Q#1. (a) Sort the given list using Insertion Sort. (10)
65, 70, 56, 51, 54, 66

Algorithm:

Element =6

n-1=6-1=5

step #1

65,70,56,51,54,66

Step#2

65,70,56,51,54,66
65,56,70,51,54,66
56,65,70,51,54,66

Step#3

56,65,70,51,54,66
56,65,51,70,54,66
56,51,65,70,54,66
51,56,65,70,54,66

Step#4

51,56,65,70,54,66
51,56,65,54,70,66
51,56,54,65,70,66
51,54,56,65,70,66

Step#5

51,54,56,65,70,66
51,54,56,65,66,70

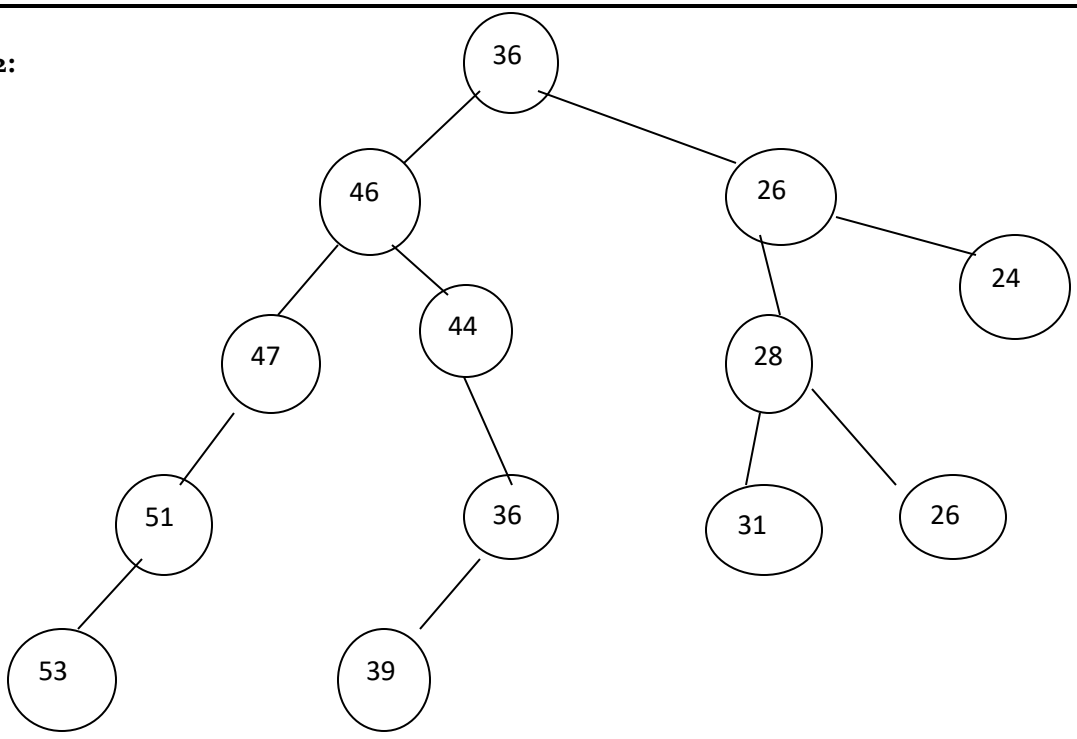
The list is now sorted…

```
// C++ program for insertion sort
#include <bits/stdc++.h>
using namespace std;

/* Function to sort an array using insertion sort*/
void insertionSort(int arr[], int n)
{
   int i, key, j;
   for (i = 1; i < n; i++)
   {
      key = arr[i];
      j = i - 1;

      /* Move elements of arr[0..i-1], that are
      greater than key, to one position ahead
```

```
        of their current position */
        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

// A utility function to print an array of size n
void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
}

int main()
{
    int arr[] = {65,70,56,51,54,66 };
    int n = sizeof(arr) / sizeof(arr[0]);

    insertionSort(arr, n);
    printArray(arr, n);

    return 0;
}
```

Output:
51, 54, 56, 65, 66, 70

Q no 2:



