**Problem Set: 50 Problems for Practice**

**Note:** Implement algorithm/function with least time complexity

1. A sorted array has been rotated; you have to find minimum number of this array.

Example: Initial array index --- 1 2 3 4 5 6 7 after rotation

4 5 6 7 1 2 3

1. You have given time, say twenty five minutes past four. Calculate angle between the hour and minute hands in a clock.
2. Implement the logic of finding all permutation of a given string. All characters are unique in given string.
3. Write a C program to find the longest word in a given string. Also print the length of the word.
4. An integer array A [] of size ***n*** is given; find the three greatest numbers in this array.
5. Implement an algorithm to print all valid (e.g., properly opened and closed) combi­nations of n-pairs of parentheses.

EXAMPLE:

input: 3 (e.g., 3 pairs of parentheses)

output: () ()(), ()(()), (())(), ((()))

1. Write a method that returns all subsets of a set.
2. A sorted array of integers and a number m is given; find the closet number to m.
3. Equilibrium Point in an integer array (sum of left side elements should be equal to sum of right hand side elements).
4. Extend above for multiplication (multiplication of left side elements equal to multiplication of right side elements).
5. Implement **memcpy (\*src,\*target)** function.
6. Two strings s1 and s2 are given; find a minimal length substring in s1 which does not contain s2 as a substring. (Do include all edge cases).
7. Suppose numbers are randomly generated and stored into an (expanding) array. How would you keep track of the median?
8. Write a method to decide if two strings are anagrams or not.
9. If you have a 2 GB file with one string per line, which sorting algorithm would you use to sort the file and why?
10. Write a method which finds the maximum of two numbers. You should not use if-else or any other comparison operator.
11. Given an integer between 0 and 999,999, print an English phrase that describes the integer (e.g., “One Thousand, Two Hundred and Thirty Four”).
12. You are given an array of integers (both positive and negative).Find the continuous sequence with the largest sum and return the sum.
13. Design an algorithm to find all pairs of integers within an array which sum to a speci­fied value.
14. Write a function that adds two numbers. You should not use + or any arithmetic op­erators.
15. Assuming that bit numbering starts from 1 from LSB to MSB. Write a C program to set a particular bit in a given number.
16. You are given two 32-bit numbers, ***a*** and ***b***, and two bit positions, i and j. Write a method to set all bits between i and j in ***b*** equal to ***a*** (e.g., ***a***becomes a substring of ***b*** located at i and starting at j).

EXAMPLE:

Input: b = 10000000000, a = 10101, i = 2, j = 6

Output: b= 10001010100

1. Write a function to determine the number of bits required to convert integer A to integer B.

Input: 31, 14

Output: 2

1. Given an integer, print the next smallest and next largest number that has the same number of 1 bit in their binary representation.
2. Write a program to swap odd and even bits in an integer with as few instructions as possible (e.g., bit 0 and bit 1 are swapped, bit 2 and bit 3 are swapped, etc).
3. To check whether the input number is even or odd without using modulus operator.
4. Write the logic for Ransom Note Problem (Hint: use hashing).
5. Describe how you could use a single array to implement three stacks.
6. Implement Queue using two stacks.

**Based on Linked List:**

1. Reverse singly linked list
2. Reverse doubly linked list
3. Write a C function to detect a loop in a singly linked list.
4. To find the middle of a singly linked list. C function to return the middle of a singly linked list
5. To return address of the **nth** node from the end of a singly linked list.
6. How to read a singly linked list backwards?
7. To remove duplicates from a sorted linked list.
8. To sort the singly linked list
9. Stack using linked list
10. Queue using linked list
11. Implement an algorithm to delete a node in the middle of a single linked list, given only access to that node.
12. Write code to remove duplicates from an unsorted linked list.
13. Swap all nodes of a linked list with their next nodes.

**Based On Binary tree:**

1. To count total number of nodes
2. To count total number of leaf nodes
3. To count total number of internal nodes
4. To calculate height of the binary tree
5. To delete a binary tree.
6. To determine if two binary trees are identical or not
7. To create a mirror copy of a tree (left nodes become right and right nodes

become left)

1. To check if a given binary tree is a binary search tree or not?