B. Analysis:

Average Running Time for an Input Array that is Random

N	Insertion sort	Mergesort	Quicksort	Heapsort
10	0	0	0	0
100	0	0	0	0
1000	0.00625	0	0	0
10000	0.096875	0.003125	0	0.003125
100000	8.546876	0.021875	0.01875	0.05625

Along with execution of different tries to run each sorting algorithms based on the given value of the array size, we perceived that as the array size increases the result of the running time gets slower. In fact, as we noticed the array size of 10 and 100 are 0 for all sorting algorithms but when the array size become 1000 it typically shows that there are certain changes for the value of different sorting algorithms where as in insertion sort the value tends to have a result of 0.00625 which means that it started to become slower based on a given array size. On the contrary as the array size increases and became 10000 the merge sort and the heapsort also started to get more slower given that they had the same result of 0.003125 while insertion sort continues to be slower and when the array size became 100000 all of the sorting algorithm started to be more slower running time in which insertion had a result of 8.546876, merge sort had a result of 0.021875, quicksort had a result of 0.01875 and the heap sort had a result of 0.05625. Therefore, in this given data we concluded that the Insertion sort had the highest running time value as array size increases which means that it is the slowest running time per seconds for this different type of sorting algorithms, while Quicksort had the lowest running time, therefore it is the fastest algorithm in computing the running time.

Average Running Time for an Input Array that is Sorted

N	Insertion sort	Mergesort	Quicksort	Heapsort
10	0	0	0	0
100	0	0	0	0
1000	0	0	0	0
10000	0	0	0.14375	0.00625
100000	0	0.01875	15.81562	0.028125

Based on the given data above we also perceived that there are also a variety of changes as the given array increases in the different sorting algorithm in getting their respective running time. Given that, in Insertion sort from 10 to 100000 array size they yielded on the same result for all the sorting algorithms which is 0. While on the MergeSort they output different value of running time for each array size where as in 10,100,1000,10000 they had a result of 0 but as it reaches 100000 they had a result of 0.01875.Moreover, in QuickSort from 10 to 1000 array size they also had a zero result but for 10000 they output 0.14375 and as it reaches to 100000 array size they had an output of 15.81562.On the contrary on the Heapsort algorithm from 10 to 1000 they also had a result of 0, but in 10000 they yielded to 0.00625 running time value and for 100000 they output 0.028125.Therefore we concluded that Quicksort had the most highest running time value for all the given sorting algorithm that lead to became slowest computing running time as it reaches the 100000 array size and the fastest in computing the running time is insertion sort.

Note: We attach an excel file for the computation of the average of the given sorting algorithm in their 5 times run.

Random

```
erminoyhana@DESKTOP-OVF7RFK:~$ nano da.cpp
erminoyhana@DESKTOP-OVF7RFK:~$ g++ da.cpp -o da
erminoyhana@DESKTOP-OVF7RFK:~$ ./da
--THIS PROGRAM USED DIFERENT SORTING ALGORITHMS(INSERTION,MERGE,QUICK,HEAP) TO
SORT RANDOMLY GENERATED INTEGERS AND THEN ANALYZE THEIR RUN TIME--

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 1
Enter a value that will determine the number of integers in the Array: 10

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0
```

```
Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 1
Enter a value that will determine the number of integers in the Array: 100

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0
```

```
Enter selection: 1
Enter a value that will determine the number of integers in the Array: 1000

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0.015625
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0
```

```
Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 1
Enter a value that will determine the number of integers in the Array: 10000

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0.09375
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0.015625
```

```
Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 1
Enter a value that will determine the number of integers in the Array: 100000

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 8.54688
Time taken in Merge sort is 0.015625
Time taken in Quick sort is 0.015625
Time taken in Heap sort is 0.078125
```

```
Choose a program that will Generate the Integers
      [1]. Random
      [2]. Sorted
      [3]. End program

Enter selection: 1
Enter a value that will determine the number of integers in the Array: 100000

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 8.78125
Time taken in Merge sort is 0.03125
Time taken in Quick sort is 0.015625
Time taken in Heap sort is 0.046875
```

Sorted

```
erminoyhana@DESKTOP-OVF7RFK:~$ g++ da.cpp -o da
erminoyhana@DESKTOP-OVF7RFK:~$ ./da
--THIS PROGRAM USED DIFERENT SORTING ALGORITHMS(INSERTION,MERGE,QUICK,HEAP) TO
SORT RANDOMLY GENERATED INTEGERS AND THEN ANALYZE THEIR RUN TIME--

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 10
Enter a positive Integer: 1

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0
```

```
Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 100
Enter a positive Integer: 13

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0
```

```
Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 1000
Enter a positive Integer: 12367

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0
```

```
Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 10000
Enter a positive Integer: 27

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0.171875
Time taken in Heap sort is 0
```

```
Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 100000
Enter a positive Integer: 10

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0.015625
Time taken in Quick sort is 17.2031
Time taken in Heap sort is 0.015625
```

CODE STRUCTURE

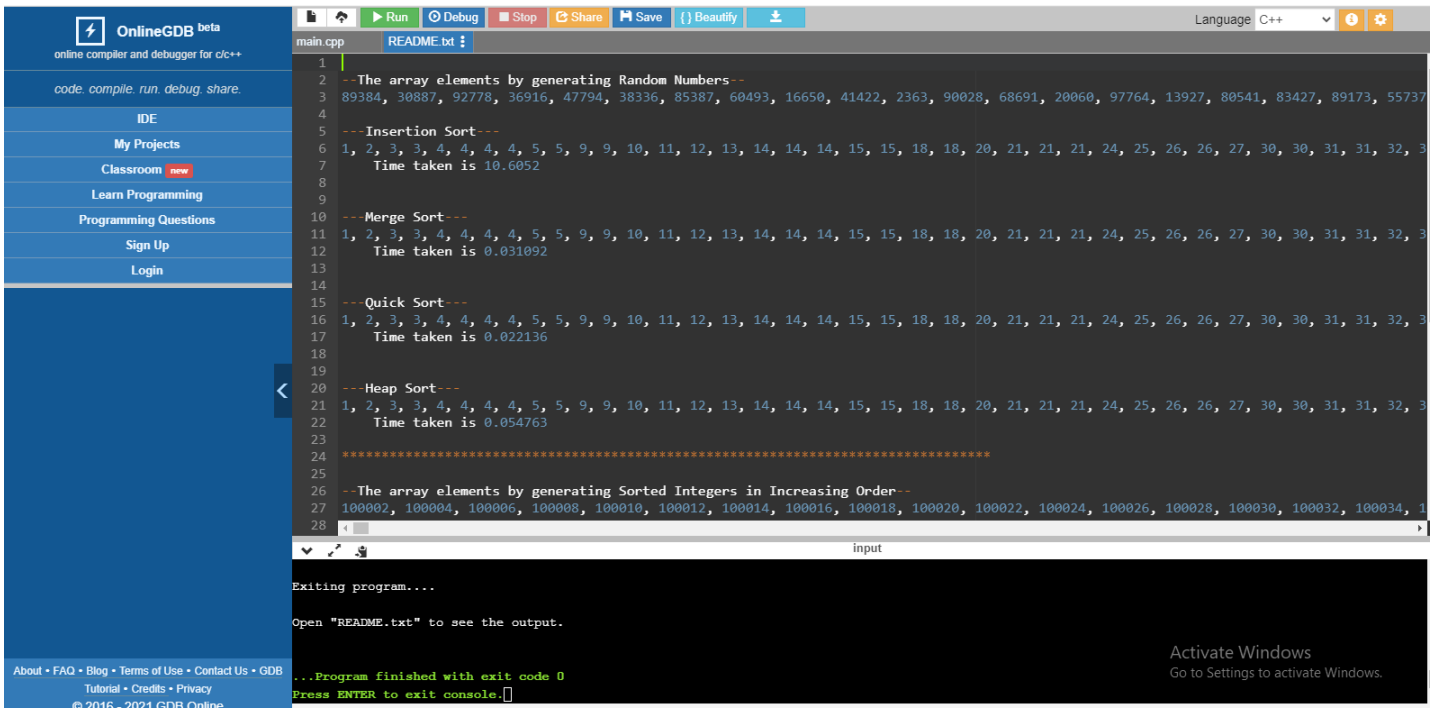
- The program asks the user to choose a program that will generate the elements of the array.
- You will choose between 1, 2, and 3. The first one is Generating the elements randomly using the function rand(), the picked numbers has a range from 0 to 1000000. The second one is Generating increasing sorted Integers by asking the user to input a positive integer X, then compute N + (i + 1) * X . i is the incrementing loop variable. The third one is to terminate the program.
- If you choose 1 or 2, the program will ask the user to input a number that will determine the size of the array and store it to N variable. (10,100,1000,10000,100000)
- The program will now generate the inputs of the array and sort the inputs into increasing order using 4 different Sorting Algorithms (Insertion Sort, Merge Sort, Quick Sort, Heap Sort). The sorting algorithms were implemented using functions.
- Next in the main function it calls a function named sorting which calls all sorting algorithms function the function also measure the time taken by the process of specific sorting algorithm.
- The start = clock(); function was declared before the calling function of specific sorting algorithm and end = clock(); function was declared after the call function.
- After the first sorting technique before proceeding to the next call the program calls a function called copy that will copy the original order of the array from the temporary array named tempArr[]; then it proceeds to the next sorting algorithm doing the same process until all sorting algorithm is called.
- The program will only terminate if you input number 3 when the program asks for selection.
- The output is outputted in an output file “README.txt” that shows the List of Array before sorting and the list of arrays after using 4 different sorting algorithms.
- In the output file, it also displays the time take by the process of specific sorting algorithms

