**Koneru Lakshmaiah Education Foundation**

**(Deemed to be University)**

**FRESHMAN ENGINEERING DEPARTMENT**

**A Project Based Lab Report**

**On**

**STOCK TABLE**

**SUBMITTED BY:**

**Suggu Sandeep 180030167**

**UNDER THE GUIDANCE OF**

**Ms.Sridevi Emandi**

**ASSOCIATE PROFESSOR**

**(Computer Science Engineering)**



**KL UNIVERSITY**

Green fields, Vaddeswaram – 522 502

Guntur Dt., AP, India.

**DEPARTMENT OF BASIC ENGINEERING SCIENCES-1**



**CERTIFICATE**

This is to certify that the project based laboratory report entitled “**STOCK TABLE**” submitted by **Mr.Suggu Sandeep** bearing Regd. No.**180030167** to the **Department of Basic Engineering Sciences-1, KL University** in partial fulfillment of the requirements for the completion of a project based Laboratory in “TECHNICAL SKILLS-1(CODING)”course in I B Tech I Semester, is a bonafide record of the work carried out by him under my supervision during the academic year 2018 – 2019.

PROJECT SUPERVISOR HEAD OF THE DEPARTMENT

**Ms.Sridevi Emandi Dr. L. SRIDHARA RAO**

**ACKNOWLEDGEMENTS**

It is great pleasure for me to express my gratitude to our honorable President **Sri.Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project based laboratory report.

I express the sincere gratitude to our principal **Dr. N.Venkataram** for his administration towards our academic growth.

I express sincere gratitude to HOD-BES-1 **Dr. L. SridharaRao** for his leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **Ms.Sridevi Emandi** for his novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

Name: Suggu Sandeep

Regd No:180030167

**ABSTRACT**

Turn on the TV news or open a newspaper, surf the internet or listen to the radio, and we will probably come across some information about the stock market. A stock is a share in the ownership of a company. Stock represents a claim on the company's assets and earnings. As we acquire more stock, our ownership stake in the company becomes greater.

A stock table may look intimidating at first because there is a lot of information present. However, to be confident in how to read stocks, we must be able to digest each data point and extract insights from the stock table

In this assignment, we will be using a stack for LIFO accounting. we should use an array based implementation for our stack based implementation or a linked list for implementing our stack. Our stack having records with the following fields.

* + - The name of the stock
    - The number of shares of a stock
    - The purchase price

To create this program we have used the user able to enter information about various stocks, the amount of shares, and the price. This user can then enter a query about a certain stock and the cost according to the LIFO accounting methods for a certain number of share

**Input**

* + Press 1
  + Press 2

**Output**

* To enter a new stock, the user needs to enter the stock symbol, and the number of shares, and the price
* To find the LIFO price for a stock the user needs to enter the stock symbol being queried and the number of shares in question

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S No** | **TITLE** | **PAGE NO** |
| 1 | Introduction | 6-7 |
| 2 | Aim of the Project | 8 |
| 2.1 | Advantages & Disadvantages | 8 |
| 2.2 | Future Implementation | 8 |
| 3 | Software & Hardware Details | 9 |
| 4 | Data Flow Diagram | 10-11 |
| 5 | Algorithm | 12 |
| 6 | Implementation | 13-15 |
| 7 | Integration and System Testing | 16-17 |
| 8 | Conclusion | 18 |

**INTRODUCTION**

C is a structured, high level machine independent language. C is a lower language. C Is a lower language which was understood by the compiler. It allows the software developers to develop programs without worrying about the hardware plat forms where there will be implemented. The c language comes from the ALGOL which gives the concept structured programming to the computer science community. It was introduced early in 1960’s.

After MARTIN RICHARDS DEVELOPED A language that that known as BCPL in1967.for this in 1970’s ken Thompson created a language from BCPL and he called as “B” both BCPL and B are types less system programming languages. After finding ALGOL BCPL, AND B then from this c is evolved from that at BELL LABORATORIES in 1972 by “DENIS RITCHE”.C-uses many concepts from these and added the concept of data type because it was developed along with a UNIX operating system. UNIX is nothing but a most popular net work operating system is used today and the heart of the internet data super high way.C-language is robust language because c-supports richest of operators and burden functions .this consist of many operators, operands, key words, special characters, many characters.

