VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE2003- Data Structures and Algorithms

J component

Functioning of Super Market Billing system

Slot: L53 + L54

Faculty: GAYATHRI P Mam

Team Members:

Jagirdar Rohit – 19BCE0763

Kandula Mona Reddy – 19BCE0814

Kandra ksheeraj - 19BCE0829

Likhitha Modugula – 19BCT0032

Kotha Brinda Vivek – 19BDS0070

Abstract:

Supermarkets play a major role in people's everyday lives. It is their first preference to buy groceries for their daily needs. This project is useful to both supermarket management and also to the consumers who is going to buy the goods. There is a need to have a perfect program to manage the database of available products and the products that are sold, which holds supermarket business together with achieving its goals.

This project introduces a comprehensive framework for managing the complexity of a supermarket structure for visualizing how a supermarket company actually does business. Using data structures, this project enables to shop without faulty billings. This program is written in C++ and runs via a C++ compilerimplementing stacks, queues, Arrays and linked lists.

It is more computerized, fast, accurate system which is more convenient to the customers and management of the supermarket when compared to the old manual methods. The customer shall be able to have list of items with specific id numbers without rising confusion in similar products. The super market management can easily add, update and delete the items from the database which makes the program more efficient.

Aim:

The aim of the supermarket billing system is to have an efficient way to serve supermarket function fast and accurate.

Objective:

1. To understand the applications and real time usage of different types of data structures such as array, stack, queue, linked list.
2. The main objective of the Supermarket billing system is to build a software and help supermarkets to calculate, display bills and serve the customer in a faster and efficient manner.

3. To construct a strong and efficient algorithm, to develop a program which is editable and can later be used as a module for bigger software mechanism.
4. To develop a real time program which is efficient and has a fast processing and also has an industrial application.

Scope / Applicability:

This project is a traditional supermarket billing system with some added functionality. This system is built for fast data processing and bill generation for supermarket customers. The supermarket billing system is built to help supermarkets calculate and display bills and serve the customer in a faster and efficient manner. The system reduces much of human efforts in calculating bill especially for huge number of products. It provides the list of items without any error. Using data structures, this project enables us to shop without faulty billings. We can have items with specific id number without rising confusion in similar products.

- Saves money and resources of organization and excludes use of paper or sheets in making bill.
- It can detect the product information and their price instantly, that saves time.
- It provides accuracy and faultless billing calculations.
- It is flexible and user-friendly.
- It also displays the purchased products through an electronic bill.

Introduction:

The supermarket billing system automates the basic functions required for the fast and correct billing which is the most important feature for smooth functioning of a supermarket. The purpose of the system is to store details of all the products that are available in the supermarket and have a record of all the items that are available, sold, remaining amount and that are out of stock. The software is made efficient and fast by using the data structures that we have learnt to minimize the complexity in code. This software is designed to ease the work for administrator (user friendly) and reduce the waiting time for the customer (efficient). The customer is also provided with privileges such as discounts to increase the customer count according to the conditions.

Literature Review:

The supermarket billing system in the super market seems to be easy but a supermarket billing software system is built to pass data processing and bill generation for customers. To make this happen various and innovative data structures are used practically.

Introduction

No customer wants to wait for their turn for a long time. This happens with an unorganised billing system. Every customer should know the items available in the supermarket, to purchase the products they intend to buy. All the staff details should be stored to record the attendance and performance of the staff members. Membership card holders should be given extra privileges.

Purpose

A process that is entirely software-based and does not need you to go through paperwork and spend hours on such a tedious task. Membership cards and discounts are given to the customers so as to

boost the business of supermarket. Not only customers but also staff should have a clear idea to add the products which are out of stock and also for products which stock is less in number.

The motive behind this system is to make the customer feel free to make his/her choice in a very convenient manner without any paperwork and the need to wait for their turn and spending precious hours waiting in the queue. This is entirely a software-based process and does not need many workers to do manage the work. It makes the system of billing faster, simpler, and more efficient.

Solution

The system helps to keep track of all the customers and their details who have shopped. The information regarding the products can be easily modified.

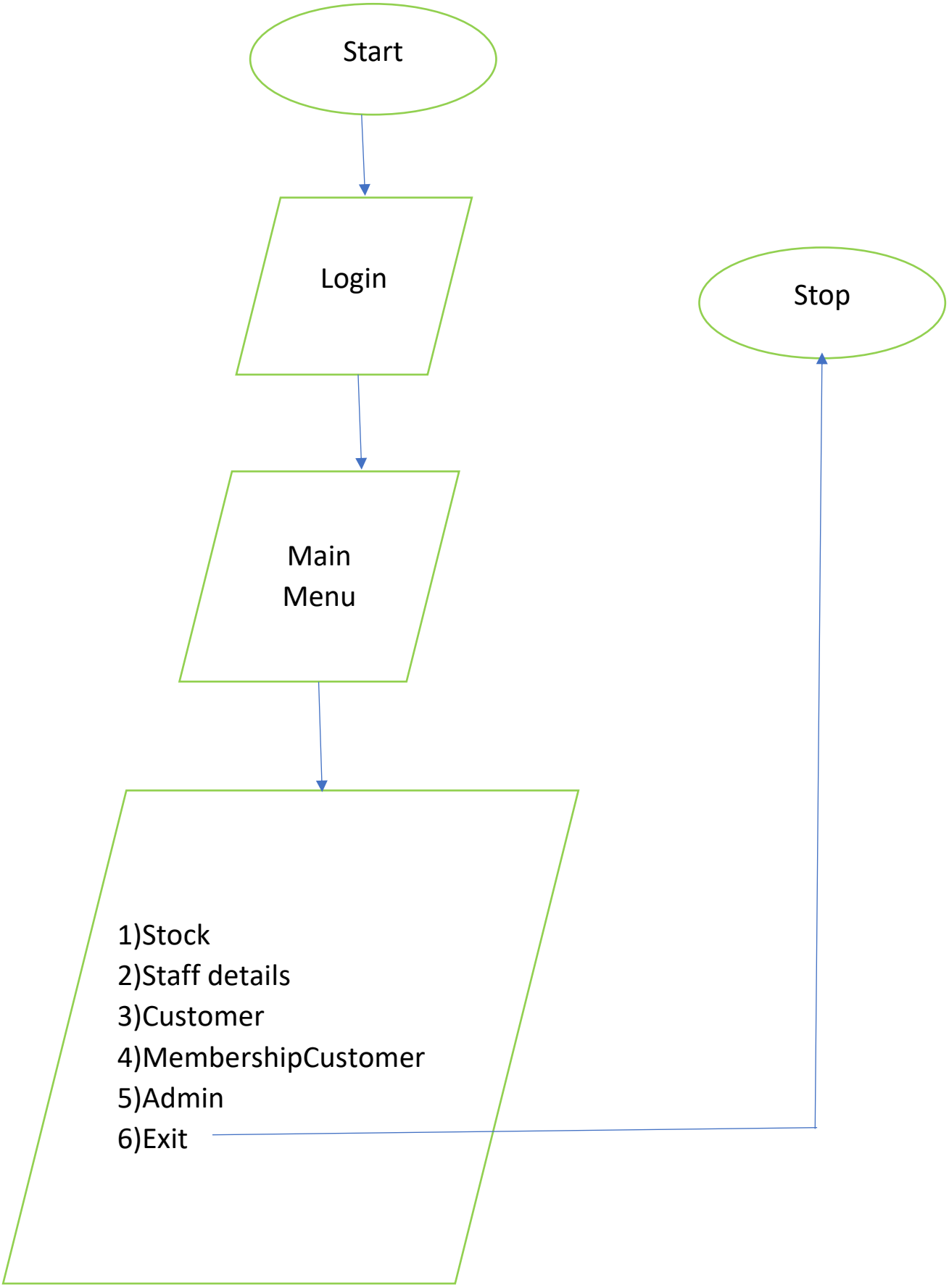
Constructing a strong and efficient algorithm and develop a program which is editable and can later be used as a module for bigger software mechanism.

