

SC165

Problem Statement:

A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes.

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

struct node
{
    char name[20];
    node *next,*down;
    int flag;
};

class list
{
    char ch[20];
    int n,i;
    node *head=NULL,*temp=NULL,*t1=NULL,*t2=NULL;
public:
    node *create();
    void insertbook();
    void insertchapter();
    void insertsection();
    void insertsubsection();
    void displaybook();
};

node *list::create()
{
    node *p=new(struct node);
    p->next=NULL;
    p->down=NULL;
    p->flag=0;
    cout<<"\nEnter the name: ";
    cin>>p->name;
    return p;
}

void list::insertbook()
{
    if(head==NULL)
    {
```

```

        t1=create();
        head=t1;
    }
    else{
        cout<<"\nBook already exists";
    }
}
void list::insertchapter()
{
    if(head==NULL)
    {
        cout<<"\nThere is no book";
    }
    else{
        cout<<"\nHow many chapters are to be inserted?: ";
        cin>>n;
        for(i=0;i<n;i++){
            t1=create();
            if(head->flag==0){
                head->down=t1;
                head->flag=1;
            }
            else{
                temp=head;
                temp=temp->down;
                while(temp->next!=NULL)
                    temp=temp->next;
                temp->next=t1;
            }
        }
    }
}

void list::insertsection()
{
    if(head==NULL)
    {
        cout<<"\nThere is no book";
    }
    else{
        cout<<"\nEnter name of chapter in which you want to enter the section: ";
        cin>>ch;
        temp=head;
        if(temp->flag==0){

```

```

        cout<<"\nThere are no chapters in the book";
    }
    else{
        temp=temp->down;
        while(temp!=NULL){
            if(!strcmp(ch,temp->name)){
                cout<<"\nHow many sections do you want to insert?";
                cin>>n;
                for(i=0;i<n;i++){
                    t1=create();
                    if(temp->flag==0){
                        temp->down=t1;
                        temp->flag=1;
                        t2=temp->down;

                    }
                    else{
                        while(t2->next!=NULL){
                            t2=t2->next;
                        }
                        t2->next=t1;
                    }
                }
                break;
            }
            temp=temp->next;
        }
    }
}
}
}

```

```

void list::insertsubsection(){
    if(head==NULL)
    {
        cout<<"\nThere is no book";
    }
    else{
        cout<<"\nEnter name of chapter in which you want to enter the sub section: ";
        cin>>ch;
        temp=head;
        if(temp->flag==0){
            cout<<"\nThere are no chapters in the book";
        }
        else{
            temp=temp->down;

```

```

while(temp!=NULL){
    if(!strcmp(ch,temp->name)){
        cout<<"\nEnter name of section in which you want to enter the
sub section: ";

        cin>>ch;
        if(temp->flag==0){
            cout<<"\nThere are no sections";
        }
        else{
            temp=temp->down;
            while(temp!=NULL){
                if(!strcmp(ch,temp->name)){
                    cout<<"\nHow many sub sections do
you want to insert?";

                    cin>>n;
                    for(i=0;i<n;i++){
                        t1=create();
                        if(temp->flag==0){
                            temp->down=t1;
                            temp->flag=1;
                            t2=temp->down;
                        }
                        else{
                            while(t2->next!=NULL){
                                t2=t2->next;
                            }
                            t2->next=t1;
                        }
                    }
                    break;
                }
                temp=temp->next;
            }
        }
        temp=temp->next;
    }
}

}

}

void list::displaybook()
{
    if(head==NULL)

```

```

        cout<<"\nBook does not exist";
    else
    {
        temp=head;
        cout<<"\nName of Book: "<<temp->name;
        if(temp->flag==1){
            temp=temp->down;
            while(temp!=NULL){
                cout<<"\nName of Chapter: "<<temp->name;;
                t1=temp;
                if(t1->flag==1){
                    t1=t1->down;
                    while(t1!=NULL){
                        cout<<"\nName of Section: "<<t1->name;
                        t2=t1;
                        if(t2->flag==1){
                            t2=t2->down;
                            while(t2!=NULL){
                                cout<<"\nName of Sub-section:
                                "\n"<<t2->name;
                                t2=t2->next;
                            }
                        }
                        t1=t1->next;
                    }
                }
                temp=temp->next;
            }
        }
    }
}

int main()
{
    list g;
    int x;
    while(1){
        cout<<"\nEnter your choice";
        cout<<"\n1.Insert Book";
        cout<<"\n2.Insert Chapter";
        cout<<"\n3.Insert Section";
        cout<<"\n4.Insert Subsection";
        cout<<"\n5.Display Book";
        cout<<"\n6.Exit";
        cin>>x;
    }
}

```

```

        switch(x){
            case 1:
                g.insertbook();
                break;
            case 2:
                g.insertchapter();
                break;
            case 3:
                g.insertsection();
                break;
            case 4:
                g.insertsubsection();
                break;
            case 5:
                g.displaybook();
                break;
            case 6:
                exit(0);
        }
    }
    return 0;
}

```

Output

Clear

```

Enter your choice
1.Insert Book
2.Insert Chapter
3.Insert Section
4.Insert Subsection
5.Display Book
6.Exit1

Enter the name: DSA

Enter your choice
1.Insert Book
2.Insert Chapter
3.Insert Section
4.Insert Subsection
5.Display Book
6.Exit2

How many chapters are to be inserted?: 3

Enter the name: Hashing

Enter the name: Trees

Enter the name: Graph

Enter your choice
1.Insert Book
2.Insert Chapter
3.Insert Section
4.Insert Subsection
5.Display Book
6.Exit3

Enter name of chapter in which you want to enter the section: Hashing

How many sections do you want to insert?1

```

Output

How many sections do you want to insert?1

Enter the name: Type

Enter your choice

- 1.Insert Book
- 2.Insert Chapter
- 3.Insert Section
- 4.Insert Subsection
- 5.Display Book
- 6.Exit4

Enter name of chapter in which you want to enter the sub section: Hashing

Enter name of section in which you want to enter the sub section: Type

How many sub sections do you want to insert?2

Enter the name: Quadratic

Enter the name: Double

Enter your choice

- 1.Insert Book
- 2.Insert Chapter
- 3.Insert Section
- 4.Insert Subsection
- 5.Display Book
- 6.Exit3

Enter name of chapter in which you want to enter the section: Trees

How many sections do you want to insert?1

Enter the name: BST

Output

Enter your choice

- 1.Insert Book
- 2.Insert Chapter
- 3.Insert Section
- 4.Insert Subsection
- 5.Display Book
- 6.Exit3

Enter name of chapter in which you want to enter the section: Trees

How many sections do you want to insert?1

Enter the name: BST

Enter your choice

- 1.Insert Book
- 2.Insert Chapter
- 3.Insert Section
- 4.Insert Subsection
- 5.Display Book
- 6.Exit5

Name of Book: DSA

Name of Chapter: Hashing

Name of Section: Type

Name of Sub-section: Quadratic

Name of Sub-section: Double

Name of Chapter: Trees

Name of Section: BST

Name of Chapter: Graph

Enter your choice

- 1.Insert Book
- 2.Insert Chapter
- 3.Insert Section
- 4.Insert Subsection
- 5.Display Book
- 6.Exit