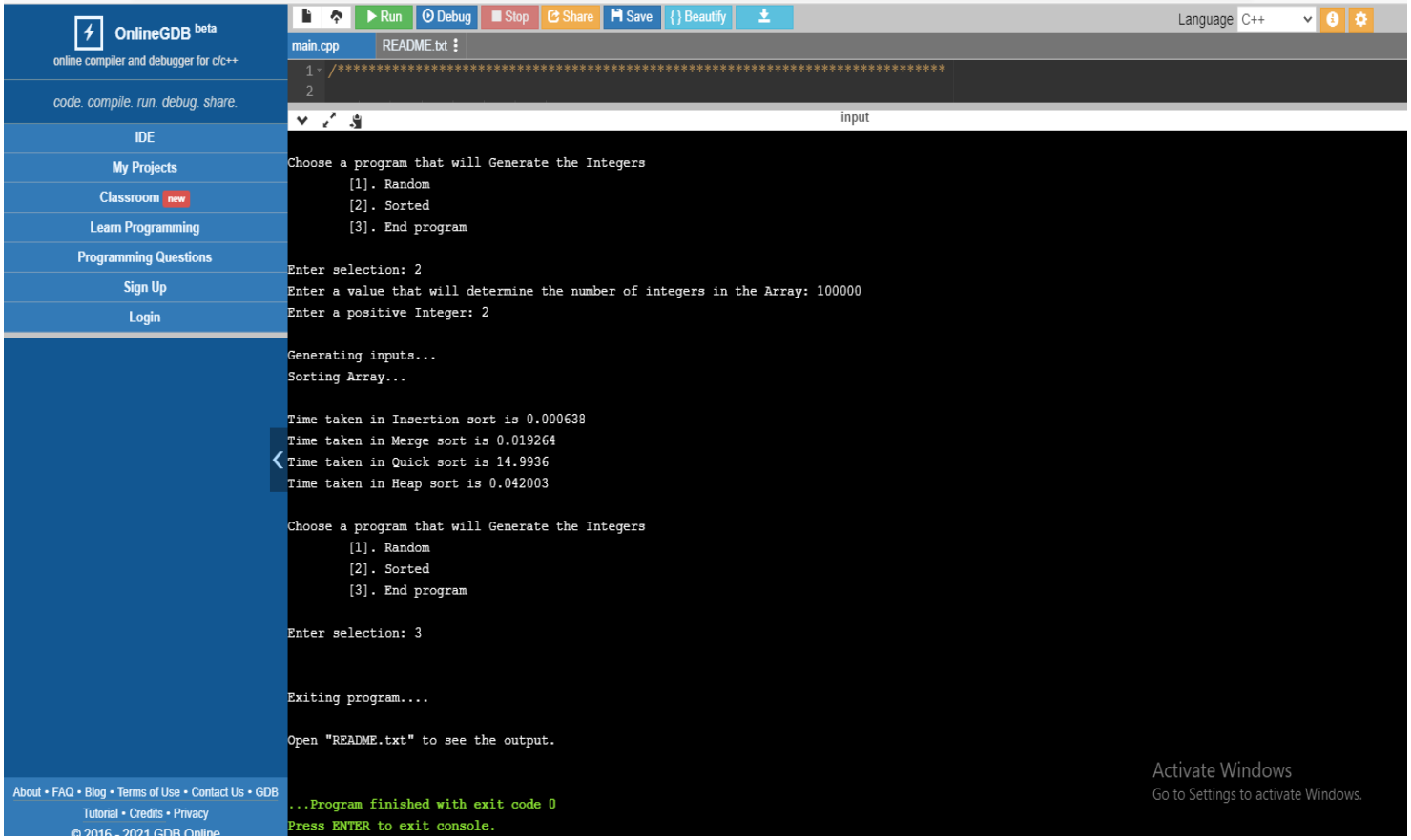
PROBLEMS ENCOUNTERED

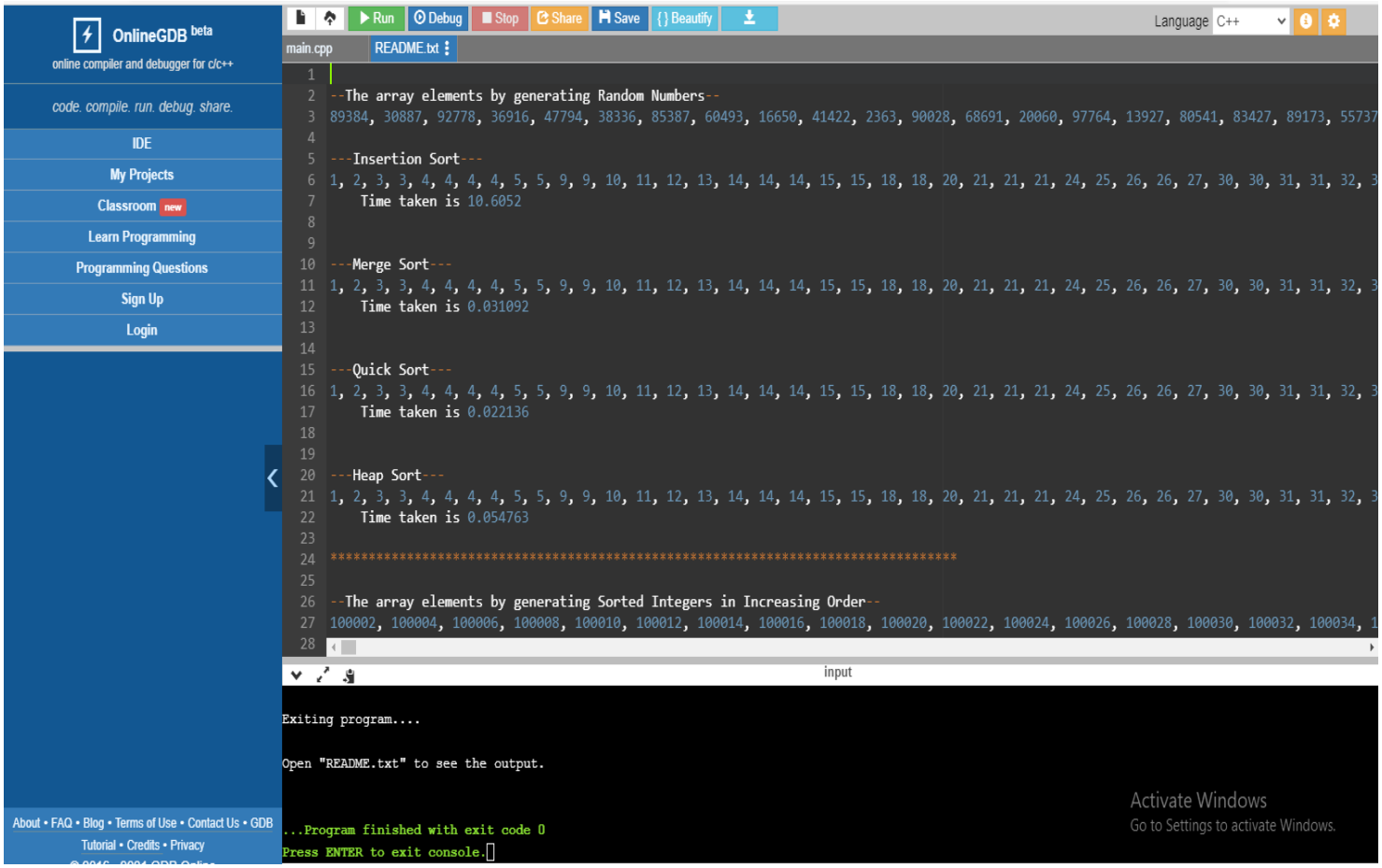
- The problem that we encountered is when we compile the program with 100k inputs. We used dev C++. If the user selected number 2 which is generating the input in increasing order, the program terminates after insertion sort and mergesort call function.

```
Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 100000
Enter a positive Integer: 2

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0.000638
Time taken in Merge sort is 0.019264
Time taken in Quick sort is 14.9936
Time taken in Heap sort is 0.042003

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 3


Exiting program....

Open "README.txt" to see the output.


...Program finished with exit code 0
Press ENTER to exit console.
```


- To solve the problem, we try to compile the program online and the program works.




OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
README.txt
input
Choose a program that will Generate the Integers
[1]. Random
[2]. Sorted
[3]. End program
Enter selection: 2
Enter a value that will determine the number of integers in the Array: 100000
Enter a positive Integer: 2
Generating inputs...
Sorting Array...
Time taken in Insertion sort is 0.000638
Time taken in Merge sort is 0.019264
Time taken in Quick sort is 14.9936
Time taken in Heap sort is 0.042003
Choose a program that will Generate the Integers
[1]. Random
[2]. Sorted
[3]. End program
Enter selection: 3
Exiting program....
Open "README.txt" to see the output.
...Program finished with exit code 0
Press ENTER to exit console.
Activate Windows
Go to Settings to activate Windows.


OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
README.txt
--The array elements by generating Random Numbers--
89384, 30887, 92778, 36916, 47794, 38336, 85387, 60493, 16650, 41422, 2363, 90028, 68691, 20060, 97764, 13927, 80541, 83427, 89173, 55737
---Insertion Sort---
1, 2, 3, 3, 4, 4, 4, 4, 5, 5, 9, 9, 10, 11, 12, 13, 14, 14, 14, 15, 15, 18, 18, 20, 21, 21, 21, 24, 25, 26, 26, 27, 30, 30, 31, 31, 32, 3
Time taken is 10.6052
---Merge Sort---
1, 2, 3, 3, 4, 4, 4, 4, 5, 5, 9, 9, 10, 11, 12, 13, 14, 14, 14, 15, 15, 18, 18, 20, 21, 21, 21, 24, 25, 26, 26, 27, 30, 30, 31, 31, 32, 3
Time taken is 0.031092
---Quick Sort---
1, 2, 3, 3, 4, 4, 4, 4, 5, 5, 9, 9, 10, 11, 12, 13, 14, 14, 14, 15, 15, 18, 18, 20, 21, 21, 21, 24, 25, 26, 26, 27, 30, 30, 31, 31, 32, 3
Time taken is 0.022136
---Heap Sort---
1, 2, 3, 3, 4, 4, 4, 4, 5, 5, 9, 9, 10, 11, 12, 13, 14, 14, 14, 15, 15, 18, 18, 20, 21, 21, 21, 24, 25, 26, 26, 27, 30, 30, 31, 31, 32, 3
Time taken is 0.054763
--The array elements by generating Sorted Integers in Increasing Order--
100002, 100004, 100006, 100008, 100010, 100012, 100014, 100016, 100018, 100020, 100022, 100024, 100026, 100028, 100030, 100032, 100034, 1
input
Exiting program....
Open "README.txt" to see the output.
...Program finished with exit code 0
Press ENTER to exit console.
Activate Windows
Go to Settings to activate Windows.

SCREENSHOTS OF EXECUTION

N=10

```
--THIS PROGRAM USED DIFERENT SORTING ALGORITHMS(INSERTION,MERGE,QUICK,HEAP) TO
SORT RANDOMLY GENERATED INTEGERS AND THEN ANALYZE THEIR RUN TIME--

Choose a program that will Generate the Integers
       [1]. Random
       [2]. Sorted
       [3]. End program

Enter selection: 1
Enter a value that will determine the number of integers in the Array: 10

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0

Choose a program that will Generate the Integers
       [1]. Random
       [2]. Sorted
       [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 10
Enter a positive Integer: 12

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0

Choose a program that will Generate the Integers
       [1]. Random
       [2]. Sorted
       [3]. End program

Enter selection: 3


Exiting program....

Open "README.txt" to see the output.

--------------------------------
Process exited after 13.97 seconds with return value 0
Press any key to continue . . .
```


README - Notepad

File Edit Format View Help

```
--The array elements by generating Random Numbers--
41, 18467, 6334, 26500, 19169, 15724, 11478, 29358, 26962, 24464,

---Insertion Sort---
41, 6334, 11478, 15724, 18467, 19169, 24464, 26500, 26962, 29358,
        Time taken is 0


---Merge Sort---
41, 6334, 11478, 15724, 18467, 19169, 24464, 26500, 26962, 29358,
        Time taken is 0


---Quick Sort---
41, 6334, 11478, 15724, 18467, 19169, 24464, 26500, 26962, 29358,
        Time taken is 0


---Heap Sort---
41, 6334, 11478, 15724, 18467, 19169, 24464, 26500, 26962, 29358,
        Time taken is 0

********************************************************************************

--The array elements by generating Sorted Integers in Increasing Order--
22, 34, 46, 58, 70, 82, 94, 106, 118, 130,

---Insertion Sort---
22, 34, 46, 58, 70, 82, 94, 106, 118, 130,
        Time taken is 0


---Merge Sort---
22, 34, 46, 58, 70, 82, 94, 106, 118, 130,
        Time taken is 0


---Quick Sort---
22, 34, 46, 58, 70, 82, 94, 106, 118, 130,
        Time taken is 0
```

N=100

```
C:\Users\HP\Documents\C++ CODE\programmingAss1.exe
--THIS PROGRAM USED DIFERENT SORTING ALGORITHMS(INSERTION,MERGE,QUICK,HEAP) TO
SORT RANDOMLY GENERATED INTEGERS AND THEN ANALYZE THEIR RUN TIME--

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 1
Enter a value that will determine the number of integers in the Array: 100

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 100
Enter a positive Integer: 20

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 3


Exiting program....

Open "README.txt" to see the output.

--------------------------------
Process exited after 13.2 seconds with return value 0
Press any key to continue . . .
```