Conclusion

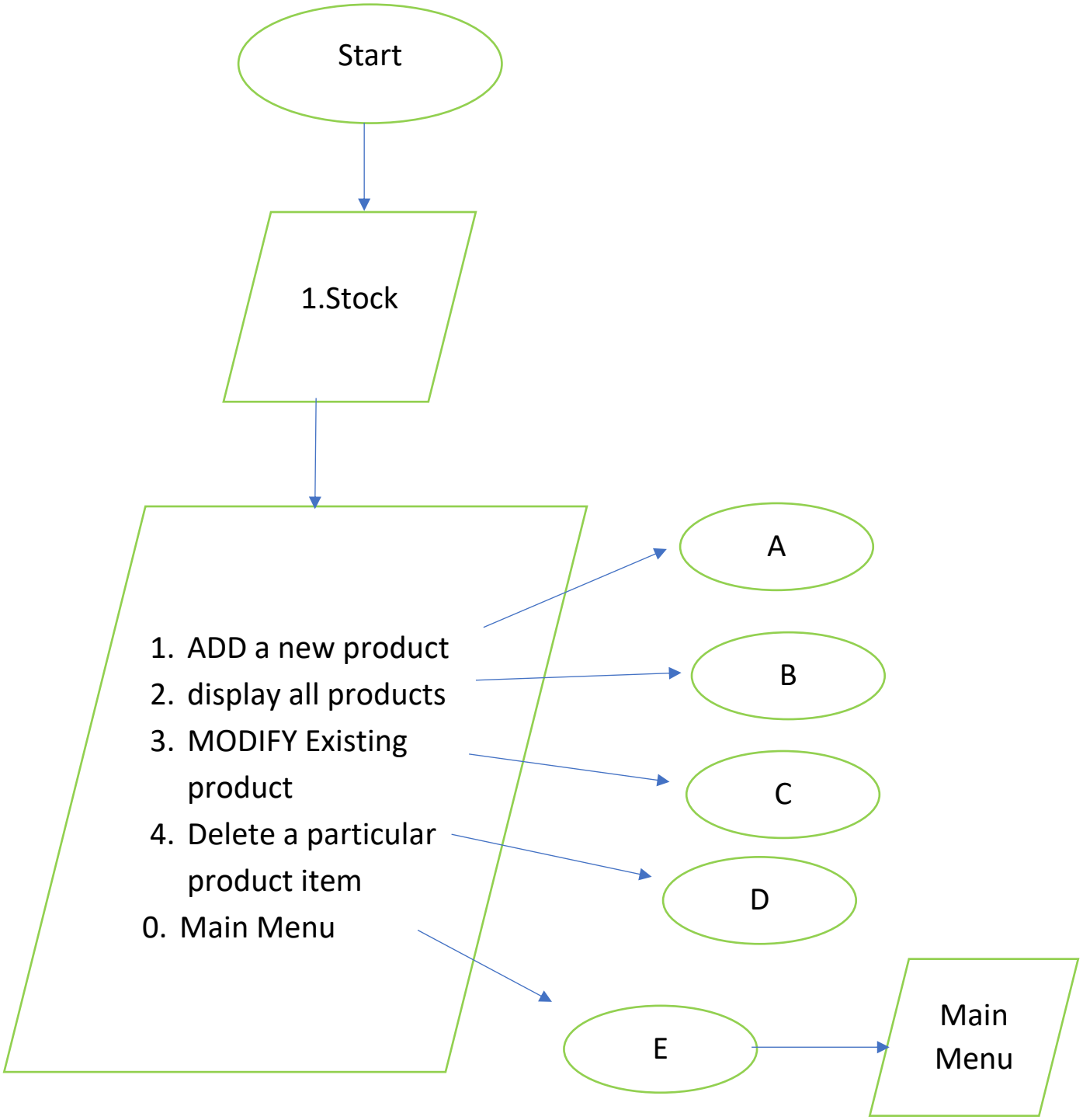
A supermarket billing System now a days is completely being operated through software mode and has more merits than any other manual system present. It makes the process easier, clear, uses less time and energy, and more effective.

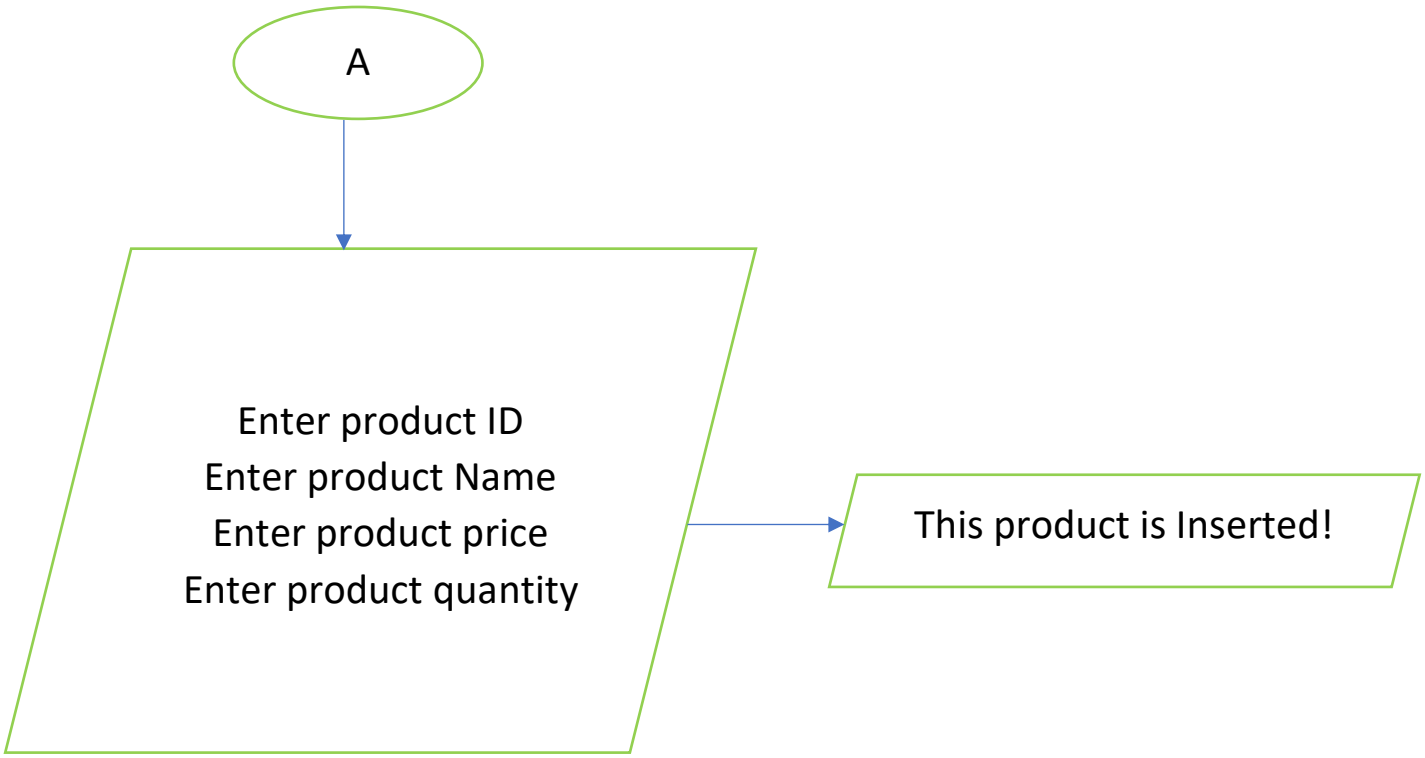
Implementation: (Flowchart)

