"""

Name :- Mahesh Gaikwad

Roll :- SA21

A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method.

"""

```
#include <iostream>
#include <string.h> using
namespace std;

struct node {
   string label;     int
ch_count;     struct node
*child[10];
} *root;

// Class Declaration
class GT { public:
   void create_tree();
void display(node *r1);
GT()
```

```
    {
        root = NULL;
    }
};

void GT::create_tree()
{
    int tbooks, tchapters, i, j, k;
root = new node;

    // Book name     cout << "Enter
name of book : ";     cin.get();
getline(cin, root->label);

    // No. of chapters
    cout << "Enter number of chapters in book : ";
cin >> tchapters;     root->ch_count = tchapters;

    for (i = 0; i < tchapters; i++)
    {
        root->child[i] = new node;

        // chapter names
        cout << "Enter the name of Chapter " << i + 1 << " : ";
cin.get();
        getline(cin, root->child[i]->label);
```

```
        // no. of sections in chapter
        cout << "Enter number of sections in Chapter : " << root->child[i]->label << " : ";
cin >> root->child[i]->ch_count;         for (j = 0; j < root->child[i]->ch_count; j++)
        {
           root->child[i]->child[j] = new node;

           //  name of section
           cout << "Enter Name of Section " << j + 1 << " : ";
cin.get();
           getline(cin, root->child[i]->child[j]->label);
        }
    }
}

void GT::display(node *r1)
{
    int i, j, k, tchapters;
if (r1 != NULL)
    {
        cout << "\n-----Book Hierarchy---";
cout << "\nBook title : " << r1->label;
tchapters = r1->ch_count;         for (i = 0; i
< tchapters; i++)
        {

           cout << "\n\tChapter " << i + 1;
cout << " : " << r1->child[i]->label;
```

```
        if (r1->child[i]->ch_count != 0)
cout << "\n\t\tSections : ";

        for (j = 0; j < r1->child[i]->ch_count; j++)
        {               cout
<< "\n\t\t\t"
                << r1->child[i]->child[j]->label;
        }
      }
   }
   cout << endl;
}

int  main()  {
int     choice;
GT          gt;
while (1)
   {
      cout << "-----------------" << endl;
cout << "Book Tree Creation" << endl;
cout << "-----------------" << endl;        cout
<< "1.Create" << endl;        cout <<
"2.Display" << endl;        cout << "3.Quit"
<< endl;

      cout << "Enter your choice : ";
cin >> choice;        switch (choice)
```

```
        {
case 1:
            gt.create_tree();
case 2:
gt.display(root);
break;          case 3:
            cout << "Thanks for using this program!!!";
exit(1);          default:
            cout << "Wrong choice!!!" << endl;
        }     }
return 0;
}
```

OUTPUT:

```
---------- Book
Tree Creation
----------
1.Create
2.Display
3.Quit
Enter your choice : 1
Enter name of book : maths
Enter number of chapters in book : 2
Enter the name of Chapter 1 :  statistics
Enter number of sections in Chapter :  statistics : 2
Enter Name of Section 1 : mean
Enter Name of Section 2 : median
```

Enter the name of Chapter 2 : discrete mathematics

Enter number of sections in Chapter : discrete mathematics : 1

Enter Name of Section 1 : binary tree

-----Book Hierarchy---

Book title : maths

  Chapter 1 : statistics

Sections :

mean edian

  Chapter 2 : discrete mathematics

Sections :

  binary tree