Walkthrough:

```
#include<iostream>
using namespace std;
class node {
public:
	int info;
	node *next;
};
class stack {
	node *top;
public:
	stack()
	{
		top = NULL;
	}
	void push()
	{
		int data;
		node *p;
		if ((p = (node*)malloc(sizeof(sizeof(node))) ==NULL)

		{
			cout << "Memory exhausted";
			exit(0);
		}
		cout << "Enter number to insert:";
		cin >> data;
		p = new node;
		p->info = data;
		p->next = NULL;
		if(top!=NULL)
		{
			p->next = top;

		}top = p;
		cout << "\n new item inserted \n";
	}
	void pop()
	{
		node *temp;
		if (top == NULL)
		{
			cout << "\n stack is empty \n";
		}
		else
		{
			temp = top;
			top = top->next;
			cout << "\n the value is popped";
			delete temp;
		}
	}
	void display()
	{
		node *p = top;
		if (top == NULL)
```

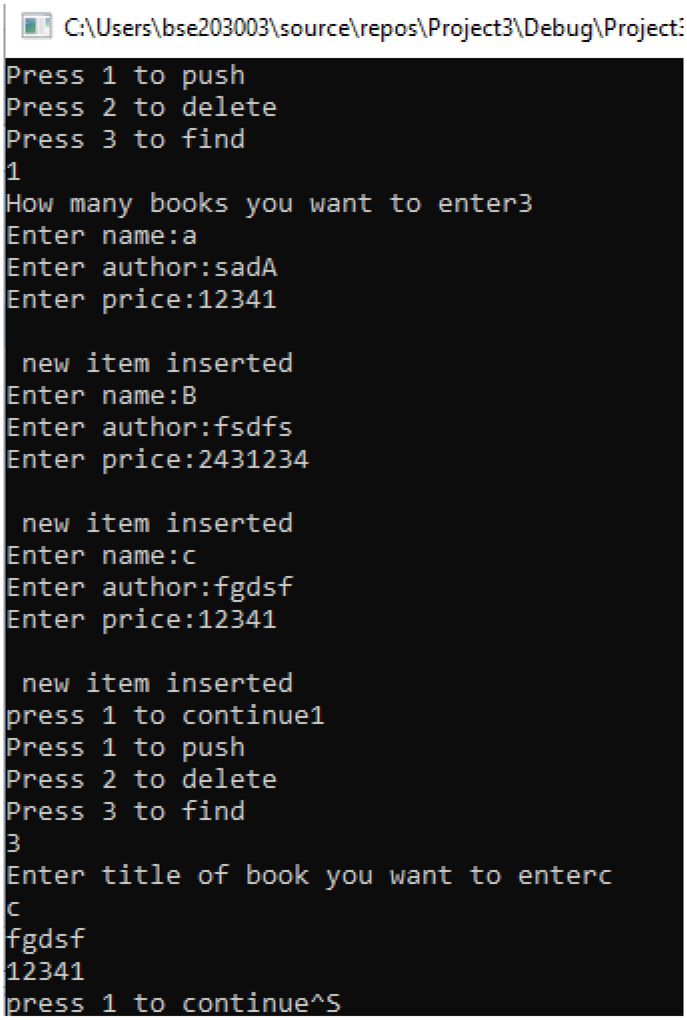
```
        {
            cout << "stack is empty";
        }
        else
        {
            while (p != NULL)
            {
                cout << p->info << endl;
                p = p->next;
            }
        }
    }
};
int main()
{
    stack s;
    int choice,ch;
    do{
        cout << "Press 1 to push" << endl;
        cout << "Press 2 to pop" << endl;
        cout << "Press 3 to display" << endl;
        cin >> choice;
        switch(choice)
            {
        case 1: s.push(); break;
        case 2: s.pop(); break;
        case 3: s.display(); break;
        }cout << "press 1 to continue";
        cin >> ch;
    } while (ch == 1);
}
```

Practice Task 1

```
#include<iostream>
#include<string>
using namespace std;
class node {
public:
    string title;
    string author;
    int price;
    node *next;
};
class stack {
    node *top;
public:
    stack()
    {
        top = NULL;
    }
    void push()
    {
        int pr; string t, a;
        node *p;
        /*if ((p = (node*)malloc(sizeof(sizeof(node))) ==NULL)

        {
```

```
                        cout << "Memory exhausted";
                        exit(0);
                }*/
                cout << "Enter name:"; cin >> t;
                cout << "Enter author:"; cin >> a;
                cout << "Enter price:"; cin >> pr;

                p = new node;
                p->title = t;
                p->author = a;
                p->price = pr;
                p->next = NULL;
                if(top!=NULL)
                {
                        p->next = top;

                }top = p;
                cout << "\n new item inserted \n";
        }
        void pop()
        {
                node *temp;
                if (top == NULL)
                {
                        cout << "\n stack is empty \n";
                }
                else
                {
                        temp = top;
                        top = top->next;
                        cout << "\n the value is popped";
                        delete temp;
                }
        }
        void find(string k)
        {
                node *p = top;
                if (top == NULL)
                {
                        cout << "stack is empty";
                }
                else
                {
                        while (p->next!= NULL && p->title!=k)
                        {

                                p = p->next;
                        }
                        cout << p->title << endl;
                        cout << p->author << endl;
                        cout << p->price << endl;
                }
        }
};
int main()
{
        stack s;
        node n;
```