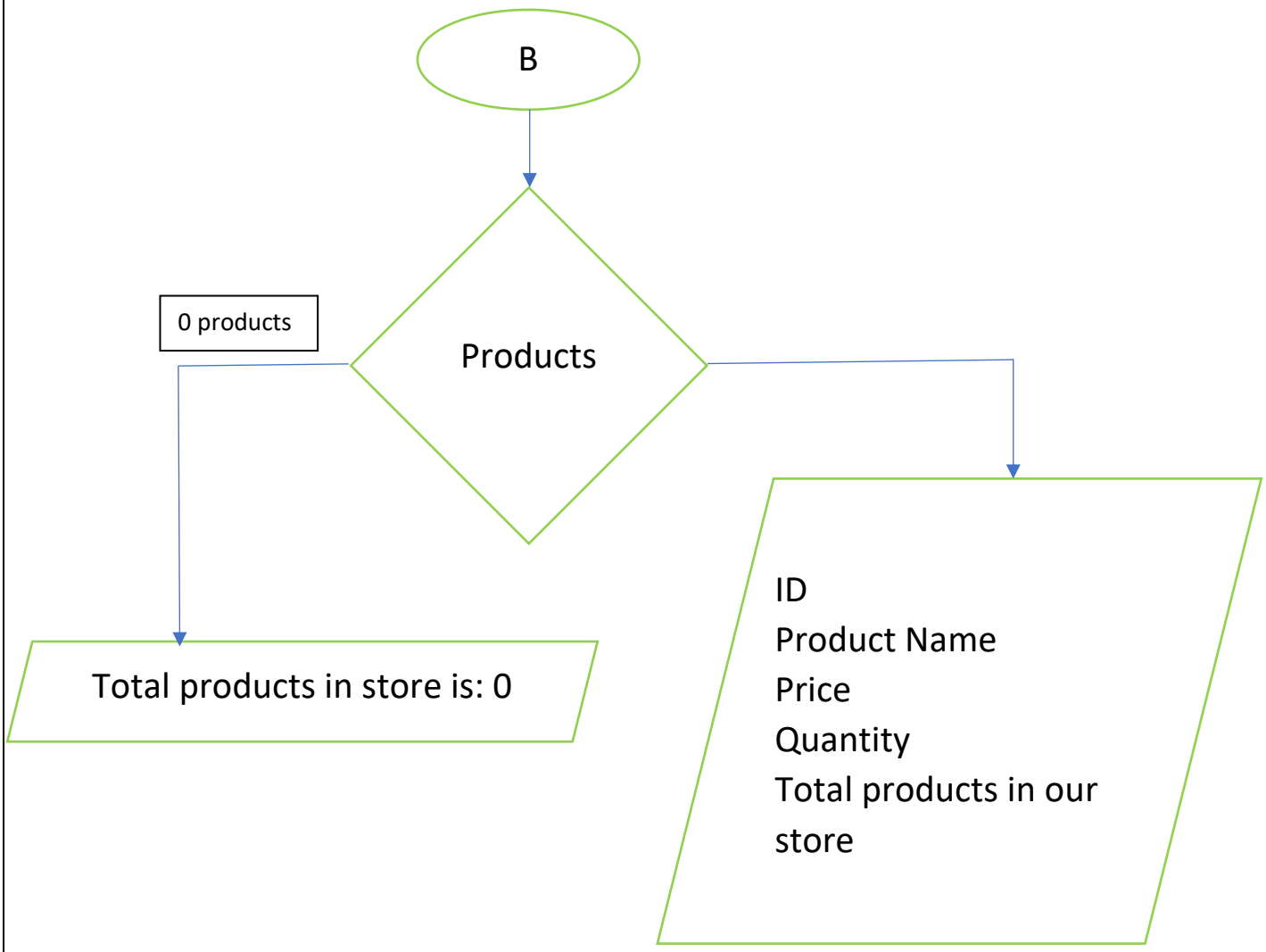
1)**MAIN MENU**

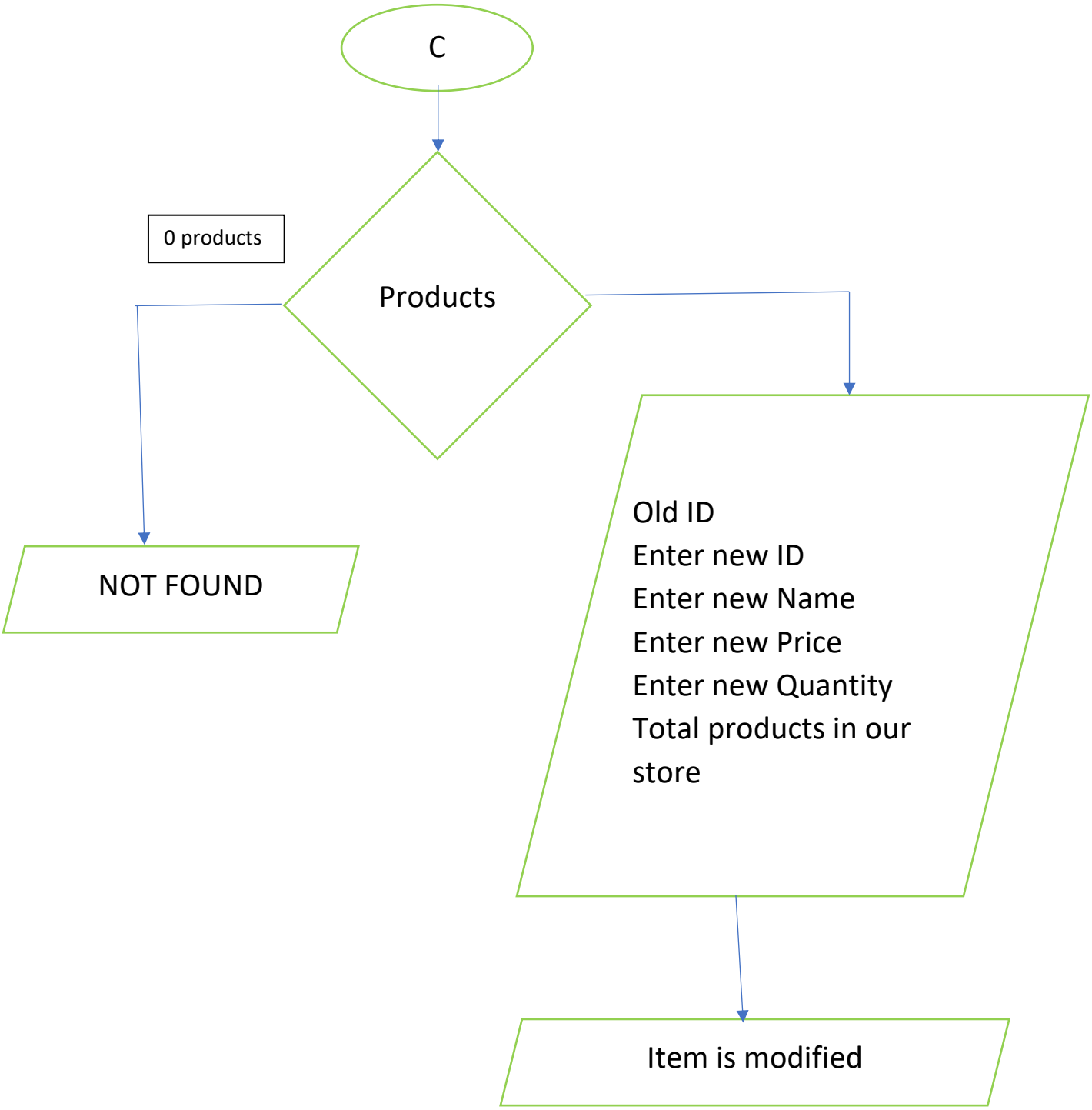


