Exercises - Problems Sheet # 4:
Search & Sort Algorithms, C File Processing, and Bitwise Operators

No. Of Questions: 14   **No. Of Pages: 2**

▶ **To be submitted during the Labs of week __ (To Be Announced Later).**

▶ **Students will lose 2 marks if this homework is not delivered on time or found out to be copied.**

▶ **The submitted solutions should be handwritten and NOT typed/printed.**

Answer the following:

1. Write a function that takes an array of integers and tells whether it is sorted or not.

2. Modify the bubble sort function to check at the end of each pass whether any swaps have been made. If none has been made, then the data must already be in the proper order, so the function should terminate. If swaps have been made, then at least one more pass is needed.

3. Rewrite the linear search algorithm recursively.

4. Create a text-based, menu-driven program that allows the user to choose whether to calculate the mean, the mode, or the median of an array. The program should then perform the appropriate calculation and display the result.

5. Write a function "replace" which takes a string as a parameter. It should replace all the spaces in that string by minus signs, and then counts and prints the number of spaces it replaced.

6. Write a program to compare the contents of two files specified by the user, displaying a message indicating whether the files are identical or different.

7. Create a program to extract (only) the alphabetic characters contained in a binary file, and dump them to a different file. The extracted characters must be those with an ASCII code between 32 and 127.

8. Write a program that writes the values of x and x^2 to a text file, for x starting from 1 to 10.

9. Write a program that asks the user to enter his/her information (Id, Name, Grade) and stores it in an existing file named student.txt. If the file does not exist, it will be created. After the user has supplied his/her information, the program will print on the screen all of the student's information available in the file.

10. Write a C program that copies a text file into another one after removing all of the empty lines.

11. Write a function that takes as an argument a pointer to a text file, and returns the number of characters in that file.

12. Write a program that displays a file's contents, 20 lines at a time. The program pauses after displaying 20 lines until the user presses either Q to quit or Return to display the next 20 lines.

13. Solve the following:
    (1) 10101 & 00000
    (2) 11111 & 11111
    (3) 10101 & 11111
    (4) 10101 | 00000
    (5) 11111 | 11111
    (6) 10101 | 11111
    (7) 10101 ^ 00000
    (8) 11111 ^ 11111
    (9) 00000 ^ 11111
    (10) 1 << 3
    (11) 100 << 1
    (12) 101 << 2
    (13) 100 >> 2
    (14) 1010 >> 1
    (15) 1010 >> 3
    (16) ~10101
    (17) ~11111
    (18) ~00000

14. Construct bitwise functions, and test them using the following inputs (10101, 11111, 11100)
    a. Get the rightmost bit of any input.
    b. Get the 3 rightmost bits of any input.
    c. Get the leftmost bit of any input.
    d. Get the 3 leftmost bits of any input.
    e. Remove the rightmost bit of any input.
    f. Remove the rightmost 3 bits of any input.
    g. Set the leftmost bit of any input to one.
    h. Set the 3 leftmost bits of any input to one.
    i. Set the 4th, 5th, 6th, and 7th rightmost bits of any input to one.
    j. Remove the 4th, 5th, 6th, and 7th rightmost bits of any input.
    k. Flip the 4th, 5th, 6th, and 7th rightmost bits of any input.
    l. Remove the first bit of any input, and add it to the right.

With our best wishes;