**Assignment No.7**

|  |
| --- |
| **Title of Assignment:**  Write X86 program to sort the list of integers in ascending/descending order. Read the input from the text file and write the sorted data back to the same text file using bubble sort |
| **Algorithm :**   1. Read the filename from user. 2. Append filename with 0 (null character). 3. Open the file in read mode (flag=0) 4. Compare rax with 0 if less then print error and goto step 38. 5. Copy rax in fd\_s (file descriptor for file1) 6. Initialize rdi to arr (array to store 64 bit numbers in file) 7. Initialize n(number of elements in file) to 0 8. Initialize actl to 0 9. Intialize rsi to char\_buff 10. Read a character from file 11. Compare rax with 0 if 0(EOF) then goto step 20 12. Compare the read character with 10 i.e new line if same then goto step 15 13. Copy the character in buffer char\_buff 14. Increment the counter actl 15. Goto st ep 9 16. Call convert (ASCII -> Hex conversion method which converts ASCII data in char\_buff of length actl bytes into hex (in rbx register)) 17. Copy the hex number in rax to rdi(pointer to array) 18. Increment n 19. Goto step 8 20. Initialize CH to n-1 21. Initialize CL to n-1 22. Initialize rsi to arr 23. Compare element at rsi and rsi+8 24. If [rsi]<[rsi+8] goto step 26 25. Swap [rsi] and [rsi+8] 26. Add 8 to rsi 27. Dec cl 28. Compare cl with 0 if not equal goto step 23 29. Dec ch 30. Compare ch with 0 if not equal goto step 21 31. Open file in write mode (flag=1) 32. Compare rax with 0 if less then print error and goto step 38 33. Initialize rsi to Array arr. 34. Write the element pointed by rsi in file by converting it to ASCII. 35. Add 8 to rsi to point to next number in array. 36. Dec n 37. If n is not zero goto step 34 else to 38 38. End |
| **Testing:**  **Test Conditions:**  **Input:**  File containing Unordered 64 bit numbers.  **Output:**  Same file containing Ordered 64 bit numbers. |
| **FAQs:**   1. Explain System call for file opening, file closing, reading from file and writing to file, un 2. Explain Bubble sort. |
| **Conclusion:**  Successfully implemented the ALP for bubble sort by reading the input from the text file and writing the sorted data back to the same text file. |