Assignment 2

Indian Institute of Technology, Roorkee
Programming Lab (CS-504)

Autumn Semester

Problem Statement 1

An array $A[0, ...2n]$ is **wiggly** if $A[0] \leq A[1] \geq A[2] \leq A[3] \geq ... \leq A[2n-1] \geq A[2n]$. Given an unsorted array B[0, ...2n] of real numbers, write an efficient, $O(n)$ time complexity, C++/JAVA program using arrays that outputs a permutation $A[0, ...2n]$ of B such that A is a **wiggly** array.

Problem Statement 2

Given an array of size n, generate and print all possible combinations of r elements in array. For example, if input array is $1, 2, 3, 4$ and r is 2, then output should be {1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4} and {3, 4}.

Problem Statement 3

An array is called ZIG-ZAG if it satisfies all of the following rules:

- It has odd number of elements
- All even indexed variables are exactly 1 larger than all odd-in-dexed variables (index starts at 0)

An operation MOVE( ) is defined as:

MOVE(A[ ], i, j, k):
Decrement A[i] by k
Increment A[j] by k

Given an array A, write a C++/JAVA program to determine whether it can be made ZIG-ZAG using only MOVE( ) operations. If it can, determine the minimum value of the sum of parameters k in all the MOVE( ) operations.

Problem Statement 4

Write an efficient algorithm to count the number of Centered Heptagonal Primes in a given array. A centered heptagonal prime is a centered heptagonal number that is prime. The first few centered heptagonal primes are 43, 71, 197, 463, 547, 953, 1471, 1933, 2647, 2843, 3697.

The centered heptagonal number for 'n', where 'n'= {0, 1, 2, ..}. is given by the formula $(7n^2-7n+2)/2$

Problem Statement 5

Merge two sorted linked lists and return it as a sorted list. The list should be made by splicing together the nodes of the first two lists.

Examples:

1. • Input: L1 = [1,2,4], L2 = [1,3,4]
   • Output: [1,1,2,3,4,4]
2. • Input: L1 = [], L2 = [0]
   • Output: []

Constraints:

- Both L1 and L2 are sorted in non-decreasing order.

Problem Statement 6

You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security systems connected and it will automatically contact the police if two adjacent houses were broken into on the same night.

Given an integer array 'nums' representing the amount of money of each house, return the maximum amount of money you can rob tonight without alerting the police.

Examples:

1. • Input: nums = [1,2,3,1]
   • Output: 4
   • Explanation: Rob house 1 (money = 1) and then rob house 3 (money = 3). Total amount you can rob = 1 + 3 = 4.
2. • Input: nums = [2,7,9,3,1]
   • Output: 12
   • Explanation: Rob house 1 (money = 2), rob house 3 (money = 9) and rob house 5 (money = 1). Total amount you can rob = 2 + 9 + 1 = 12.