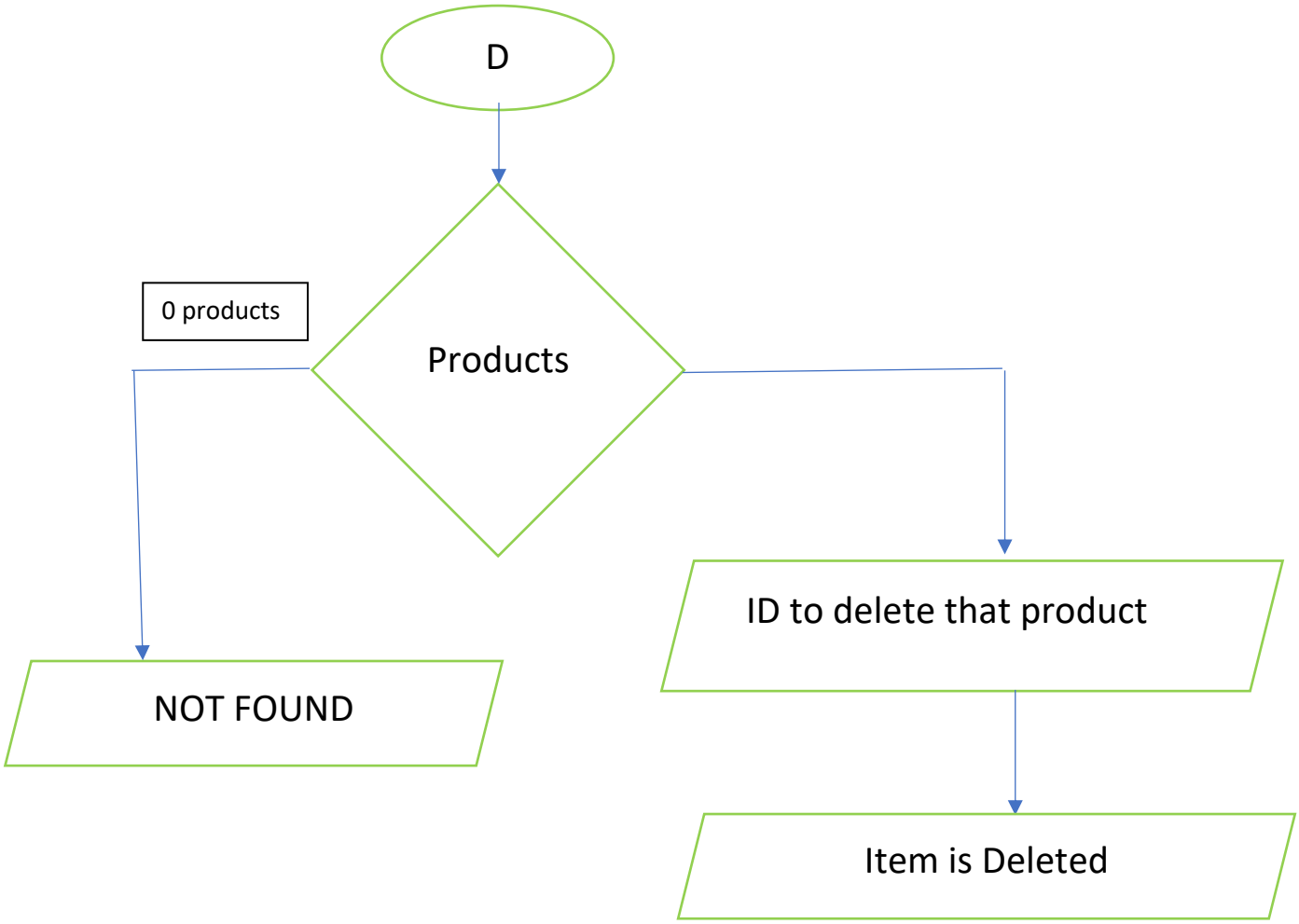
2)STOCK (LINKED LIST,ARRAY)



