DATA STRUCTURES

BATCH – B
[THURSDAY, MARCH 02, 2017: 2:00 PM – 5:00 PM]

ASSIGNMENTS – 7 CODE: assign07

INSTRUCTIONS: [Total Marks: 20]

i) Read all assignments and each problem has to be answered in the same c file.
ii) Create a .c file following the file name convention: abc-assign07.c
Where abc is your roll number and assign07 is the assignment code
iii) Strictly follow the file name convention and do not use scanf()

PROBLEMS: (Remember that you are working ONLY with BINARY TREEs)

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

Define a node - BTNODE - of a binary tree with the following fields:

```
proID: <int> - [1000, 4999]
rank: <int> - [1, 5000]
level: <int> - it can range from [0, h] in every level where h is the
                height of the binary tree
mrp: <float> - [99.0, 899.00] – Maximum Retail Price (MRP)
discount: <float> - [2.0, 9.8] – deduct from cost
charges:  <float> - [60.0, 200.0] – Add to (cost – discount)
```

The values of these fields could be generated using a random number generator with specified range. Remember that actual cost = mrp – discount + charges. Do not have a separate field for actual cost (in turn, compute it during run time).

2) **[Marks: 17 marks]**
Using above data structure and the function prototypes given below,
Write your code for following tasks:

a) **[Marks: 5 marks]**
Assume that you have n items in your database.
Create a binary tree with these n nodes.

```
BTNODE *genBinaryTree (BTNODE *btnode, int n);
```

This function should internally insert an element with a unique proID into the binary tree in such a way that the resulting binary tree is complete. Remember that NO TWO nodes can have the same proID.

b) **[Marks: 2 marks]**
Write a function to print the details of each item (space separated in a row)

```
void printElements(BTNODE*btnode);
```

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

Write a function to search based on the discount amount of each item. To do this, you have to find the discount of each item and check whether this discount falls within the specified range with `low` (minimum discount of the `mrp`) and `high` (maximum discount of the `mrp`)

```
void RangeSearchByDiscount(BTNODE *item, float low, float high);
```

In the `main()` function, generate the values of `low` and `high` in the range `[10.0, 99.0]` in such a way that `low` is minimum amount of the discount and `high` is maximum is the maximum amount of the discount.

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

Write a function to search an item with the highest cost. Use this function to find and print details of two nodes with the first highest and the second highest cost.

```
void SearchMaximumCost(BTNODE *btnode);
```

e) **[Marks: 4 marks]**

Write a function to delete all elements with the specified `flag`.

`flag` = `0`, delete all nodes whose proID is a `EVEN` number
`flag` = `1`, delete all nodes whose proID is a `ODD` number
`flag` = `2`, delete all nodes whose proID is a `PRIME` number
`flag` = `3`, delete all nodes that are `LEAF` nodes in the binary tree

```
BTNODE *deleteElements(BTNODE*btnode, int flag);
```