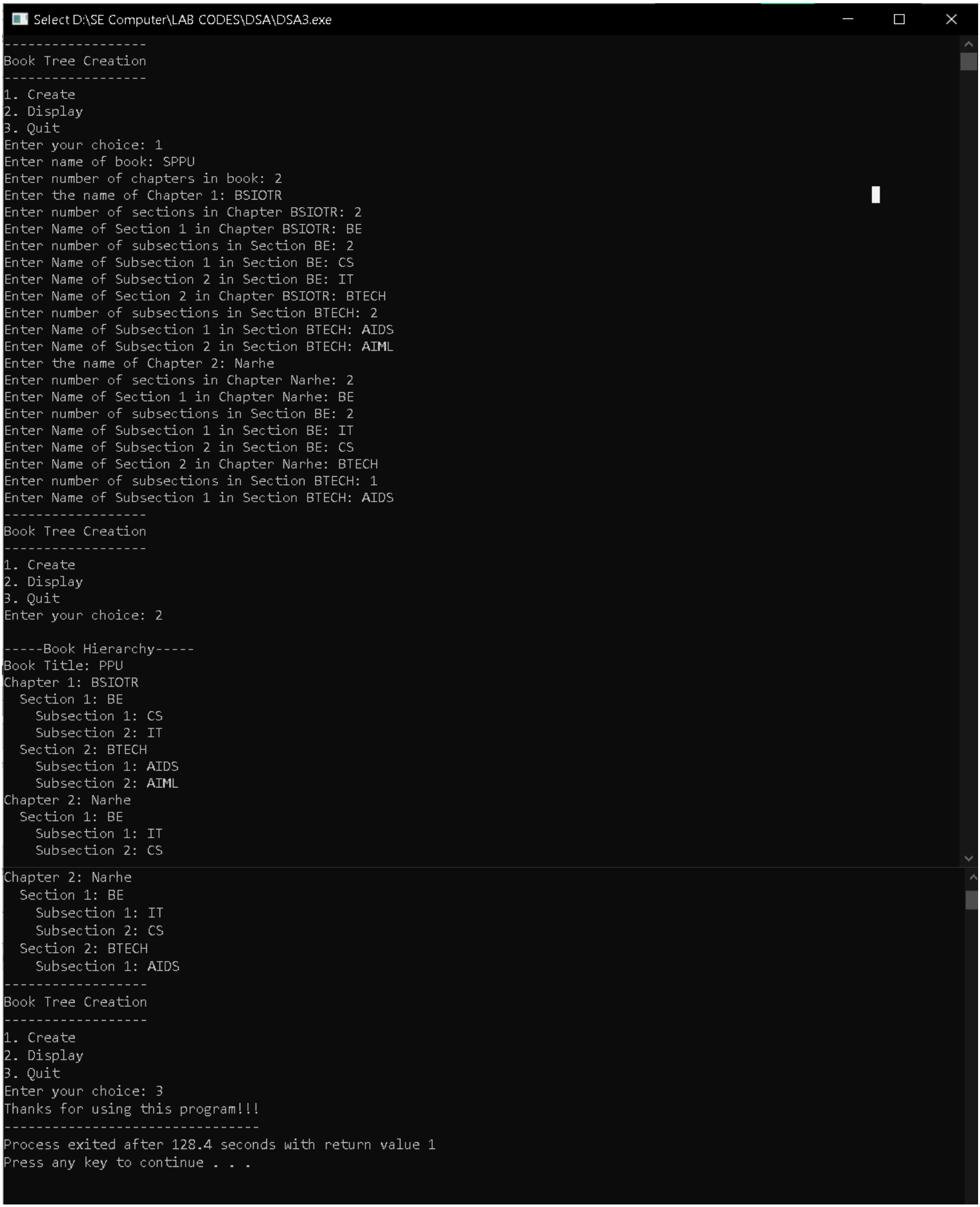
```cpp
#include <iostream>
#include <string.h>
using namespace std;

struct node // Node Declaration
{
    string label;
    int ch_count;            // Number of children (chapters or sections)
    struct node *child[10];  // Child nodes (sections or subsections)
    int sub_count;           // Number of subsections (for section nodes)
} * root;

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

    GT()
    {
        root = NULL;  // Explicitly initializing root to NULL
    }
};

// Function definitions outside the class

void GT::create_tree()
{
    int tchapters, tsections, tsubsections, i, j, k;

    root = new node;
    cout << "Enter name of book: ";
    cin.get();  // To handle any leftover newline
    getline(cin, root->label);  // C++98 compatible method for reading full line

    cout << "Enter number of chapters in book: ";
    cin >> tchapters;
    cin.get();  // Consume the newline character after integer input
    root->ch_count = tchapters;

    for (i = 0; i < tchapters; i++)
    {
        root->child[i] = new node;
        cout << "Enter the name of Chapter " << i + 1 << ": ";
        getline(cin, root->child[i]->label);  // C++98 compatible method for reading full line

        cout << "Enter number of sections in Chapter " << root->child[i]->label << ": ";
```

```
48          cin >> tsections;
49          cin.get();  // Consume the newline character after integer input
50          root->child[i]->ch_count = tsections;
51
52          for (j = 0; j < tsections; j++)
53          {
54              root->child[i]->child[j] = new node;
55              cout << "Enter Name of Section " << j + 1 << " in Chapter " << root->child[i]->label << ": ";
56              getline(cin, root->child[i]->child[j]->label);  // C++98 compatible method for reading full line
57
58              cout << "Enter number of subsections in Section " << root->child[i]->child[j]->label << ": ";
59              cin >> tsubsections;
60              cin.get();  // Consume the newline character after integer input
61              root->child[i]->child[j]->sub_count = tsubsections;
62
63              for (k = 0; k < tsubsections; k++)
64              {
65                  root->child[i]->child[j]->child[k] = new node;
66                  cout << "Enter Name of Subsection " << k + 1 << " in Section " << root->child[i]->child[j]->label << ": ";
67                  getline(cin, root->child[i]->child[j]->child[k]->label);  // C++98 compatible method for reading full line
68              }
69          }
70      }
71  }
72
73  void GT::display(node *r1)
74  {
75      int i, j, k;
76
77      if (r1 != NULL)
78      {
79          cout << "\n-----Book Hierarchy-----";
80          cout << "\nBook Title: " << r1->label;
81          int tchapters = r1->ch_count;
82
83          for (i = 0; i < tchapters; i++)
84          {
85              cout << "\nChapter " << i + 1 << ": " << r1->child[i]->label;