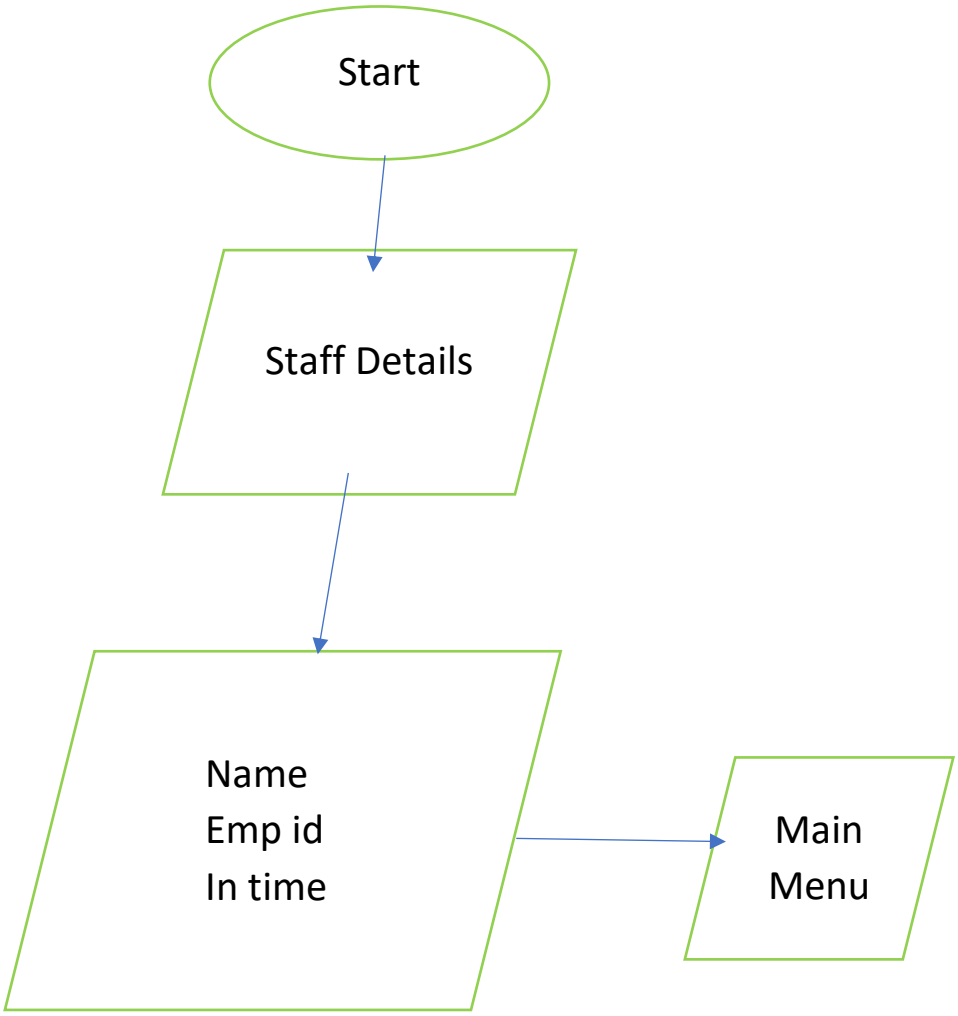

A
Enter product ID
Enter product Name
Enter product price
Enter product quantity
This product is Inserted!


B
Products
0 products
Total products in store is: 0
ID
Product Name
Price
Quantity
Total products in our store

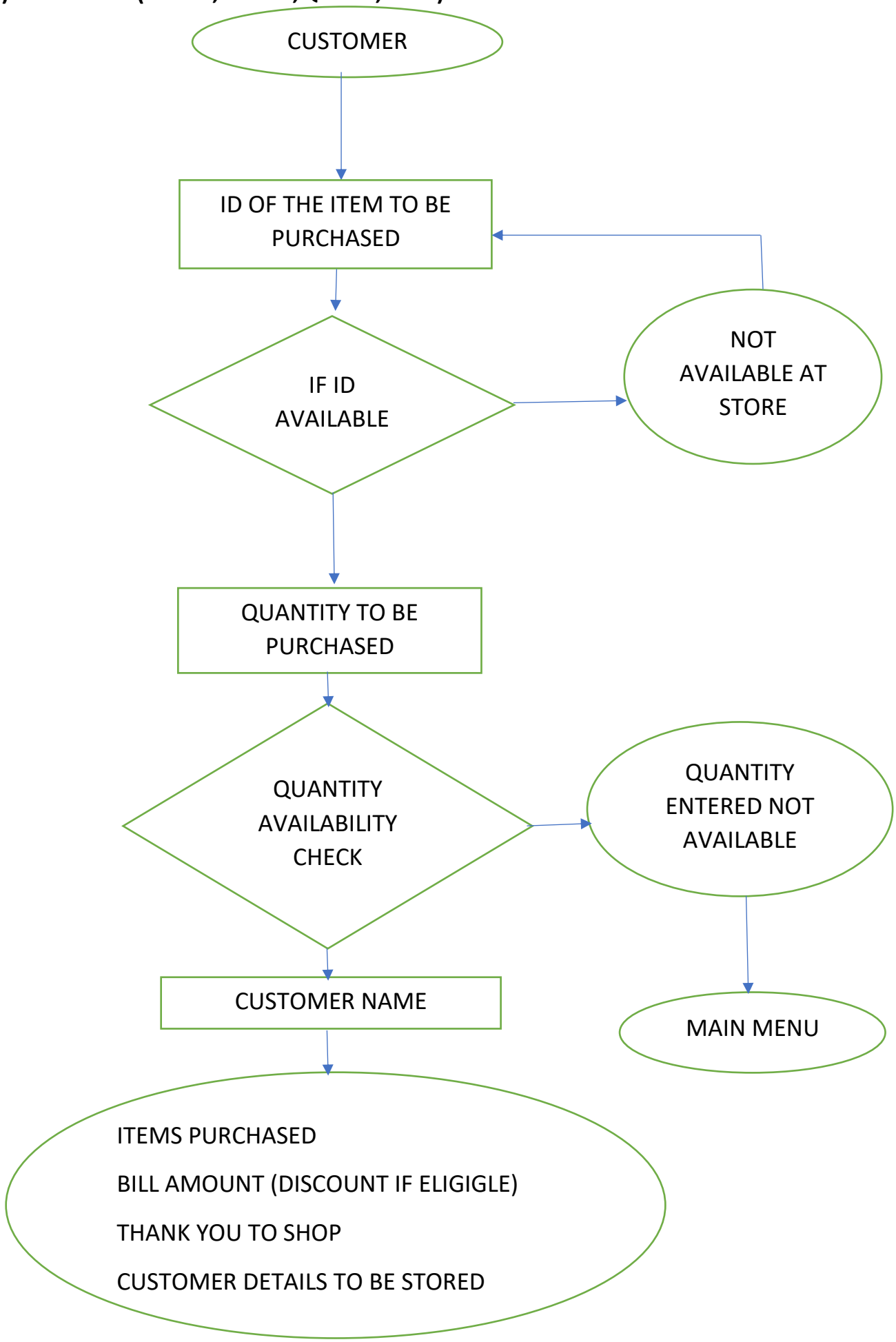

C
0 products
Products
NOT FOUND
Old ID
Enter new ID
Enter new Name
Enter new Price
Enter new Quantity
Total products in our store
Item is modified


