DATA STRUCTURES

BATCH – B
[THURSDAY MARCH 9, 2017: 2:00 PM – 5:00 PM]

ASSIGNMENTS – 8 CODE: assign08

INSTRUCTIONS: [Total Marks: 30]

iv) Read all assignments and each problem has to be answered in the same c file.
v) Create a .c file following the file name convention: abc-assign08.c
Where abc is your roll number and assign08 is the assignment code
vi) Strictly follow the file name convention and do not use scanf()

PROBLEMS: (on Binary Search Trees)

3) **[Marks: 5 marks]**

Define a node - BSTREE - of a Binary Search Tree (BST) with following fields:

```
movieID: <int> - a Unique Movie Identifier - [100, 2000]
mname: <char> - it can hold a maximum of 16 characters
likes: <int> - size [50, 10000] – likes of the people
year: <int> - year of release of the movie
rating: <float> - [1, 10] – rating of the movie
budget: <float> - [10.0, 20.00] – making cost of the movie(in crores)
revenue: <float> - total box office collection made by the movie in
                   the range - [1.0, 1000.0] (amount in crores)
time: <int> – [20.0, 180.0] – running time in minutes
```

The values of these fields could be generated using a random number generator in the specified range. Assume a list of specific names for the field "mname"

4) **[Marks: 25 marks]**
Using above data structure and function prototypes given below, write your code for following tasks:

g) **[Marks: 5 marks]**
Assume that we are going to generate the Binary Search Tree with the details of n (=20) movies. Create a binary search tree with n nodes.

```
BSTREE *genMovieDataset (BSTREE *bstree, int n);
```

This function should insert an element into the binary search tree in such a way that the nodes insertion is based on movieID.

h) **[Marks: 3 marks]**
Write a function to print the details of movies by year one per line:

```
void printMovieDetailsByYear(BSTREE *bstree, int year);
```

i) **[Marks: 3 marks]**

Write a function to search and print the details of the movie by the given moviename

```
void FindMovieByName(BSTREE *bstree, char *moviename);
```

j) **[Marks: 3 marks]**

Write a function to modify the rating of all movies with specific rating ranging from oldrating in [4.15, 5.35] to newrating in [6.25, 8.27]

```
void ModifyRating(BSTREE *bstree,  float oldrating, float newrating);
```

k) **[Marks: 5 marks]**

Write a function to convert the existing Binary Search Tree into two subtrees of near equal heights based on the running time. Note that each subtree should be a complete binary tree.

```
void SplitBST(BSTREE *bstree, BSTREE *one, BSTREE *two);
```

l) **[Marks: 6 marks]**

Write a function to delete all movies that has a specific profit within the given range [min, max]. Here profit has to be calculated by profit = (revenue – budget). You may round off the min and max values to the nearest integer.

```
BSTREE *deleteMovies(BSTREE *bstree, float min, float max);
```

At the end of the function call print the details of the remaining nodes in the binary search tree.