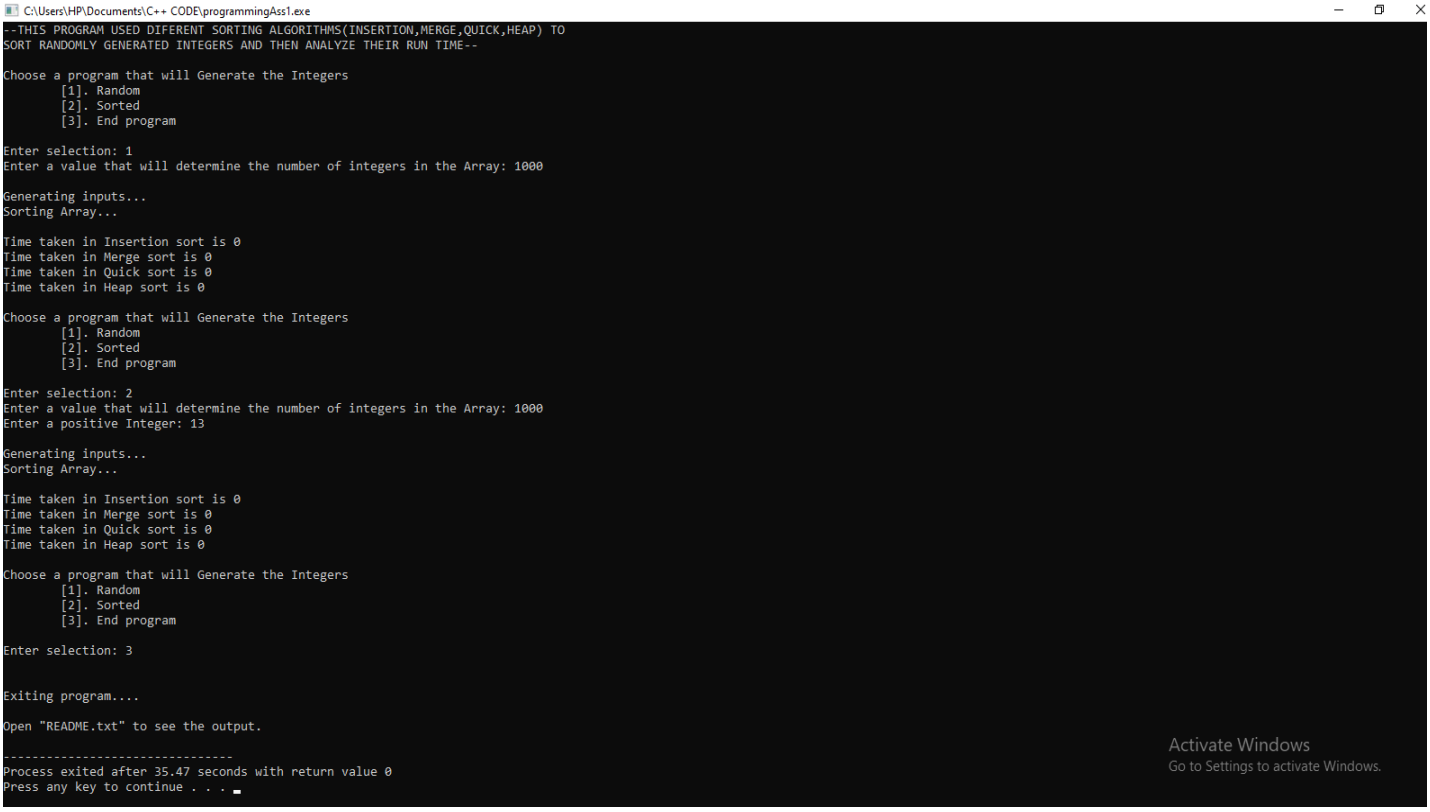

README - Notepad

File Edit Format View Help

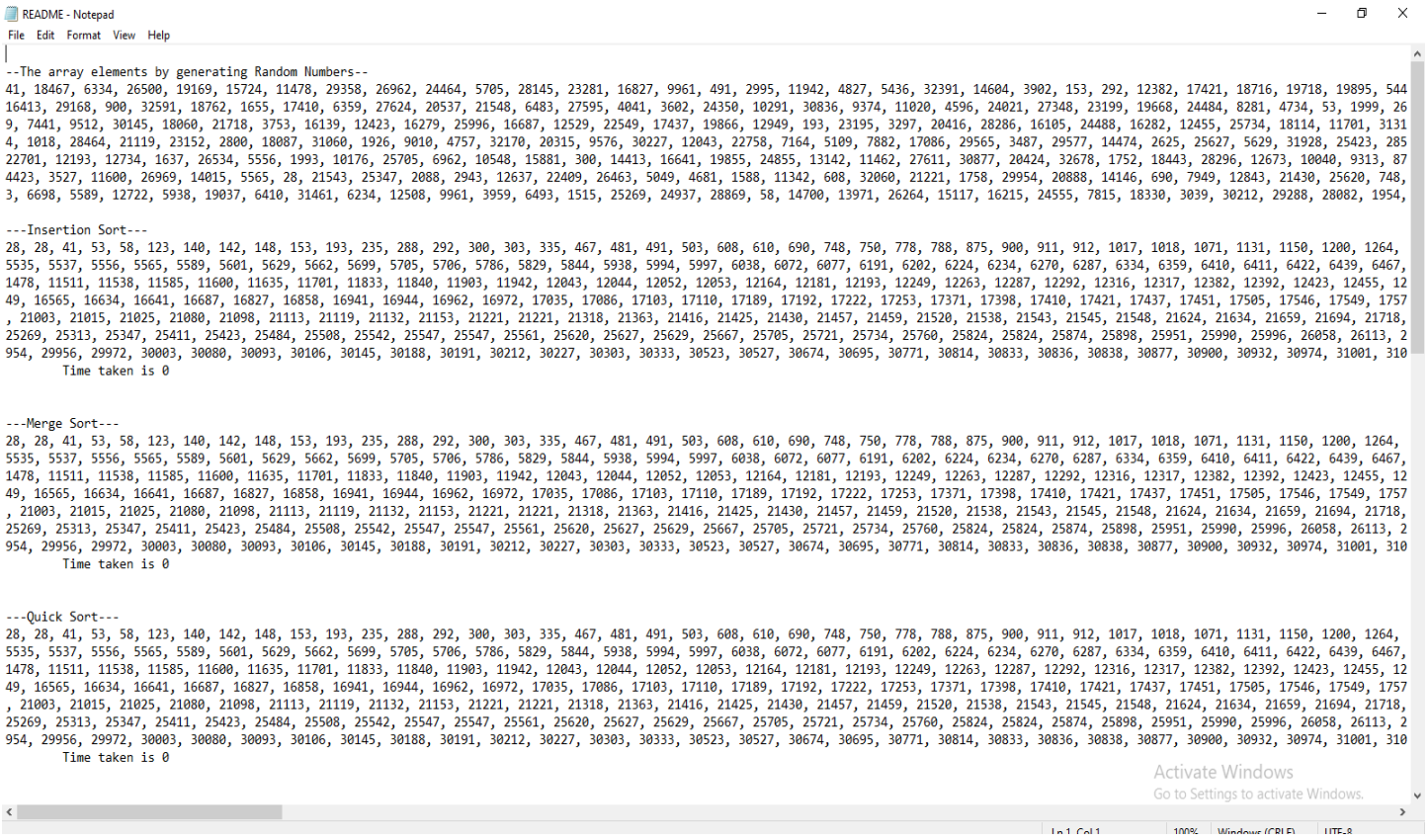
```
--The array elements by generating Random Numbers--
41, 18467, 6334, 26500, 19169, 15724, 11478, 29358, 26962, 24464, 5705, 28145, 23281, 16827, 9961, 491, 2995, 11942, 4827, 5436, 32391, 14604, 3902, 153, 292, 12382, 17421, 18716, 19718, 19895, 544

---Insertion Sort---
41, 153, 288, 292, 491, 778, 1842, 1869, 2082, 2995, 3035, 3548, 3902, 4664, 4827, 4966, 5436, 5447, 5537, 5705, 6334, 6729, 6868, 7376, 7711, 8723, 8942, 9040, 9741, 9894, 9961, 11323, 11478, 1153
        Time taken is 0


---Merge Sort---
41, 153, 288, 292, 491, 778, 1842, 1869, 2082, 2995, 3035, 3548, 3902, 4664, 4827, 4966, 5436, 5447, 5537, 5705, 6334, 6729, 6868, 7376, 7711, 8723, 8942, 9040, 9741, 9894, 9961, 11323, 11478, 1153
        Time taken is 0


---Quick Sort---
41, 153, 288, 292, 491, 778, 1842, 1869, 2082, 2995, 3035, 3548, 3902, 4664, 4827, 4966, 5436, 5447, 5537, 5705, 6334, 6729, 6868, 7376, 7711, 8723, 8942, 9040, 9741, 9894, 9961, 11323, 11478, 1153
        Time taken is 0


---Heap Sort---
41, 153, 288, 292, 491, 778, 1842, 1869, 2082, 2995, 3035, 3548, 3902, 4664, 4827, 4966, 5436, 5447, 5537, 5705, 6334, 6729, 6868, 7376, 7711, 8723, 8942, 9040, 9741, 9894, 9961, 11323, 11478, 1153
        Time taken is 0

*********************************************************************************

--The array elements by generating Sorted Integers in Increasing Order--
120, 140, 160, 180, 200, 220, 240, 260, 280, 300, 320, 340, 360, 380, 400, 420, 440, 460, 480, 500, 520, 540, 560, 580, 600, 620, 640, 660, 680, 700, 720, 740, 760, 780, 800, 820, 840, 860, 880, 90

---Insertion Sort---
120, 140, 160, 180, 200, 220, 240, 260, 280, 300, 320, 340, 360, 380, 400, 420, 440, 460, 480, 500, 520, 540, 560, 580, 600, 620, 640, 660, 680, 700, 720, 740, 760, 780, 800, 820, 840, 860, 880, 90
        Time taken is 0


---Merge Sort---
120, 140, 160, 180, 200, 220, 240, 260, 280, 300, 320, 340, 360, 380, 400, 420, 440, 460, 480, 500, 520, 540, 560, 580, 600, 620, 640, 660, 680, 700, 720, 740, 760, 780, 800, 820, 840, 860, 880, 90
        Time taken is 0


---Quick Sort---
120, 140, 160, 180, 200, 220, 240, 260, 280, 300, 320, 340, 360, 380, 400, 420, 440, 460, 480, 500, 520, 540, 560, 580, 600, 620, 640, 660, 680, 700, 720, 740, 760, 780, 800, 820, 840, 860, 880, 90
        Time taken is 0
```