D
Products
0 products
NOT FOUND
ID to delete that product
Item is Deleted

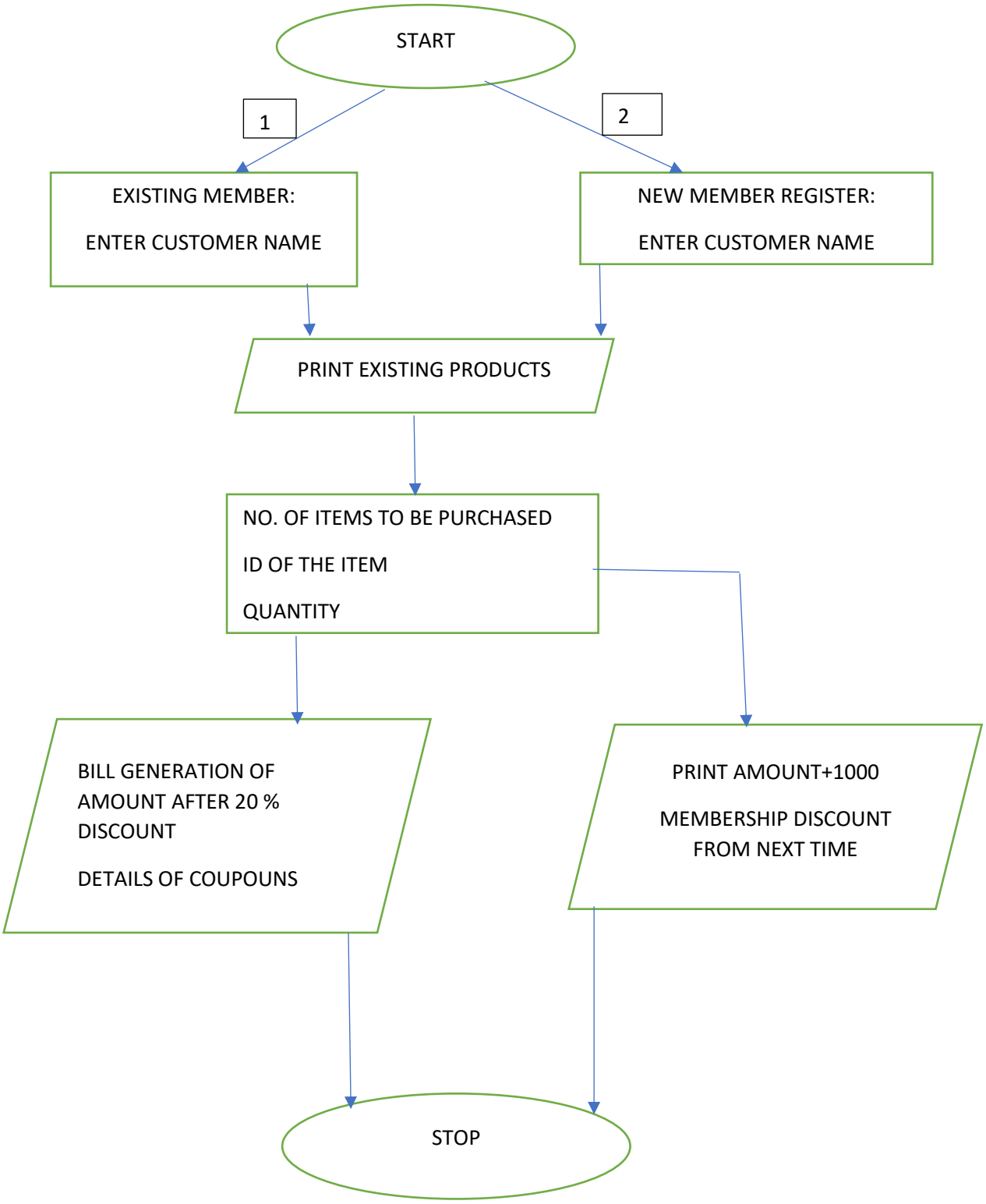
3)STAFF DETAILS (FILES)



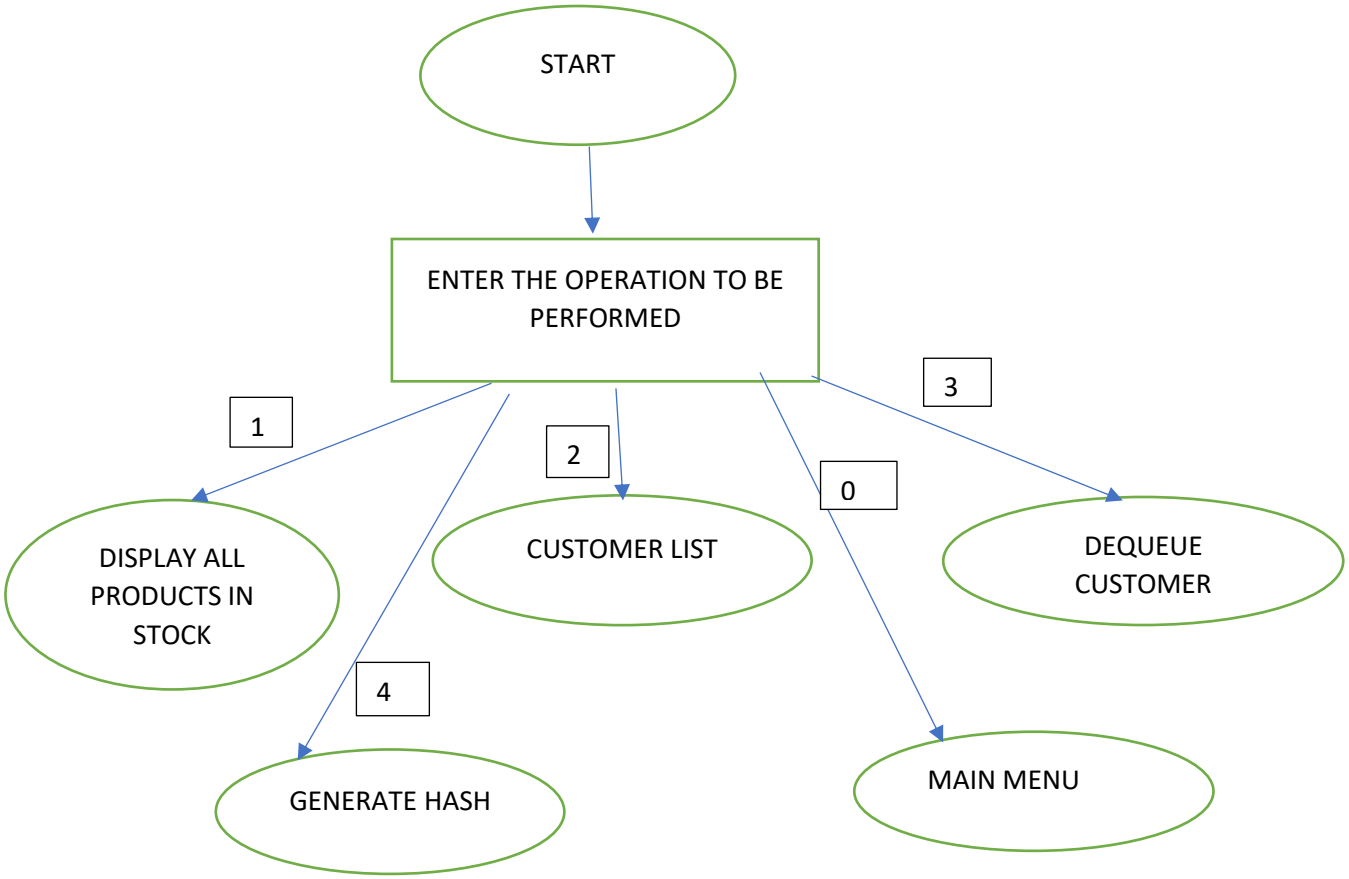
4)CUSTOMER (STACK,ARRAY,QUEUE,FILES)



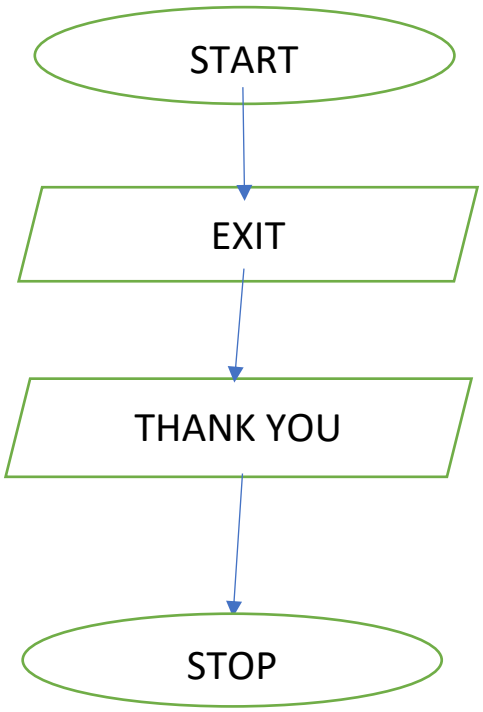
5)MEMBERSHIP CUSTOMER (STACK, ARRAY, QUEUE)



6)ADMIN (ARRAY, QUEUE, HASHING)



7)EXIT



Concepts of data structures used:

a. Linked list
b. Array
c. Stack
d. Queue
e. Hashing
f. Files

List of modules:

- Stock - Rohit Jagirdar
- Staff Details - Ksheeraj Kandra
- Customer - Likhitha Modugula
- Membership Customer - Kotha Brinda Vivek
- Admin - KandulaMona Reddy

Module description:

- **Stock**

  a. Add product: Add product in the store by entering ID,Name, price and Quantity using linked list data structure.

  b. Display Products: The available products are displayed in the store along with their prices and quantity using array data structure.

  c. Modify Products: If the configuration of a product is found to be faulty, or if the products stock has increased, the product details can be modified.

d. Remove items: If the inventory of the product is unavailable or expired the Stock can remove the product from the list.

e. Back to main menu: move back to main menu to go to customer menu or to exit.

