ACC311

Question 1: Define Reference Variable, Dangling Reference & Const

Answer:
In the C++ programming language, a reference is a simple reference datatype that is less powerful but safer than the pointer type inherited from C. The name C++ reference may cause confusion, as in computer science a reference is a general concept datatype, with pointers andC++ references being specific reference datatype implementations Dangling Reference & Const Dangling pointers and wild pointers in computer encoding are pointers that do not point to a valid object of the suitable type. These are special cases of violations of memory safety

Question 2: Describe any two uses of priority queues?

Answer:
Priority queue being used at many places especially in the operating systems. In operating systems, we have queue of different processes. If some process comes with higher priority, it will be processed first. Here we have seen a variation of queue. We will use the priority queue in the simulation. The events will be inserted in the queue and the event going to occur first in future, will be popped.
(page 101)

Question 3: Determine what the following recursive "mystery" function computes when given a pointer to the root node of a binary tree.

```
struct bt_s { int key; struct bt_s *left, *right; } bt_t;
int MFunc (bt_t *T) {
int N1, N2;
if (T == NULL) return -1;
N1 = MFunc(T->left);
N2 = MFunc(T->right);
return (N1 > N2 N1 : N2) + 1;
}
```

Question 4: Draw AVL Tree by following digits 78, 21, 25, 28, 38, 36, 75 and also perform necessary rotation, while showing all the intermediate trees being created in the process. In each stage, the AVL transformation should be conducted at a discrepancy that is farthest from the root.

Question 5: Following is the while loop used in level-order traversal:

```
while(!q.empty()) {
 treeNode = q.dequeue();
 cout << *(treeNode->getInfo()) << " ";
 if(treeNode->getLeft() != NULL )
 q.enque( treeNode->getLeft());
 if(treeNode->getRight() != NULL )
}
```

Question 6: Here is a small binary tree:

```
14
/
2 11
/ /
1 3 10 30
/ /
```