**Experiment no :3**

**Program code:**

#include<iostream>

using namespace std;

struct node {

string name;

node \*B[5];

};

class book {

int c, s, sub;

public:

node \*temp = new node;

void Getnewnode() {

for(int i = 0; i < 5; i++) {

temp->B[i] = new node;

temp->B[i]->name = "empty";

for(int j = 0; j < 5; j++) {

temp->B[i]->B[j] = new node;

temp->B[i]->B[j]->name = "empty";

for(int k = 0; k < 5; k++) {

temp->B[i]->B[j]->B[k] = new node;

temp->B[i]->B[j]->B[k]->name = "empty";

}

}

}

}

void add\_title() {

cout << ": ENTER THE TITLE OF THE BOOK = ";

cin >> temp->name;

Getnewnode();

}

void add\_chapter() {

string cname;

cout << ": ENTER NUMBER OF CHAPTERS IN THE BOOK = ";

cin >> c;

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

cout << endl << "Chapter " << i + 1 << " = ";

cin >> cname;

temp->B[i]->name = cname;

}

}

void add\_section() {

string sname;

int snum, chnum;

cout << ": CHAPTER NUMBER WHERE YOU NEED TO ADD SECTIONS = ";

cin >> chnum;

cout << ": ENTER NUMBER OF SECTIONS = ";

cin >> snum;

s = snum;

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

cout << endl << "Section " << chnum << "." << i + 1 << " = ";

cin >> sname;

temp->B[chnum-1]->B[i]->name = sname;

}

}

void add\_sub\_section() {

string subname;

int subnum, snum, chnum;

cout << ": CHAPTER NUMBER WHERE YOU WANT TO ADD SECTION = ";

cin >> chnum;

cout << ": SECTION NUMBER WHERE YOU WANT TO ADD SUB-SECTION = ";

cin >> snum;

cout << ": ENTER NUMBER OF SUB-SECTION = ";

cin >> subnum;

sub = subnum;

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

cout << endl << "SUB-SECTION " << chnum << "." << snum << "." << i + 1 << " = ";

cin >> subname;

temp->B[chnum-1]->B[snum-1]->B[i]->name = subname;

}

}

void display() {

cout << endl << "======= INDEX =======" << endl;

cout << endl << ": TITLE : " << temp->name;

cout << endl << ": CHAPTERS : " << endl;

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

if (temp->B[i]->name != "empty")

cout << endl << ": " << i + 1 << ". " << temp->B[i]->name;

for (int j = 0; j < s; j++) {

if (temp->B[i]->B[j]->name != "empty")

cout << endl << " : " << i + 1 << "." << j + 1 << " " << temp->B[i]->B[j]->name;

for (int k = 0; k < sub; k++) {

if (temp->B[i]->B[j]->B[k]->name != "empty")

cout << endl << " : " << i + 1 << "." << j + 1 << "." << k + 1 << " " << temp->B[i]->B[j]->B[k]->name;

}

}

}

cout << endl;

}

};

int main() {

book s;

s.add\_title();

s.add\_chapter();

s.add\_section();

s.add\_sub\_section();

s.display();

return 0;

}

**Output:**

: ENTER THE TITLE OF THE BOOK = DSA

: ENTER NUMBER OF CHAPTERS IN THE BOOK = 2

Chapter 1 = Hashing

Chapter 2 = Trees

: CHAPTER NUMBER WHERE YOU NEED TO ADD SECTIONS = 1

: ENTER NUMBER OF SECTIONS = 2

Section 1.1 = Concepts

Section 1.2 = Operation

: CHAPTER NUMBER WHERE YOU WANT TO ADD SECTION = 1

: SECTION NUMBER WHERE YOU WANT TO ADD SUB-SECTION = 1

: ENTER NUMBER OF SUB-SECTION = 2

SUB-SECTION 1.1.1 = hash\_function

SUB-SECTION 1.1.2 = Hash\_table

======= INDEX =======

: TITLE : DSA

: CHAPTERS :

: 1. Hashing

: 1.1 Concepts

: 1.1.1 hash\_function

: 1.1.2 Hash\_table

: 1.2 Operation

: 2. Trees