|  |  |
| --- | --- |
| **Application/ Program name:** | L1-3 |
| **Written by:** | Bailey Nichols |

|  |
| --- |
| **Purpose or problem definition:** |
| This program is made to demonstrate the differences in time for different sort methods. The program will make an array of integers and use three different sort methods to sort this array, printing all this information in the console. |
|  |
| **Program Procedures:** |
| * The program will fill the array with random values. * The program will sort the array using Selection sort, Bubble sort, and Insertion sort. * The program will output this information to the console. |
|  |
| **Algorithm/Processing/Conditions:** |
| **Inputs:** |
| * None |
| **Processes:** |
| * Make Array * Init array to ascending, descending and random values * Sort the array using Bubble sort, Selection sort and Insertion sort |
| **Outputs:** |
| * The program will output all the information in a format shown in notes. |
|  |
| **Notes & Restriction:** |
| Loading Inital Array :  Processing 10000 elemnts  with random values took  178 Microseconds  \*-----------------------------------------------------------------------------\*  \*-------------------------------------------\*  \* Testing - sorts with Random array values \*  \*-------------------------------------------\*  Bubble Sort array :  Before Sort: 1804289383 846930886 1681692777 1714636915 1957747793 424238335 719885386 1649760492 596516649 1189641421 1025202362 1350490027 783368690 1102520059 2044897763  After Sort: 100669 172621 479345 871000 881759 1121937 1323524 1648609 1769972 1839949 1887329 2044757 2054525 2236196 2237012  Processing 10000 Elements  with Bubble Sort took  224405 microseconds  \*-----------------------------------------------------------------------------\*  Insertion Sort array :  Before Sort: 89057537 1840048410 427773756 762677667 742585312 1447032062 1904054136 1665967229 1762956846 1222509902 465628956 817114752 71473853 1887484999 518096547  After Sort: 92804 272267 436756 976213 1233556 1588997 1906952 2283803 2664172 2841954 3099897 3180931 3193828 3551387 3610932  Processing 10000 Elements  with Insertion Sort took  65510 microseconds  \*-----------------------------------------------------------------------------\*  Selection Sort array :  Before Sort: 851227066 1907169184 1672626973 15224425 1327707705 1030165428 454990854 1575562599 908131722 980607030 373733361 905506050 1312242361 139236507 1143607569  After Sort: 56172 460933 598272 624561 698609 713388 810024 812720 835106 868235 1102558 1220554 1327314 1495908 1605802  Processing 10000 Elements  with Selection Sort took  89998 microseconds  \*-----------------------------------------------------------------------------\*  Loading Inital Array :  Processing 10000 elemnts  with ascending caues took  52 Microseconds  \*-----------------------------------------------------------------------------\*  \*-------------------------------------------\*  \* Testing - sorts with Ascending array values \*  \*-------------------------------------------\*  Bubble Sort array :  Before Sort: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  After Sort: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  Processing 10000 Elements  with Bubble Sort took  178550 microseconds  \*-----------------------------------------------------------------------------\*  Insertion Sort array :  Before Sort: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  After Sort: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  Processing 10000 Elements  with Insertion Sort took  50 microseconds  \*-----------------------------------------------------------------------------\*  Selection Sort array :  Before Sort: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  After Sort: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  Processing 10000 Elements  with Selection Sort took  89477 microseconds  \*-----------------------------------------------------------------------------\*  Loading Inital Array :  Processing 10000 elemnts  with descending values took  47 Microseconds  \*-----------------------------------------------------------------------------\*  \*-------------------------------------------\*  \* Testing - sorts with Descending array values \*  \*-------------------------------------------\*  Bubble Sort array :  Before Sort: 9999 9998 9997 9996 9995 9994 9993 9992 9991 9990 9989 9988 9987 9986 9985  After Sort: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  Processing 10000 Elements  with Bubble Sort took  263223 microseconds  \*-----------------------------------------------------------------------------\*  Insertion Sort array :  Before Sort: 9999 9998 9997 9996 9995 9994 9993 9992 9991 9990 9989 9988 9987 9986 9985  After Sort: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  Processing 10000 Elements  with Insertion Sort took  129677 microseconds  \*-----------------------------------------------------------------------------\*  Selection Sort array :  Before Sort: 9999 9998 9997 9996 9995 9994 9993 9992 9991 9990 9989 9988 9987 9986 9985  After Sort: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  Processing 10000 Elements  with Selection Sort took  103659 microseconds  \*-----------------------------------------------------------------------------\*  Code References:  thanks to wikipedia for code for selection sort  Anyon, Selection Sort, Wikipedia, <https://en.wikipedia.org/wiki/Selection_sort>  thanks to Wikipedia for sudocode  Anyon, Insertion sort, Wikipedia <https://en.wikipedia.org/wiki/Insertion_sort>  Thanks to Dr Knuth for the explanation of an a Bubble sort  Donald Knuth. The Art of Computer Programming,  Volume 3: Sorting and Searching, Second Edition.  Addison-Wesley, 1998. ISBN 0-201-89685-0. Pages 106–110 |
|  |
| **Comments:** |
|  |