**STACKS:**

A Stack is a data structure which is used to store data in a particular order. Two operations that can be performed on a Stack are: Push operation which inserts an element into the stack. Pop operation which removes the last element that was added into the stack. It follows Last In First Out(LIFO) Order. The C programs in this section dealing with Stack. The section deals with various implementations of Stacks, to reverse a Stack using recursion and without using recursion, to implement two Stacks using a single array and check for Overflow and Underflow conditions and implementing a Stack using linked list. A stack overflow is an undesirable condition in which the program tries to use more memory space than the call stack has available.

If a Stack is empty and yet a Pop operation is attempted, then it results in Stack Underflow condition.

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO(First In Last Out).

Mainly the following three basic operations are performed in the stack:

* **Push:** Adds an item in the stack. If the stack is full, then it is said to be an Overflow condition.
* **Pop:** Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an Underflow condition.
* **Peek or Top:** Returns top element of stack.
* **Empty:** Returns true if stack is empty, else false.

**AIM**

**To Implement The Stock Table**

**Advantages:-**

* Stacks provide a unique way to work with continous memory.
* Very similar to Arrays and Lists, Stacks provide a way for users to access different pieces of contiguous data in a *Last In First Out* manner.

**Disadvantages:-**

* + Inflexible
  + Lack of scalability
  + Unable to Copy & Paste

**Future enhancements:-**

The world is running on C-powered devices. We use these devices every day whether we realize it or not. C is the past, the present, and, as far as we can see, still the future for many areas of software.

**SYSTEM REQUIREMENTS**

* **SOFTWARE REQUIREMENTS:**

The major software requirements of the project are as follows:

Language : Turbo-C

Operating system**:**Windows Xp or later

* **HARDWARE REQUIREMENTS:**

The hardware requirements that map towards the software are as follows:

RAM : 8 GB

Processor : INTEL i5 8TH GEN

**DATA FLOW DIAGRAM**

int id, shares,i,j,choice,c

Float price

Char medicine name, company name mfg\_date, exp\_date,info

i=0

i< 100

m[[i].id=0; m[i].price=0

m[i].share=0

Enter new stock and enter info about the stock

i < 100

choice

choice

Enter new stock and enter info about the stock

A

Print to continue with other options enter 1 else any other number

Enter new stock

Increment n

i++

c++

Information about stock

n+1

i++

Entered name not found

Flag==0

m[i].id==id1

Enter new stock and info

Print enter id of the stock you want to see review and info

c++

Enter new stock (int,number,struct STOCKm)

Enter stock symbol, enter number of shares, enter price

Display stock with ID , added successfully\n,m[number].id

Flag=1

Print these are the details of stock

i++

A

i=0

i<number

**ALGORITHM**

Step1 : Start

Step2 : Define Stock

Step3 : Read medicine name, company, mfg\_date, exp\_date, info

Step4 : Display 1/ enter new stock, 2/ information about the stock

Step5 : Switch (choice)

Case 1. Increment n

Enter new stock

Break

Case 2.

Enter information about stock

Break

Step6 : Print, to continue with other options enter 1 or any other number

Step7 : While (1) {return to step-4}

Step8 : Display enter ID of the stock you want to see Review and info

Step9 : Assign i=0

Increment of i

Step10: if (m\_id==id1)

Print, these are the details of stock

else if ( flag ==0)

Print, entered name not found

else

Print, enter stock symbol

Print, enter number of shares

Step11: Display stock with ID added successfully

Step12: Stop

**IMPLEMENTATION**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<ctype.h>

struct STOCK

{

int id,share;

float price;

char medicneName[100],Company[100],Mfg\_Date[11],Exp\_Date[11],info[5000];

}m[100];

void Informationaboutstock(int number);

void enternewstock(int number,struct STOCK m[ ]);

main( )

{

int i,j,choice,number=0,c;

for(i=0;i<100;i++)

{

m[i].id=0;

m[i].price=0;

m[i].share=0;

}

do

{

printf("Enter\n1 - Enter New Stock\n2 - Information about the stock");

scanf("%d",&choice);

switch(choice)

{

case 1:

{

++number;

enternewstock(number,m);

break;

}

case 2:

{

Informationaboutstock(number+1);

break;

}

}

printf("To Continue with other Options Enter 1 Else any other number\n");

scanf("%d",&c);

}

while(c==1);

}

void Informationaboutstock(int number)

{

int i,flag=0,id1;

char name[100];

printf("Enter ID of the stock you want to see Review and Info\n");

fflush(stdin);

scanf("%d",&id1);

for(i=0;i<number;i++)