- **Staff Details**
  a. The details of all the staff members are taken as the input from the user in the beginning of the day and are stored using file data structure (new data structure).

- **Customer**
  a. Assign basket: Once the customer function is selected, it will assign the customer a trolley number using stack data structure.

  b. Buy items: Customer should enter the number of items he wants to buy and then all the products customer has chosen will go to the assign basket along with the quantity of each item selected. If the product is out of stock or id entered by the user is invalid, it directs us back to the main menu.This can be done using array data structure.

  c. Enter customer queue: After customer has selected all the required items, he/she will be added to the customer queue using queue data structure.

- **Membership customer**
  a. Existing Member: If a customer is already having membership card he’s given special privileges like discount and coupons.

b. New Member Register: If a customer wants to register for a new membership card, the additional amount for it will be added to the final amount and will be given discounts for their next purchases.
c. All the other functions and data structures used are same as the customer module.

- **Admin**

a. Display Products: The available products are displayed in the stock module along with their prices and quantity using array data structure.
b. Customers list: To regulate online traffic, first come first serve has to be implemented. This can be done using linked queue data structure. The customer will be put in a queue of all the other buyers for this product. The customer is added using ENQUEUE.
c. Dequeue customer: Once the customer has been assigned the product, he is taken off the waiting list by DEQUEUE Once the customer's transactions are complete, he has to be removed from the server, can be done using dequeue.
d. Generate Hash: To avoid the crowding of customers, hash is generated and customers are assigned accordingly.

Code:

Header file 1:

queue1.h

```
#include <iostream>

#include<conio.h>

#include<bits/stdc++.h>

using namespace std;


// Structure of Node.

struct Node

{

string cname;


Node *link;

};


Node *front = NULL;

Node *rear = NULL;


//Function to check if queue is empty or not

bool isempty()

{

 if(front == NULL && rear == NULL)

 return true;
```

```
 else
 return false;
}

//function to enter elements in queue
void enqueue ( string name )
{
 Node *ptr = new Node();
 ptr->cname= name;
 ptr->link = NULL;

 //If inserting the first element/node
 if( front == NULL )
 {
  front = ptr;
  rear = ptr;
 }
 else
 {
  rear ->link = ptr;
  rear = ptr;
 }
}
```

```
//function to delete/remove element from queue
void dequeue ( )
{
 if( isempty() )
 cout<<"Queue is empty\n";
 else
 //only one element/node in queue.
 if( front == rear)
 {
  free(front);
  front = rear = NULL;
 }
 else
 {
  Node *ptr = front;
  front = front->link;
  free(ptr);
 }
}

//function to show the element at front
void showfront( )
{
 if( isempty())
```

```
 cout<<"Queue is empty\n";
 else
 cout<<"element at front is:"<<front->cname<<"\n";
}

//function to display queue
void displayQueue()
{
 if (isempty())
  cout<<"Queue is empty\n";
 else
 {
  Node *ptr = front;
  while( ptr !=NULL)
  {
   cout<<"\t"<<ptr->cname<<"\t  "<<endl;
   ptr= ptr->link;

  }
 }
}
```

Header file :2

animation.h

```
#include<iostream>

using namespace std;



COORD coord = {0, 0};

void gotoxy(int x, int y)

{

 COORD coord;

 coord.X = x;

 coord.Y = y;

 SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);

}

//////////////////////////////////////////////////////////////////////////////////////

void animation()

{

	for (int i=45; i>=1; i--)

	{

		Sleep(30);

		gotoxy(1,i);

		//clreol();

	}

	for (int i=1; i<=20; i++)
```

```
        {
                //clreol();
                Sleep(40);
                gotoxy(1,i);
        }
}
```

Header file : 3

stackme.h

```
#include <bits/stdc++.h>
using namespace std;



struct bucket
 {
        int data;
   bucket* link;
   };

struct bucket* top;

void push(int data)
{
        struct bucket* temp;
        temp = new bucket();
```

```
        if (!temp) {
                cout << "\nHeap Overflow";
                exit(1);
        }

        temp->data = data;

        temp->link = top;

        top = temp;
}

int isEmpty()
{
        return top == NULL;
}

int peek()
{

        if (!isEmpty())
                return top->data;
        else
```

```
                exit(1);
}



void bpop()
{
        struct bucket* temp;



        if (top == NULL) {
                cout << "\nStack Underflow" << endl;
                exit(1);
        }
        else {


                temp = top;
                top = top->link;
                temp->link = NULL;


                // release memory of top node
                free(temp);


        }
                cout<<"Your Trolli No is :"<<top->data<<endl;
```

```
        cout<<"                  ___"<<endl;
            cout<<"                /  |"<<endl;
            cout<<"  _______________/   --"<<endl;
            cout<<" |___/__ /___/_|     "<<endl;
            cout<<" |__/___/___/__|     "<<endl;
            cout<<" |_/___/___/___|     "<<endl;
            cout<<"    _______/         "<<endl;
            cout<<"     O   O           "<<endl;



}

void bdisplay()
{
        struct bucket* temp;



        if (top == NULL) {
                cout << "\nStack Underflow";
                exit(1);
        }
        else {
                temp = top;
                while (temp != NULL)
```

```
 {
        cout << temp->data << " ";
                temp = temp->link;
            }
        }
}
//int main() {
//   int ch, val;
//   cout<<"1) Push in stack"<<endl;
//   cout<<"2) Pop from stack"<<endl;
//   cout<<"3) Display stack"<<endl;
//   cout<<"4) Peek stack"<<endl;
//   cout<<"5) Exit"<<endl;
//   do {
//      cout<<"\nEnter choice: "<<endl;
//      cin>>ch;
//      switch(ch) {
//         case 1: {
//            cout<<"Enter value to be pushed:"<<endl;
//            cin>>val;
//            push(val);
//            break;
//         }
//         case 2: {
```

```
//            pop();
//            break;
//        }
//        case 3: {
//            display();
//            break;
//        }
//        case 4:
//                {
//                    cout<<"\npeek value is "<<peek();
//                        break;
//                }
//        case 5: {
//            cout<<"Exit"<<endl;
//            break;
//        }
//        default: {
//            cout<<"Invalid Choice"<<endl;
//        }
//    }
//  }while(ch!=5);
//    return 0;
//}
```

