## Programming: Sheet 2

1. Given two positive numbers num1 and num2, write a program to check if num2 is power of num1 or not.

2. Given a first element a and common difference d, write a program to find sum of an arithmetic progression AP having n number of elements.

**Array:**

1. Given an array, write a program to find largest, smallest element, second largest element and second smallest element present in this array. Example: Consider array[6] = {4, 54,61, 21,74,8}, 1st largest = 74, 2nd largest = 61, 1st smallest = 4 and 2nd smallest = 8.

2. Given an array and a value k, write a program to left and right rotate the array by k elements. Example: Consider array[5] = {1,2,3,4,5} and k=2. After left rotating array by 2 elements, array becomes {3,4,5,1,2}, after right rotating array by 2 elements, array becomes {4,5,1,2,3}.

3. Given marks of students in the form of an array, write a program to find average marks gained by students and also calculate median. Example: Consider array[5] = {41,78,65,88,80}, average marks = 70.4 and median = 78.

4. Given a sorted array and a key value, write a program to find number of array elements >= key and number of array elements < key. Example: Consider array[5] = {78,451,64,105,251} and key = 200. Therefore, no. of elements >= key : 2 and no. of elements < key : 3.

5. Given an array, write a program to find minimum number that needs to be added to this array to make the sum of array even. Note: minimum number should be greater than 0 and implement it by passing array to a function.

6. Given an array of even size, write a program to find minimum number that need to be added to this array to make it balanced. An array is said to be balanced if sum of values of left half part of array is equal to sum of values of right half part of array. Example: Consider array[6] = {12,43,52,17,31,14}. Left half sum = 12+43+52 = 107,right half sum = 17+31+14 = 62. Therefore, minimum number required to make array balance = 107-62 = 45.

7. Given an array (array may have duplicate values) and a key value which is present in array. Write a program to find the start index and end index, where start index will represent the first index of the array where key is present and end index will represent the last index of the array where the key is present. Example: array[8] = {41,45,24,41,41,45,68,21} and key = 41. Therefore, start index of (41) = 1, end index of (41) = 5.

**Matrices:**

Given two 2-dimensional matrices mat1[n][m] and mat2[a][b] where n and a is number of rows and m and b is number of columns. Write a program to perform following operations on both matrices:

a) matrix addition b) matrix subtraction c) multiplication d)matrix transpose
e) matrix inversion f) sum of each row g) sum of each column
h) sum of main diagonal elements i) sum of opposite diagonal elements

**Trees:**

1. Write a program to implement binary trees having following basic operations:
a) CreateTree() - creates a root node
b) InsertNode() - insert a node in tree
c) DeleteNode() - delete a node from tree

d) FindHeight() - find the height of tree
e) FindSize() - find number of nodes in tree
f) Search() - search whether given specific element present in tree or not.

2. Write a program to implement following types of traversing in the tree:
a) inorder() - (1) traverse left subtree, (2) traverse root, (3) traverse right subtree
b) postorder() - (1) traverse root, (2) traverse left subtree, (3) traverse right subtree
c) preorder() - (1) traverse left subtree, (2) traverse right subtree, (3) traverse root

3. Write a program to implement binary search tree having following BST operations:
a) Create() - creates a root node.
b) Insert() - insert a node in tree.
c) Delete() - delete a node from tree.
d) FindHeight() - return height of tree.
e) FindDepth() - return depth of tree.
f) Search() - search whether given key is present in tree or not.

4. Given a binary tree, write a program to check whether this tree is heap or not. Tree should satisfy following heap properties:
a) Tree should be complete binary tree
b)Every nodes value should be greater than or equal to its child node (incase of max heap)

5. Given a complete binary tree, write a program to find Kth largest element in it. (Hint: Max heap)

**Graphs:**
1. Given a (directed/undirected) graph, design an algorithm and implement it using a program to find if a path exists between two given vertices or not. (Hint: use DFS)

2. Given a graph, design an algorithm and implement it using a program to find if a graph is bipartite or not. (Hint: use BFS)

3. Given a directed graph, design an algorithm and implement it using a program to find whether cycle exists in the graph or not.

4. Assume that a project of road construction to connect some cities is given to your friend. Map of these cities and roads which will connect them (after construction) is provided to him in the form of a graph. Certain amount of rupees is associated with construction of each road. Your friend has to calculate the minimum budget required for this project. The budget should be designed in such a way that the cost of connecting the cities should be minimum and number of roads required to connect all the cities should be minimum (if there are N cities then only N-1 roads need to be constructed). Now, you have to help your friend by designing an algorithm which will find minimum cost required to connect these cities. Implement this problem using:
a) Prim's algorithm b) Kruskal's algorithm

5. Given a graph G and a source vertex u, write a program to find shortest paths from source to:
a) a destination vertex v. b) all other vertices of G.

6. Given a directed graph with two vertices ( source and destination). Write a program to find the weight of the shortest path from source to destination with exactly k edges on the path.

**Searching:**

1. Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Binary search)

2. Given a sorted array of positive integers and a key value, write a program to find:
a) two indices iand j, such that arr[i] + arr[j] = key.
b) three indices i, j, k such that arr[i] + arr[j] = arr[k].

3. Given an already sorted array of positive integers, write a program to find whether a given key element is present in the sorted array or not. For an array arr[n], search at the indexes arr[0], arr[2], arr[4],.....,arr[2k] and so on. Once the interval (arr[2k] < key < arr[ 2k+1] ) is found, perform a linear search operation from the index 2k to find the element key. (Jump Search)

**Sorting:**

1. Given a binary array (array containing 0 and 1 only), write a program to sort this array in increasing order. Example: Consider array[6] = {1,0,1,1,0,1}, after sorting array becomes {0,0,1,1,1,1}.

2. Write a program to sort a given array in such a way that left part of array contains odd numbers while right part contains even numbers. These sub arrays of odd and even numbers need not to be in sorted order. Perform this operation with only one scan of array. Example: Consider arr[6] = {14,1,6,2,3,11}, after sorting array = {1,3,11,14,6,2}.

3. Given an unsorted array of integers, write a program to sort the array using:
a) insertion sort b) selection sort c) bubble sort

4.Given an unsorted array of alphabets containing duplicate element, write a program to find which alphabet has maximum number of occurrences and print it.