86              int tsections = r1->child[i]->ch_count;
87
88              for (j = 0; j < tsections; j++)
89              {
90                  cout << "\n  Section " << j + 1 << ": " << r1->child[i]->child[j]->label;
91                  int tsubsections = r1->child[i]->child[j]->sub_count;
92
```

```
                for (k = 0; k < tsubsections; k++)
                {
                    cout << "\n    Subsection " << k + 1 << ": " << r1->child[i]->child[j]->child[k]->label;
                }
            }
        }
    }
    cout << endl;
}

int main()
{
    int choice;
    GT gt;

    while (1)
    {
        cout << "----------------------\n";
        cout << "Book Tree Creation\n";
        cout << "----------------------\n";
        cout << "1. Create\n";
        cout << "2. Display\n";
        cout << "3. Quit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        cin.get();  // Consume the newline character after integer input

        switch (choice)
        {
        case 1:
            gt.create_tree();
            break;

        case 2:
            gt.display(root);
            break;

        case 3:
            cout << "Thanks for using this program!!!";
            exit(1);
            break;

        default:
            cout << "Wrong choice!!!" << endl;
        }
    }
    return 0;
}
```

Output :


Select D:\SE Computer\LAB CODES\DSA\DSA3.exe
Book Tree Creation
1. Create
2. Display
3. Quit
Enter your choice: 1
Enter name of book: SPPU
Enter number of chapters in book: 2
Enter the name of Chapter 1: BSIOTR
Enter number of sections in Chapter BSIOTR: 2
Enter Name of Section 1 in Chapter BSIOTR: BE
Enter number of subsections in Section BE: 2
Enter Name of Subsection 1 in Section BE: CS
Enter Name of Subsection 2 in Section BE: IT
Enter Name of Section 2 in Chapter BSIOTR: BTECH
Enter number of subsections in Section BTECH: 2
Enter Name of Subsection 1 in Section BTECH: AIDS
Enter Name of Subsection 2 in Section BTECH: AIML
Enter the name of Chapter 2: Narhe
Enter number of sections in Chapter Narhe: 2
Enter Name of Section 1 in Chapter Narhe: BE
Enter number of subsections in Section BE: 2
Enter Name of Subsection 1 in Section BE: IT
Enter Name of Subsection 2 in Section BE: CS
Enter Name of Section 2 in Chapter Narhe: BTECH
Enter number of subsections in Section BTECH: 1
Enter Name of Subsection 1 in Section BTECH: AIDS
Book Tree Creation
1. Create
2. Display
3. Quit
Enter your choice: 2
-----Book Hierarchy-----
Book Title: PPU
Chapter 1: BSIOTR
Section 1: BE
Subsection 1: CS
Subsection 2: IT
Section 2: BTECH
Subsection 1: AIDS
Subsection 2: AIML
Chapter 2: Narhe
Section 1: BE
Subsection 1: IT
Subsection 2: CS
Chapter 2: Narhe
Section 1: BE
Subsection 1: IT
Subsection 2: CS
Section 2: BTECH
Subsection 1: AIDS
Book Tree Creation
1. Create
2. Display
3. Quit
Enter your choice: 3
Thanks for using this program!!!
Process exited after 128.4 seconds with return value 1
Press any key to continue . . .