Ln 1, Col 1 100% Windows (CRLF) UTF-8

N=1000



```
C:\Users\HP\Documents\C++ CODE\programmingAss1.exe
--THIS PROGRAM USED DIFERENT SORTING ALGORITHMS(INSERTION,MERGE,QUICK,HEAP) TO
SORT RANDOMLY GENERATED INTEGERS AND THEN ANALYZE THEIR RUN TIME--

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 1
Enter a value that will determine the number of integers in the Array: 1000

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 1000
Enter a positive Integer: 13

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 3


Exiting program....

Open "README.txt" to see the output.

--------------------------------
Process exited after 35.47 seconds with return value 0
Press any key to continue . . .
```

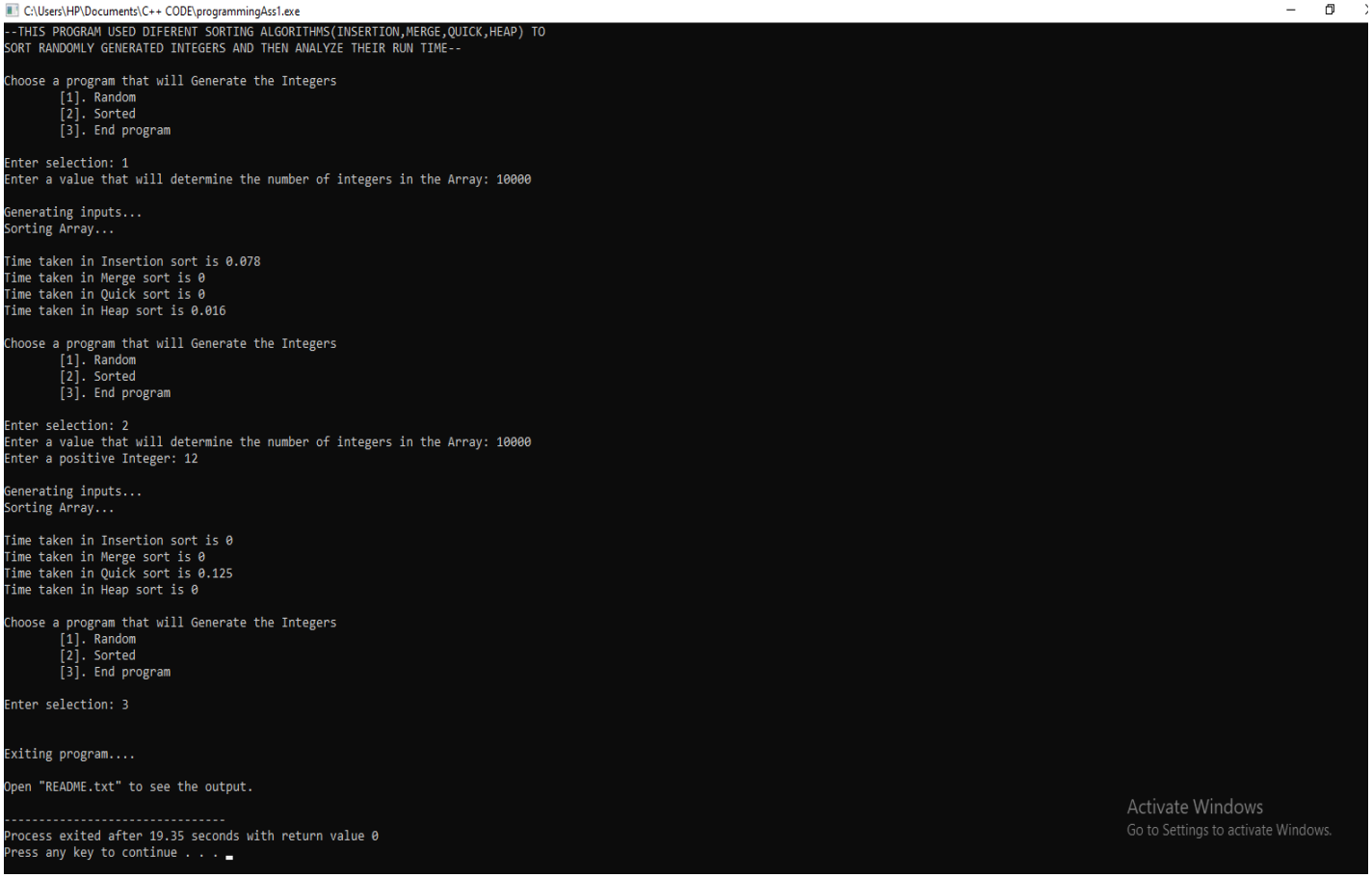
README - Notepad

File Edit Format View Help

```
--The array elements by generating Random Numbers--
41, 18467, 6334, 26500, 19169, 15724, 11478, 29358, 26962, 24464, 5705, 28145, 23281, 16827, 9961, 491, 2995, 11942, 4827, 5436, 32391, 14604, 3902, 153, 292, 12382, 17421, 18716, 19718, 19895, 544
16413, 29168, 900, 32591, 18762, 1655, 17410, 6359, 27624, 20537, 21548, 6483, 27595, 4041, 3602, 24350, 10291, 30836, 9374, 11020, 4596, 24021, 27348, 23199, 19668, 24484, 8281, 4734, 53, 1999, 26
9, 7441, 9512, 30145, 18060, 21718, 3753, 16139, 12423, 16279, 25996, 16687, 12529, 22549, 17437, 19866, 12949, 193, 23195, 3297, 20416, 28286, 16105, 24488, 16282, 12455, 25734, 18114, 11701, 3131
4, 1018, 28464, 21119, 23152, 2800, 18087, 31060, 1926, 9010, 4757, 32170, 20315, 9576, 30227, 12043, 22758, 7164, 5109, 7882, 17086, 29565, 3487, 29577, 14474, 2625, 25627, 5629, 31928, 25423, 285
22701, 12193, 12734, 1637, 26534, 5556, 1993, 10176, 25705, 6962, 10548, 15881, 300, 14413, 16641, 19855, 24855, 13142, 11462, 27611, 30877, 20424, 32678, 1752, 18443, 28296, 12673, 10040, 9313, 87
4423, 3527, 11600, 26969, 14015, 5565, 28, 21543, 25347, 2088, 2943, 12637, 22409, 26463, 5049, 4681, 1588, 11342, 608, 32060, 21221, 1758, 29954, 20888, 14146, 690, 7949, 12843, 21430, 25620, 748,
3, 6698, 5589, 12722, 5938, 19037, 6410, 31461, 6234, 12508, 9961, 3959, 6493, 1515, 25269, 24937, 28869, 58, 14700, 13971, 26264, 15117, 16215, 24555, 7815, 18330, 3039, 30212, 29288, 28082, 1954,

---Insertion Sort---
28, 28, 41, 53, 58, 123, 140, 142, 148, 153, 193, 235, 288, 292, 300, 303, 335, 467, 481, 491, 503, 608, 610, 690, 748, 750, 778, 788, 875, 900, 911, 912, 1017, 1018, 1071, 1131, 1150, 1200, 1264,
5535, 5537, 5556, 5565, 5589, 5601, 5629, 5662, 5699, 5705, 5706, 5786, 5829, 5844, 5938, 5994, 5997, 6038, 6072, 6077, 6191, 6202, 6224, 6234, 6270, 6287, 6334, 6359, 6410, 6411, 6422, 6439, 6467,
1478, 11511, 11538, 11585, 11600, 11635, 11701, 11833, 11840, 11903, 11942, 12043, 12044, 12052, 12053, 12164, 12181, 12193, 12249, 12263, 12287, 12292, 12316, 12317, 12382, 12392, 12423, 12455, 12
49, 16565, 16634, 16641, 16687, 16827, 16858, 16941, 16944, 16962, 16972, 17035, 17086, 17103, 17110, 17189, 17192, 17222, 17253, 17371, 17398, 17410, 17421, 17437, 17451, 17505, 17546, 17549, 1757
, 21003, 21015, 21025, 21080, 21098, 21113, 21119, 21132, 21153, 21221, 21221, 21318, 21363, 21416, 21425, 21430, 21457, 21459, 21520, 21538, 21543, 21545, 21548, 21624, 21634, 21659, 21694, 21718,
25269, 25313, 25347, 25411, 25423, 25484, 25508, 25542, 25547, 25547, 25561, 25620, 25627, 25629, 25667, 25705, 25721, 25734, 25760, 25824, 25824, 25874, 25898, 25951, 25990, 25996, 26058, 26113, 2
954, 29956, 29972, 30003, 30080, 30093, 30106, 30145, 30188, 30191, 30212, 30227, 30303, 30333, 30523, 30527, 30674, 30695, 30771, 30814, 30833, 30836, 30838, 30877, 30900, 30932, 30974, 31001, 310
        Time taken is 0


---Merge Sort---
28, 28, 41, 53, 58, 123, 140, 142, 148, 153, 193, 235, 288, 292, 300, 303, 335, 467, 481, 491, 503, 608, 610, 690, 748, 750, 778, 788, 875, 900, 911, 912, 1017, 1018, 1071, 1131, 1150, 1200, 1264,
