is as splitting the array and combining the array takes up more time than rather sorting the array with a simple iterative algorithm. The bad performance of Bubble sort is due to n^2 complexity.