{

if(m[i].id==id1)

{

flag=1;

printf("These are the details of Stock\n");

printf("no of shares=%d\n price=%f",m[i].share,m[i].price);

}

}

if(flag==0)

{

printf("Entered Name Not Found\n");

}

}

void enternewstock(int number,struct STOCK m[ ])

{

char name[100];

printf("Enter stock symbol\n");

scanf("%d",&(m[number].id));

fflush(stdin);

printf("Enter number of shares\n");

fflush(stdin);

scanf("%d",&(m[number].share));

printf("Enter price\n");

fflush(stdin);

scanf("%f",&(m[number].price));

printf("stock with id %d Added Successfully\n",m[number].id);

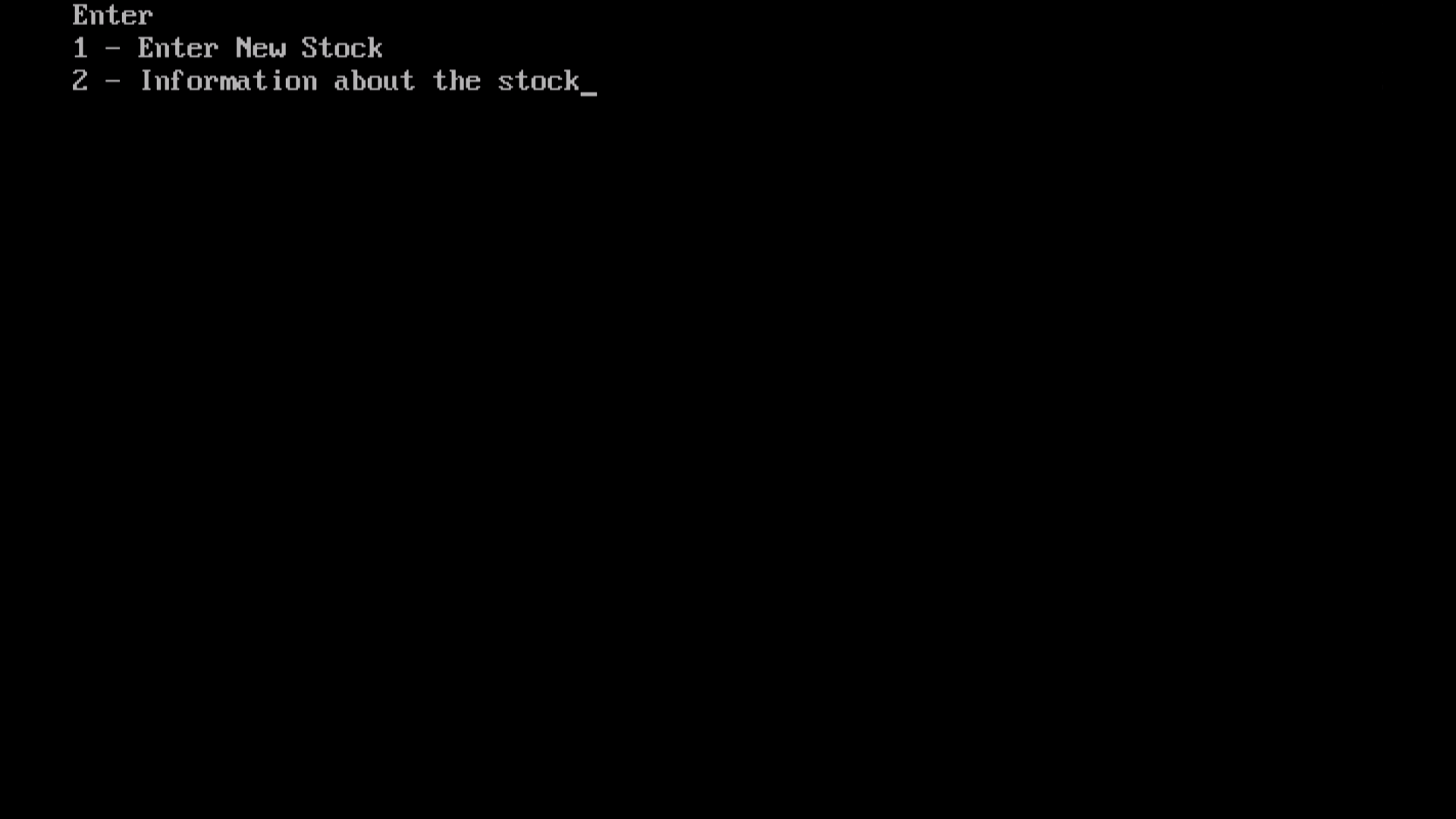
}

**INTEGRATION AND SYSTEM TESTING**

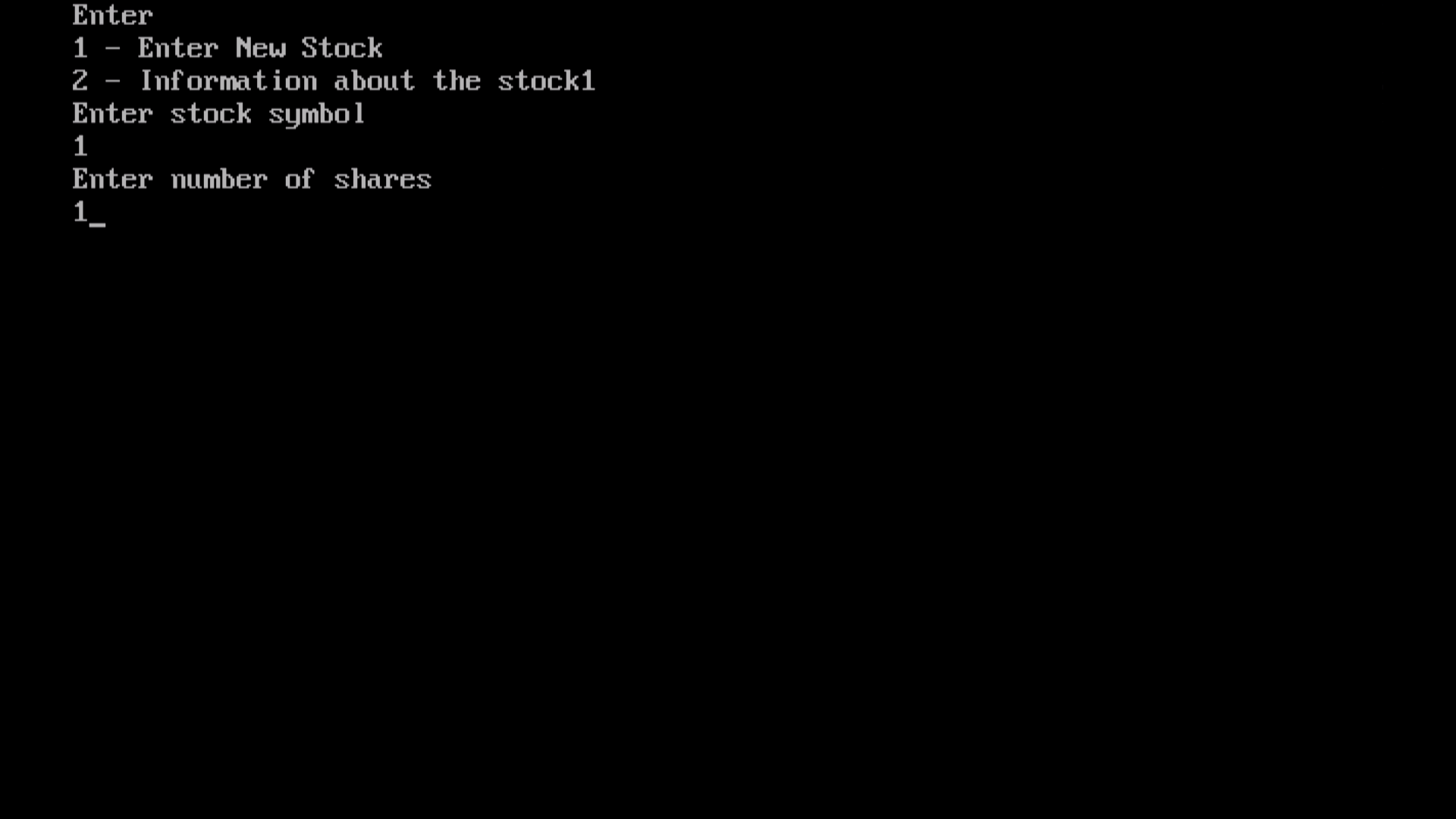
**OUTPUTS**

**Screen Shots:**

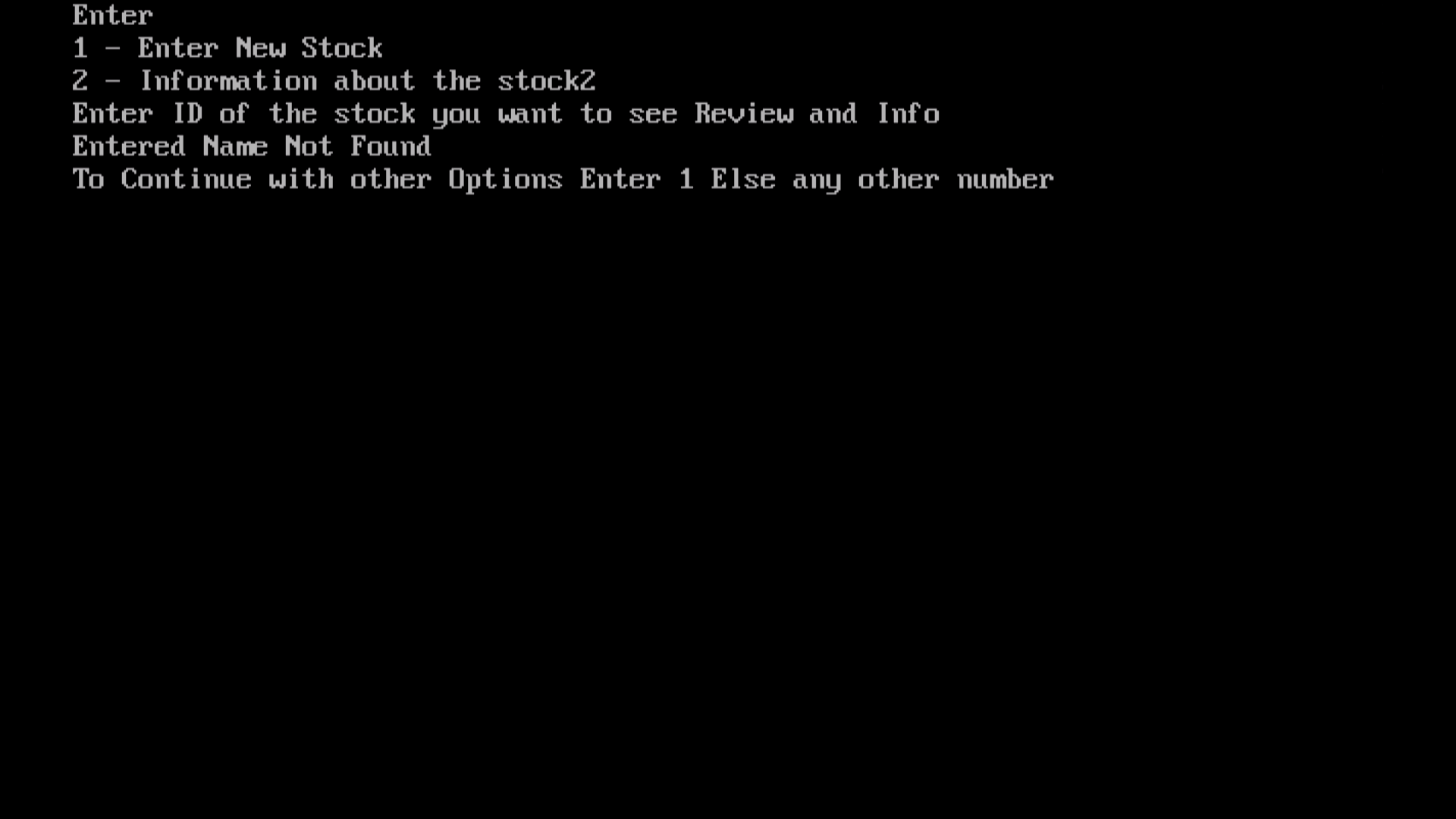
Menu Driven Program:



Entering Stock And Symbol:



Getting Stock LIFO Price:



**CONCLUSION**

The push ( ) operation inserts a node at the front of the list. The two pop ( ) operations remove the node at the front of the list, throwing an exception if the stack is empty. One provides this item in the parameter. Similarly, top ( ) simply returns the item at the front of the list, throwing an exception if the stack is empty. In theory it is possible that the operating system will fail to provide additional memory. To the object when it calls new( )in both the copy constructor and the push( )method, and this failure should be detected in a serious application. Stack is one of the efficient way to implement discipline to system.