5535, 5537, 5556, 5565, 5589, 5601, 5629, 5662, 5699, 5705, 5706, 5786, 5829, 5844, 5938, 5994, 5997, 6038, 6072, 6077, 6191, 6202, 6224, 6234, 6270, 6287, 6334, 6359, 6410, 6411, 6422, 6439, 6467,
1478, 11511, 11538, 11585, 11600, 11635, 11701, 11833, 11840, 11903, 11942, 12043, 12044, 12052, 12053, 12164, 12181, 12193, 12249, 12263, 12287, 12292, 12316, 12317, 12382, 12392, 12423, 12455, 12
49, 16565, 16634, 16641, 16687, 16827, 16858, 16941, 16944, 16962, 16972, 17035, 17086, 17103, 17110, 17189, 17192, 17222, 17253, 17371, 17398, 17410, 17421, 17437, 17451, 17505, 17546, 17549, 1757
, 21003, 21015, 21025, 21080, 21098, 21113, 21119, 21132, 21153, 21221, 21221, 21318, 21363, 21416, 21425, 21430, 21457, 21459, 21520, 21538, 21543, 21545, 21548, 21624, 21634, 21659, 21694, 21718,
25269, 25313, 25347, 25411, 25423, 25484, 25508, 25542, 25547, 25547, 25561, 25620, 25627, 25629, 25667, 25705, 25721, 25734, 25760, 25824, 25824, 25874, 25898, 25951, 25990, 25996, 26058, 26113, 2
954, 29956, 29972, 30003, 30080, 30093, 30106, 30145, 30188, 30191, 30212, 30227, 30303, 30333, 30523, 30527, 30674, 30695, 30771, 30814, 30833, 30836, 30838, 30877, 30900, 30932, 30974, 31001, 310
        Time taken is 0


---Quick Sort---
28, 28, 41, 53, 58, 123, 140, 142, 148, 153, 193, 235, 288, 292, 300, 303, 335, 467, 481, 491, 503, 608, 610, 690, 748, 750, 778, 788, 875, 900, 911, 912, 1017, 1018, 1071, 1131, 1150, 1200, 1264,
5535, 5537, 5556, 5565, 5589, 5601, 5629, 5662, 5699, 5705, 5706, 5786, 5829, 5844, 5938, 5994, 5997, 6038, 6072, 6077, 6191, 6202, 6224, 6234, 6270, 6287, 6334, 6359, 6410, 6411, 6422, 6439, 6467,
1478, 11511, 11538, 11585, 11600, 11635, 11701, 11833, 11840, 11903, 11942, 12043, 12044, 12052, 12053, 12164, 12181, 12193, 12249, 12263, 12287, 12292, 12316, 12317, 12382, 12392, 12423, 12455, 12
49, 16565, 16634, 16641, 16687, 16827, 16858, 16941, 16944, 16962, 16972, 17035, 17086, 17103, 17110, 17189, 17192, 17222, 17253, 17371, 17398, 17410, 17421, 17437, 17451, 17505, 17546, 17549, 1757
, 21003, 21015, 21025, 21080, 21098, 21113, 21119, 21132, 21153, 21221, 21221, 21318, 21363, 21416, 21425, 21430, 21457, 21459, 21520, 21538, 21543, 21545, 21548, 21624, 21634, 21659, 21694, 21718,
25269, 25313, 25347, 25411, 25423, 25484, 25508, 25542, 25547, 25547, 25561, 25620, 25627, 25629, 25667, 25705, 25721, 25734, 25760, 25824, 25824, 25874, 25898, 25951, 25990, 25996, 26058, 26113, 2
954, 29956, 29972, 30003, 30080, 30093, 30106, 30145, 30188, 30191, 30212, 30227, 30303, 30333, 30523, 30527, 30674, 30695, 30771, 30814, 30833, 30836, 30838, 30877, 30900, 30932, 30974, 31001, 310
        Time taken is 0
```


Ln 1, Col 1 100% Windows (CRLF) UTF-8

N=10000



```
C:\Users\HP\Documents\C++ CODE\programmingAss1.exe
--THIS PROGRAM USED DIFERENT SORTING ALGORITHMS(INSERTION,MERGE,QUICK,HEAP) TO
SORT RANDOMLY GENERATED INTEGERS AND THEN ANALYZE THEIR RUN TIME--

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 1
Enter a value that will determine the number of integers in the Array: 10000

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0.078
Time taken in Merge sort is 0
Time taken in Quick sort is 0
Time taken in Heap sort is 0.016

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 2
Enter a value that will determine the number of integers in the Array: 10000
Enter a positive Integer: 12

Generating inputs...
Sorting Array...

Time taken in Insertion sort is 0
Time taken in Merge sort is 0
Time taken in Quick sort is 0.125
Time taken in Heap sort is 0

Choose a program that will Generate the Integers
        [1]. Random
        [2]. Sorted
        [3]. End program

Enter selection: 3


Exiting program....

Open "README.txt" to see the output.

--------------------------------
Process exited after 19.35 seconds with return value 0
Press any key to continue . . .
```