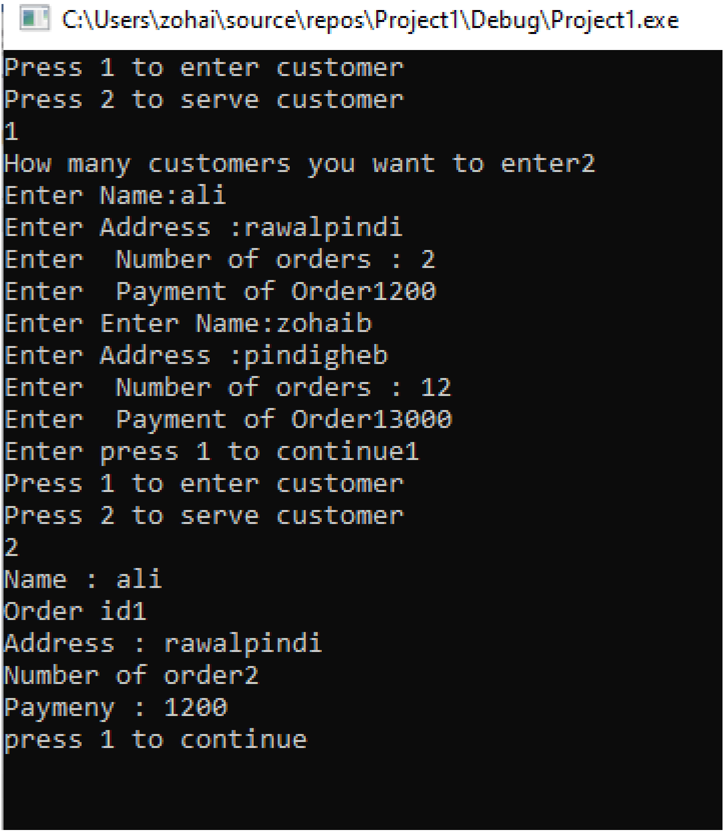
```
    string ss;
    int choice,ch,bb;
    do{
        cout << "Press 1 to push" << endl;
        cout << "Press 2 to delete" << endl;
        cout << "Press 3 to find" << endl;
        cin >> choice;
        switch(choice)
            {
        case 1:
            cout << "How many books you want to enter";
            cin >> bb;
            for (int i = 0; i < bb; i++)
            {
                s.push();
            }
             break;
        case 2: s.pop(); break;
        case 3:cout << "Enter title of book you want to enter";
            cin >> ss;
            s.find(ss); break;
        }cout << "press 1 to continue";
        cin >> ch;
    } while (ch == 1);
}
```



```
Press 1 to push
Press 2 to delete
Press 3 to find
1
How many books you want to enter3
Enter name:a
Enter author:sadA
Enter price:12341

 new item inserted
Enter name:B
Enter author:fsdfs
Enter price:2431234

 new item inserted
Enter name:c
Enter author:fgdsf
Enter price:12341

 new item inserted
press 1 to continue1
Press 1 to push
Press 2 to delete
Press 3 to find
3
Enter title of book you want to enterc
c
fgdsf
12341
press 1 to continue^S
```

Practice Task 2:

```
#include<iostream>
#include<string>
using namespace std;
class node {
public:
        int orderID;
        string name;
        string address;
        int no_order;
        int payment;
        node *next;
        node()
        {
                next = NULL;

        }
        node( string n, string a, int no, int pay)
        {

                name = n;
                address = a;
                no_order = no;
                payment = pay;
        }
        void showdata()
        {
                cout << "Name : " << name << endl;
                cout << "Order id" << orderID << endl;
                cout << "Address : " << address << endl;
                cout << "Number of order" << no_order << endl;
                cout << "Paymeny : " << payment << endl;
        }
};
class queue {
        node *front, *rear;
public:
        queue()
        {
                front = rear = NULL;
        }
        void enqueue( string n, string a, int no, int pay)
        {

                node *q = new node( n, a, no, pay);
                q->orderID++;
                if (rear == NULL)
                {
                        front = rear = q;
                        return;
                }
                rear->next = q;
                rear = q;

        }
        void dequeue()
        {
                if (front == NULL)
                {
                        return;

                }
                node *ptr = front;
                ptr->showdata();
                front = front->next;
                if (front == NULL)
                {
                        rear = NULL;
                }
                delete ptr;
        }

};
int main()
{
        queue q;
        node n;
        string ss;
        string na; string a;
        int no;
int pay;
int choice, ch, bb;
do {
       cout << "Press 1 to enter customer" << endl;
       cout << "Press 2 to serve customer" << endl;

       cin >> choice;
       switch (choice)
       {
       case 1:
              cout << "How many customers you want to enter";
              cin >> bb;
              for (int i = 0; i < bb; i++)
              {
                     cout << "Enter Name:";
                     cin >> na;
                     cout << "Enter Address :";
                     cin >> a;
         cout << "Enter  Number of orders : ";
                     cin >> no;
         cout << "Enter  Payment of Order";
                     cin >> pay;
                     cout << "Enter ";
                     q.enqueue(na,a,no,pay);
              }
              break;
       case 2: q.dequeue(); break;

       }cout << "press 1 to continue";
       cin >> ch;
} while (ch == 1);
```



C:\Users\zohai\source\repos\Project1\Debug\Project1.exe

```
Press 1 to enter customer
Press 2 to serve customer
1
How many customers you want to enter2
Enter Name:ali
Enter Address :rawalpindi
Enter  Number of orders : 2
Enter  Payment of Order1200
Enter Enter Name:zohaib
Enter Address :pindigheb
Enter  Number of orders : 12
Enter  Payment of Order13000
Enter press 1 to continue1
Press 1 to enter customer
Press 2 to serve customer
2
Name : ali
Order id1
Address : rawalpindi
Number of order2
Paymeny : 1200
press 1 to continue
```

```
}
```