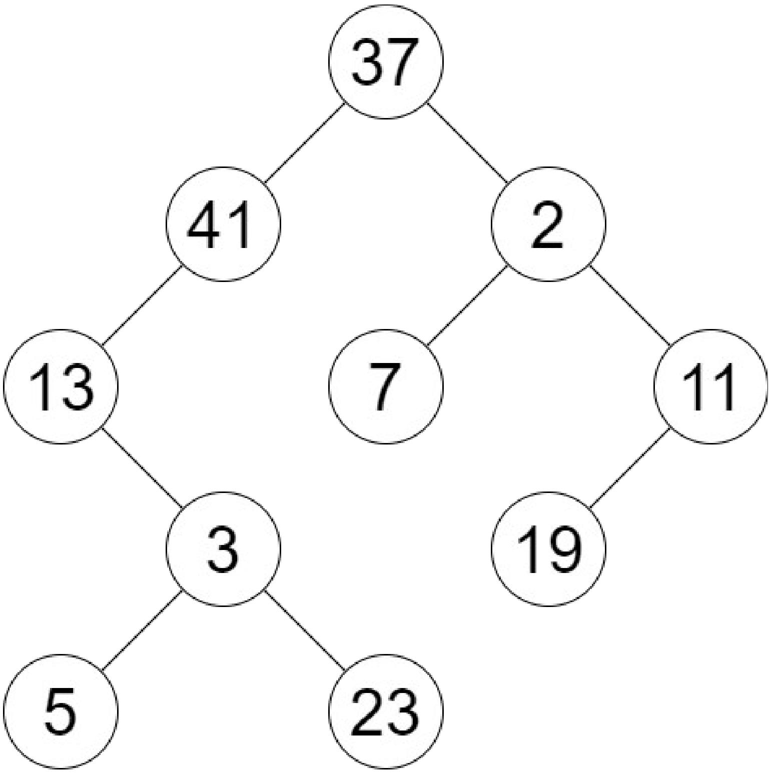

```
#include  <iostream>
using  namespace  std  ;

struct  Cell
{
    int  data   ;
    int  left   ;
    int  right  ;
};

typedef  struct Cell * ATree  ;

int  tree_root(ATree atree, int nbre)
{
    if(nbre <= 0)
        return  -1  ;

    bool*  root_array  =  new  bool[nbre]   ;

    for(int i = 0; i < nbre ; i++)
        root_array[i]  =  true  ;

    for(int i = 0; i < nbre ; i++)
    {
        if(atree[i].left != -1)
            root_array[atree[i].left]  =  false  ;

        if(atree[i].right != -1)
            root_array[atree[i].right]  =  false  ;
    }

    for(int i = 0; i < nbre ; i++)
        if(root_array[i])
            return  i  ;

    return  -1  ;
}

void  tree_leaves(ATree atree, int nbre)
{
    for(int i = 0; i < nbre ; i++)
        if((atree[i].left == -1)&&(atree[i].right == -1))
            cout  <<  "leaf : "  <<  atree[i].data  <<  endl  ;

    return  ;
}

int  main()
{
    Cell  atree[]  =  { {23, -1, -1}, {2, 4, 5}, {3, 3, 0}, {5, -1, -1}, {7, - 1, -1},
                      {11, 9, -1}, {13, -1, 2}, {37, 8, 1}, {41, 6, -1}, {19, -1, - 1} } ;

    cout  <<  "the root index is : "  <<  tree_root(atree, 10)  <<  endl  ;

    tree_leaves(atree, 10)  ;

    return  0  ;
}
```

Postfix traversal : 5, 23, 3, 13, 41, 7, 19, 11, 2, 37
Prefix traversal :   37, 41, 13, 3, 5, 23, 2, 7, 11, 19
infix traversal :   13, 5, 3, 23, 41, 37, 7, 2, 19, 11