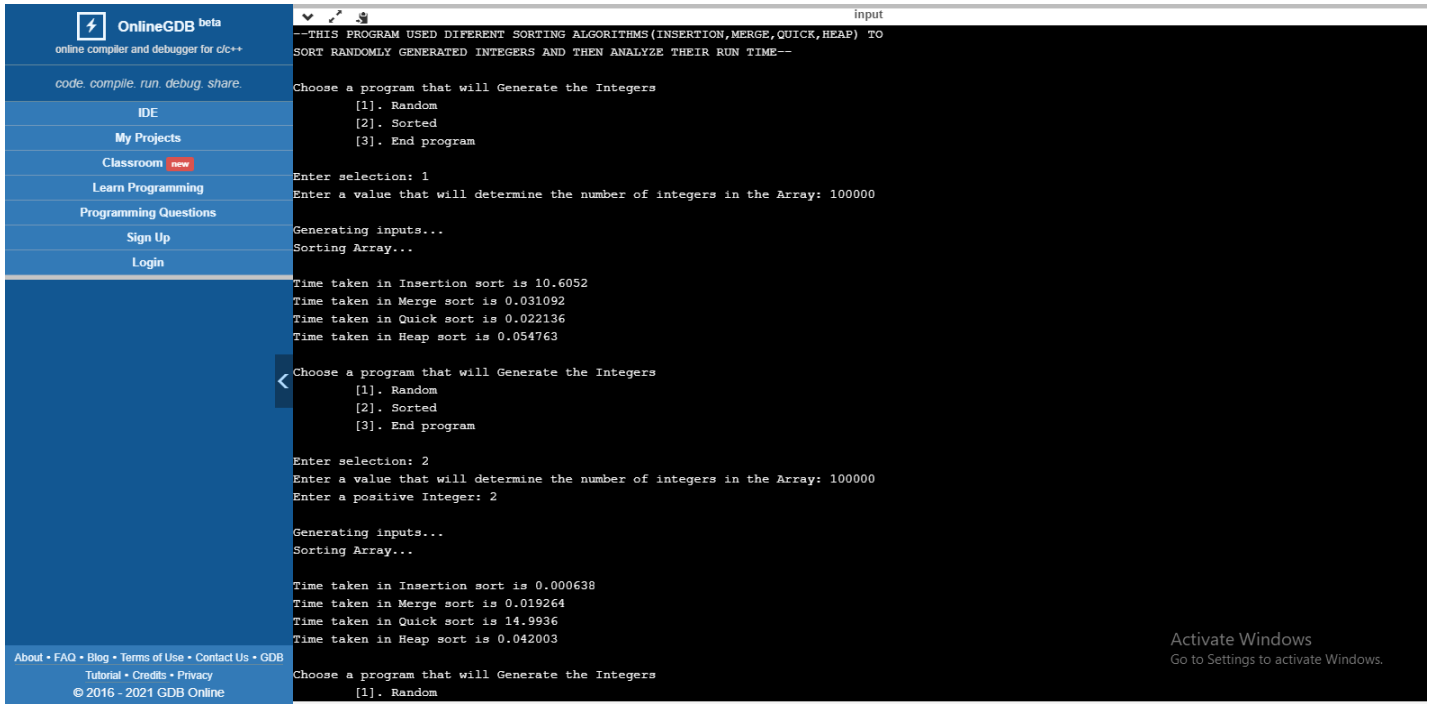

README - Notepad

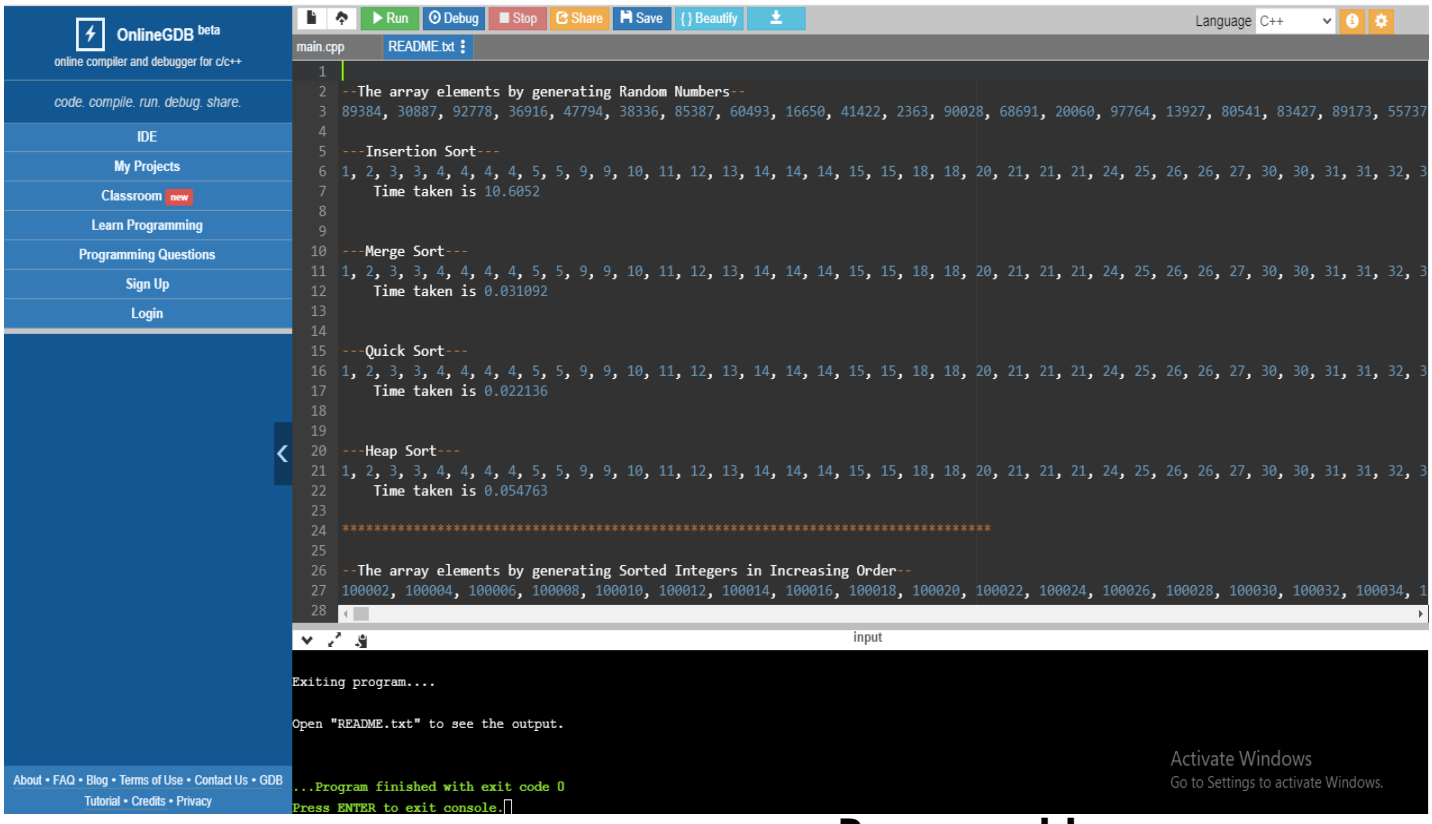
File Edit Format View Help

```
--The array elements by generating Random Numbers--
41, 18467, 6334, 26500, 19169, 15724, 11478, 29358, 26962, 24464, 5705, 28145, 23281, 16827, 9961, 491, 2995, 11942, 4827, 5436, 32391, 14604, 3902, 153, 292, 12382, 17421, 18716, 19718, 19895, 544
16413, 29168, 900, 32591, 18762, 1655, 17410, 6359, 27624, 20537, 21548, 6483, 27595, 4041, 3602, 24350, 10291, 30836, 9374, 11020, 4596, 24021, 27348, 23199, 19668, 24484, 8281, 4734, 53, 1999, 26
9, 7441, 9512, 30145, 18060, 21718, 3753, 16139, 12423, 16279, 25996, 16687, 12529, 22549, 17437, 19866, 12949, 193, 23195, 3297, 20416, 28286, 16105, 24488, 16282, 12455, 25734, 18114, 11701, 3131
4, 1018, 28464, 21119, 23152, 2800, 18087, 31060, 1926, 9010, 4757, 32170, 20315, 9576, 30227, 12043, 22758, 7164, 5109, 7882, 17086, 29565, 3487, 29577, 14474, 2625, 25627, 5629, 31928, 25423, 285
22701, 12193, 12734, 1637, 26534, 5556, 1993, 10176, 25705, 6962, 10548, 15881, 300, 14413, 16641, 19855, 24855, 13142, 11462, 27611, 30877, 20424, 32678, 1752, 18443, 28296, 12673, 10040, 9313, 87
4423, 3527, 11600, 26969, 14015, 5565, 28, 21543, 25347, 2088, 2943, 12637, 22409, 26463, 5049, 4681, 1588, 11342, 608, 32060, 21221, 1758, 29954, 20888, 14146, 690, 7949, 12843, 21430, 25620, 748,
3, 6698, 5589, 12722, 5938, 19037, 6410, 31461, 6234, 12508, 9961, 3959, 6493, 1515, 25269, 24937, 28869, 58, 14700, 13971, 26264, 15117, 16215, 24555, 7815, 18330, 3039, 30212, 29288, 28082, 1954,
146, 18392, 13037, 3925, 10647, 29458, 6602, 30807, 4098, 27830, 29292, 14600, 7278, 20799, 18352, 20448, 13882, 540, 28315, 4575, 8762, 9567, 22336, 18397, 31418, 19897, 25828, 13851, 26816, 24230
25547, 21577, 9789, 25605, 30195, 27594, 7950, 16343, 754, 2481, 1012, 11672, 23439, 13428, 5912, 9762, 5967, 24408, 4415, 1908, 17223, 11759, 26434, 5204, 29486, 4319, 958, 5945, 26806, 8166, 1870
, 6365, 32075, 7586, 1386, 7833, 8360, 13330, 26048, 8928, 29492, 12433, 23840, 6766, 1735, 19810, 11599, 11837, 21892, 31982, 7328, 29352, 11369, 31244, 21794, 6608, 9252, 11647, 17432, 9535, 7208
24081, 2995, 2678, 24676, 27753, 20899, 11784, 15565, 3093, 13608, 6172, 11243, 29929, 7514, 10168, 5055, 11191, 5973, 8922, 6748, 5651, 10986, 2144, 16446, 31577, 26517, 14629, 29916, 5874, 15791,
2, 2529, 9981, 2112, 13476, 4381, 8247, 26890, 16671, 8805, 32372, 30032, 3989, 9320, 23165, 15431, 9658, 11293, 17206, 26578, 16948, 2206, 27171, 18166, 3396, 16697, 31020, 23694, 15529, 14788, 30
508, 10080, 5340, 12076, 9058, 31493, 7740, 8546, 20474, 24773, 19097, 8880, 23335, 11072, 23400, 707, 22955, 20666, 4141, 23588, 12481, 17168, 28315, 19396, 16225, 1009, 22012, 18136, 11455, 18762
54, 3822, 32707, 24245, 1338, 144, 22290, 31339, 23154, 24604, 4623, 22225, 20078, 21724, 31981, 2330, 29733, 28223, 20594, 29130, 18846, 4987, 29445, 18805, 8616, 5750, 20489, 27338, 21963, 28135,
67, 32425, 26867, 13189, 10542, 13063, 21547, 30502, 29617, 9099, 25023, 17226, 31203, 30048, 21051, 1570, 8636, 9458, 25967, 8456, 9405, 11531, 29962, 26819, 7975, 10556, 7531, 31495, 8044, 30591,
841, 12686, 28888, 3917, 31717, 1892, 13698, 11267, 11749, 65, 10389, 6932, 25619, 16081, 2003, 29130, 13028, 18631, 14589, 23152, 18630, 19172, 19864, 6407, 12295, 25428, 29681, 18490, 26610, 2617
9866, 30953, 30308, 28223, 18684, 30792, 22667, 10048, 32469, 13930, 30811, 19814, 16090, 5427, 23743, 1604, 10599, 16474, 7195, 506, 5158, 17589, 9858, 27809, 17889, 11447, 40, 3818, 9364, 17975,
4, 2944, 30259, 821, 7058, 24643, 17668, 10677, 119, 29857, 23041, 8891, 25264, 1623, 6915, 8072, 17929, 841, 4715, 17615, 12536, 14957, 27759, 700, 13452, 5093, 24241, 11829, 12448, 5227, 11798, 1
3, 6266, 27651, 15275, 21528, 532, 10761, 5469, 32503, 32729, 17107, 27892, 2451, 17953, 28392, 22570, 13519, 19472, 23407, 22494, 9505, 28440, 20383, 14262, 21409, 28607, 18038, 4360, 5471, 11171,
981, 11503, 26569, 6816, 1883, 25367, 5385, 28402, 5230, 17157, 28681, 15567, 8310, 1866, 3687, 13171, 3477, 31245, 32764, 6238, 27671, 2047, 26115, 4592, 27311, 32657, 1405, 53, 7171, 20580, 22740