Verification:
53,51,47,46,44,39,36,36,31,28,26,26,24

---

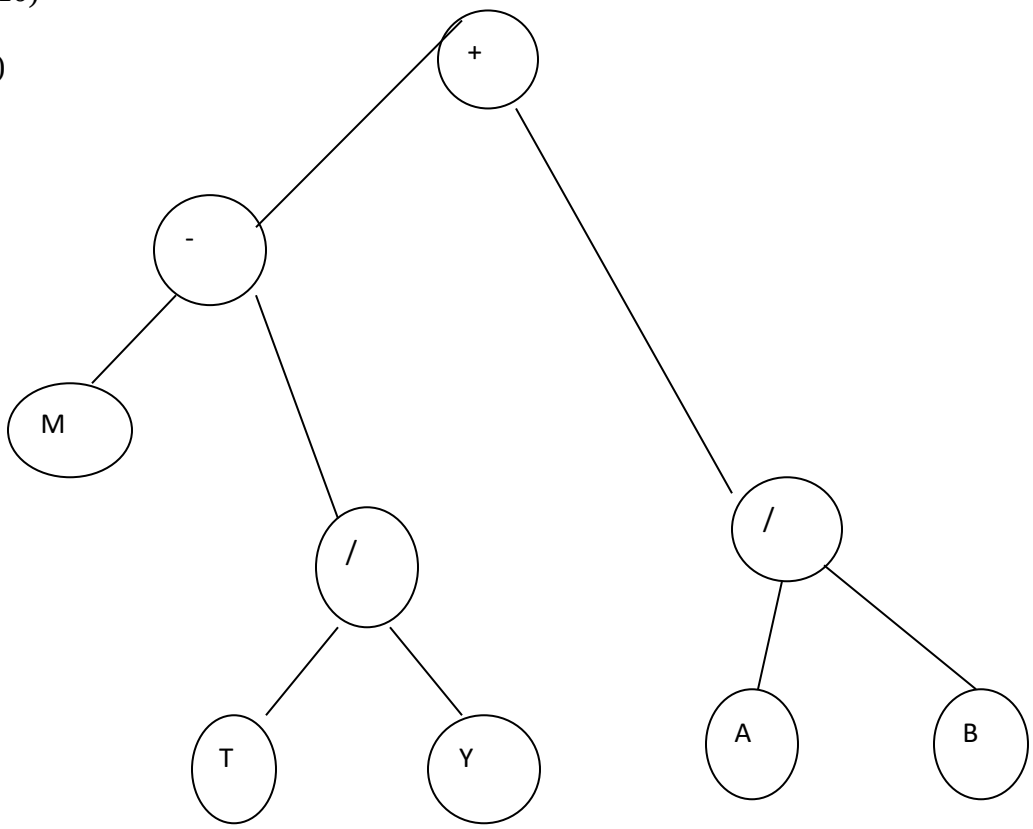
i. M - T / Y + (A / B) (10)

ii. A * B + R ^ K (10)

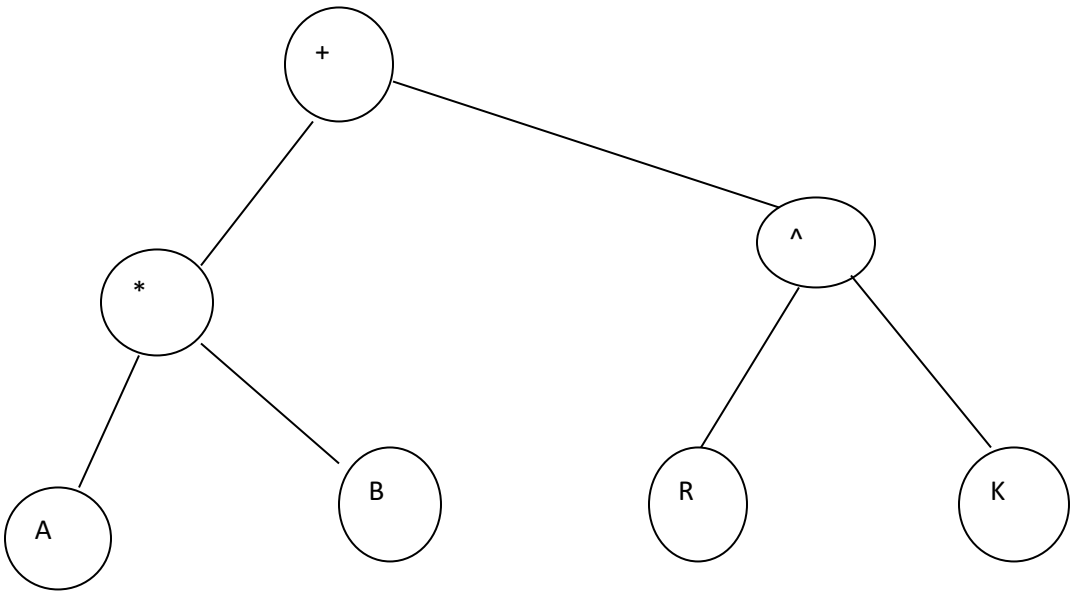
Ans:

M - T / Y + (A / B)

Q no3 part (a)



Q no 3 part(b)



Q no 4:

In order traversal:

M,-,t,/,y,+,A,/B

PRE-ORDER:

+,-,M,/,T,Y,/,A,B

POST ORDER:

M,T,Y,/,-,A,B,/,+