**ASSINGMENT NO:-3**

**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> // Use <string> for C++ strings

using namespace std;

// Node Declaration

struct node {

string label; // Label of the node (book, chapter, or section name)

int ch\_count; // Number of children (chapters or sections)

struct node \*child[10]; // Array of pointers to child nodes

} \*root;

// General Tree Class

class GT {

public:

void create\_tree(); // Function to create the tree

void display(node \*r1); // Function to display the tree

GT() {

root = NULL;

}

};

// Function to create the book hierarchy tree

void GT::create\_tree() {

int tchapters;

root = new node;

cin.ignore(); // Clear the input buffer

cout << "Enter name of book: ";

getline(cin, root->label);

cout << "Enter number of chapters in book: ";

cin >> tchapters;

root->ch\_count = tchapters;

for (int i = 0; i < tchapters; i++) {

root->child[i] = new node;

cin.ignore(); // Clear newline character

cout << "Enter the name of Chapter " << i + 1 << ": ";

getline(cin, root->child[i]->label);

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

cin >> root->child[i]->ch\_count;

for (int j = 0; j < root->child[i]->ch\_count; j++) {

root->child[i]->child[j] = new node;

cin.ignore(); // Clear newline character

cout << "Enter Name of Section " << j + 1 << ": ";

getline(cin, root->child[i]->child[j]->label);

}

}

}

// Function to display the book hierarchy

void GT::display(node \*r1) {

if (r1 != NULL) {

cout << "\n----- Book Hierarchy -----";

cout << "\nBook Title: " << r1->label;

for (int i = 0; i < r1->ch\_count; i++) {

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

cout << "\n Sections:";

for (int j = 0; j < r1->child[i]->ch\_count; j++) {

cout << "\n - " << r1->child[i]->child[j]->label;

}

}

cout << endl;

} else {

cout << "\nNo book data found. Please create a tree first.\n";

}

}

// Main function

int main() {

int choice;

GT gt;

while (1) {

cout << "\n--------------------------" << endl;

cout << " Book Tree Menu " << 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();

break; // <-- Important: without this, it falls through to case 2

case 2:

gt.display(root);

break;

case 3:

cout << "Thanks for using this program!" << endl;

exit(0);

default:

cout << "Invalid choice. Please try again!" << endl;

}

}

return 0;

}

OutPut:-