8, 26624, 22511, 7954, 4235, 1127, 14633, 28240, 1862, 15127, 27577, 23817, 2340, 10142, 15100, 28400, 6558, 21476, 3154, 10935, 16430, 5831, 21295, 3537, 15623, 17922, 2927, 28029, 30211, 16211, 1
3995, 29171, 7603, 28942, 18011, 7879, 31034, 14512, 24593, 530, 14161, 23234, 10258, 21926, 5906, 5191, 14443, 6188, 30227, 32719, 26484, 5224, 19093, 30435, 13224, 28094, 3582, 8867, 8627, 22005,
2, 7104, 1768, 29768, 26641, 16296, 11262, 10720, 21244, 27918, 9625, 21219, 17584, 18174, 10984, 10459, 13149, 17014, 5695, 12503, 5032, 13851, 1537, 26814, 8363, 4651, 1219, 24290, 25917, 21683,
435, 16693, 15072, 17916, 6716, 25328, 1511, 32125, 11838, 27296, 18939, 11544, 1056, 12826, 12902, 26758, 24588, 12021, 22586, 14108, 29777, 12628, 21979, 26947, 12549, 1280, 13245, 25128, 14062,
28290, 1898, 985, 21447, 1084, 592, 18725, 8996, 29855, 24828, 15338, 17342, 14428, 29795, 30742, 30037, 5506, 6767, 5053, 9793, 28505, 24146, 3609, 13924, 19854, 29465, 12239, 26206, 24209, 15492,
8709, 13371, 28642, 16339, 6336, 32621, 21501, 26034, 18129, 21358, 1436, 28469, 22897, 28504, 4798, 4202, 23105, 1238, 15032, 20065, 12516, 14481, 15579, 9023, 1727, 1771, 31313, 10382, 6140, 2481
6, 1936, 25584, 10790, 21617, 29182, 19066, 23116, 6388, 9425, 13101, 5610, 6450, 19982, 28621, 5193, 27038, 16754, 4593, 14895, 20280, 5638, 25167, 17420, 4345, 16883, 23865, 24696, 30833, 10390,
679, 20736, 19555, 25633, 1954, 930, 10252, 19904, 20279, 12488, 24870, 31754, 26072, 32176, 26701, 15415, 18086, 4231, 17480, 19975, 30260, 20514, 8938, 16657, 14990, 19438, 28981, 15447, 5242, 24
86, 21011, 10976, 7547, 12121, 23202, 7245, 6702, 28564, 31539, 2572, 2257, 2370, 21920, 31654, 19359, 11080, 7298, 23391, 23960, 30510, 29591, 25867, 22033, 1675, 12136, 30499, 17775, 25790, 14376
32019, 259, 22313, 2435, 1028, 4101, 10468, 22869, 456, 6731, 7578, 26074, 11395, 30843, 31274, 13316, 15024, 30511, 18585, 5722, 8971, 10057, 14095, 8174, 5038, 2183, 26658, 17295, 25722, 16983, 6
8759, 12755, 11417, 22844, 3554, 19443, 6824, 31999, 28422, 121, 15733, 31413, 19917, 15793, 21873, 12060, 31175, 9074, 5985, 32501, 31622, 13208, 27395, 11026, 26720, 26291, 31988, 10929, 9072, 30
6226, 29472, 27479, 19593, 32482, 3005, 18536, 15333, 11105, 27587, 30732, 15638, 31165, 1064, 14518, 8399, 5032, 17508, 13896, 13438, 23978, 22638, 8245, 4059, 24169, 744, 27362, 24769, 11049, 265
16, 26182, 19029, 25316, 3059, 13758, 4583, 22399, 1135, 8357, 20070, 15271, 22342, 17780, 13877, 30406, 28295, 17686, 2403, 8252, 15718, 23035, 7285, 28428, 16678, 25960, 101, 20949, 2364, 26385,
389, 20735, 22024, 18895, 28745, 6170, 11483, 28717, 28392, 28752, 11324, 19773, 27740, 18255, 4468, 22070, 16843, 10030, 15756, 10765, 3114, 2980, 1458, 12358, 8124, 6008, 25311, 6667, 21717, 1592
, 18616, 4961, 28717, 15907, 5920, 13245, 22166, 30121, 27782, 2782, 27006, 6491, 31870, 14615, 22195, 13143, 11763, 14771, 23592, 14935, 19217, 25412, 5881, 13467, 26721, 16952, 17322, 1828, 18230
2, 13485, 15822, 20003, 5161, 11896, 26141, 2150, 28432, 5872, 32361, 4770, 32313, 27165, 25645, 8120, 1810, 32128, 25970, 6621, 3596, 15004, 8514, 20799, 23210, 25269, 19552, 32235, 471, 16344, 11
9, 9843, 7004, 20101, 29384, 12920, 31987, 5136, 29052, 7582, 1528, 6366, 1472, 29193, 26958, 4179, 28983, 5987, 10823, 397, 11096, 16594, 8901, 17653, 14520, 30503, 13472, 20, 1932, 4953, 10824, 5
14255, 26742, 16748, 30103, 22815, 14047, 11073, 28965, 4749, 137, 32169, 14331, 7970, 26053, 8768, 28052, 13178, 11156, 24981, 12695, 17586, 13929, 731, 4278, 17859, 8752, 5089, 31958, 20238, 7054
4, 7953, 15760, 3586, 25419, 28735, 16616, 20113, 13335, 28592, 29018, 9265, 9370, 22336, 26525, 12746, 17644, 19512, 29482, 2815, 5105, 25726, 23201, 773, 25306, 3080, 31593, 23990, 4591, 24214, 3
11906, 32485, 24999, 28048, 23930, 573, 26658, 25816, 8884, 20373, 20031, 22604, 9783, 15791, 25422, 8010, 20987, 32364, 568, 5994, 579, 24023, 25709, 31657, 4141, 29285, 8477, 14023, 27272, 8828,
36, 11755, 15174, 18477, 1154, 24729, 10882, 15851, 19769, 11237, 28301, 15676, 13604, 27073, 24161, 21197, 8595, 2125, 5485, 15062, 29229, 27560, 7718, 29073, 8079, 8709, 9662, 10224, 11574, 7682,
```


Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

IN N=100,000 INPUTS WE USEDONLINE COMPILER





Prepared by:

Arvy Llave

Alliana Ermino

Ella Mae Poche

Bscs2-A