Supermarket billing main code:

```
#include<iostream>

#include<string>

#include <sstream>

#include <bits/stdc++.h>

#include<windows.h>

#include"queue1.h"

#include"animation.h"

#include"stackme.h"


using namespace std;


int search(int);

int display();

string check(int);       //   for checking quantity


//////////////////////////////////////////////////////////////////////////////

struct node

 {

          int ID;

          string proName;

          double prePrice;

          int quantity;

    struct    node* next;
```

```
};

struct node *head=NULL;

//////////////////////////////////////////////////////////////////////////////
 void beg()
{
        system("cls");
        int id,quant;          //  quant    for quantity
        string name;
        double pre;            //  pre for price
        struct node *t=new node;
        struct node *p=head;



        cout<<"\t\t\tEnter product ID:-";
        cin>>id;
        t->ID=id;
        cout<<"\t\t\tEnter product Name:-";
        cin>>name;
        t->proName=name;
        cout<<"\t\t\tEnter product price:-";
        cin>>pre;
        t->prePrice=pre;
```

```
        cout<<"\t\t\tEnter product quantity:-";
        cin>>quant;
        t->quantity=quant;
        if(head==NULL)
        {
        t->next=head;
        head=t;
        }
        else
        {
                while(p->next!=NULL)
{
                p=p->next;
                  }
    p->next=t;
    t->next=NULL;


        }
        system("cls");
cout<<"\n\n\t\t\t\tThis product is Inserted!\n\n\n";


        }
```

```
////////////////////////////////////////////////////////////////////////////////////////////////
        int search(int id)                  //    for search item in list
 {
        int count=1;
        struct node *p=head;
        while(p!=NULL)
        {
                if(p->ID==id)
                        break;
                else
                        count++;
                        p=p->next;
        }
        return count;
 }
 ////////////////////////////////////////////////////////////////////////////////////
 int hash(int x, int mod)
{
        return x % mod;


}


////////////////////////////////////////////////////////////////////////////////////
        void delPro()
```

```
        {
                system("cls");
                display();
                int id;
                struct node *cur=head;
                struct node *pre=head;
                cout<<"\n\nEnter ID to delete that product:\n\n";
                cin>>id;
                 if (head == NULL)
    {
        system("cls");
      cout<<"List is empty"<<endl;
    }
        int pos=0;
        int count=display();            //   for load no of nodes
        pos=search(id);                    //   for check weather desire node is exist or not
        if(pos<=count){

                while(cur->ID!=id){              //  for delete middle area products
                        pre=cur;
                        cur=cur->next;
  }
                pre->next=cur->next;
                system("cls");
```

```
                cout<<"\n<<item is deleted>>\n";
        }else{
                cout<<"\n<<<Not found>>\n\n";
        }
        }

////////////////////////////////////////////////////////////////////////////////
        void modify()
        {
                int id;
                double pre;       //    pre for price
                string pName;
                int nid; int nq;     //   pName for new name
                if (head == NULL)
    {
        system("cls");
      cout<<"List is empty"<<endl;
    }
       else
        {
            display();
                cout<<"\n\nEnter ID to modify product Name and its price:\n";
                cin>>id;
                struct node *cur=head;
```

```
            int pos=0;
            int count=display();            //   for load no of nodes
        pos=search(id);
                        //   for check weather desire node is exist or not
        if(pos<=count)
    {
            while(cur->ID!=id)
      {
                    cur=cur->next;
            }
            cout<<"\nOld ID : "<<cur->ID<<endl;
            cout<<"\nOld Name : "<<cur->proName<<endl;
            cout<<"\nOld Price : "<<cur->prePrice<<endl;
            cout<<"\nOld Quantity : "<<cur->quantity<<endl;

            cout<<endl<<endl;
            cout<<"Enter new ID:";
            cin>>nid;
            cur->ID=nid;
            cout<<"Enter new Name:";
            cin>>pName;
            cur->proName=pName;
            cout<<"Enter new Price:";
            cin>>pre;
```

```
                cur->prePrice=pre;
                cout<<"Enter new Quantity:";
                cin>>nq;
                cur->quantity=nq;
        }

            else
    {
                cout<<id<<" is <<<Not found>>\n\n";
        }
        }
}

////////////////////////////////////////////////////////////////////////////////////////////////

int display()
{
                system("cls");
                int c=0;            //   c for count products
                struct node *p=head;
                cout<<"Existing products are:\n";
                cout<<"ID\t\tProduct Name\t\tPrice\t\tQuantity\n";

        cout<<"=================================================
============|\n";
```

```
                while(p!=NULL)
                {
                        cout<<p->ID<<"\t\t"<<p->proName<<"\t\t\t"<<p-
>prePrice<<"\t\t\t"<<check(p->quantity)<<"\n"; //    call   check func and pass quantity
                        p=p->next;
                        c=c+1;
                }
                cout<<"\nTotal products in our store is : "<<c<<"\n\n\n";
                return c;
            }
////////////////////////////////////////////////////////////////////////////////////////////////////
            string check(int quant)

        {           //        check function
            int a = quant;
      stringstream ss;
      ss << a;
      string quantity = ss.str();

                if(quant<=0)
                return "out of stock!";
                else
                return quantity;
                }
```

```
//////////////////////////////////////////////////////////////////////////////
        void buy()
        {
      system("cls");
      display();
            string products[20];
              //   for display sold items
            int pay=0,no,c=0,price,id,i=1;


      if(head==NULL)
      {
          cout<<"\n<<<<There is no items to buy>>>>\n\n";
      }
          else
      {
            cout<<"How many items you want to buy!\n";
            cin>>no;
      int count=display();          //   for store no of nodes  in c
            while (i<=no)
      {
                    struct node *cur=head;
        int quant,cho;   a:          //   quant   for quantity  and cho for choice
        cout<<"Enter id of item that you want to buy: ";
        int id,pos=0;
```

```
    cin>>id;
    if(id==-1){system("cls"); return;    }
          pos=search(id);
          if(pos<=count)
{
                        //     item is available in store
            while(cur->ID!=id)
{
                    cur=cur->next;
            }
   cout<<"How many quantities you want:";
         cin>>quant;
         if(cur->quantity<quant)
   {
      cout<<"\n\n\t\t\t----The Quantity You Entered is not available---"<<endl;
      cout<<"\n\t\t\t-----(Press -1 for Back to Main Menu)------"<<endl;
      goto a;


   }
   products[c]=cur->proName;   // this will conatin the items buy names in array;
              c++;


         pay=pay+(cur->prePrice*quant);         //     calculate Bill
         cur->quantity=cur->quantity-quant;           //    change quantity
```

```
                i++;


              }
             else
        {


                    cout<<"\n<<<<<<<<<This item is not available in our store at this
time>>>>\n\n";


             }
}
        string customer;
        cout<<"\n\t\t Enter Your Name :"; cin>>customer;
        enqueue(customer);
             system("cls");


        cout<<"\n\n\n\n\t\t\tYou have bought : ";
        for(int i=0;i<no;i++)
       {              //    show that item you have bought
        cout<<products[i]<<" ";
       }
          if(pay>=3000){
        price=pay*(0.90);           //    with 10% discount
             cout<<"\n\nOriginal price : "<<pay;
        cout<<"\n with 10% discount: "<<price<<"\nThank you for shopping !\n\n";
```

```
 }
else{
price=pay;          //    with no discount
cout<<"\n Bill Amount: "<<price<<"\nThank you for shopping !\n\n";
          }


          {
 ofstream fout;
          string line;
          fout.open("recent_customer.txt");


          while (fout) {
                    getline(cin, line);
                    if (line == "-1")
                    break;
                    fout << line << endl;
          }
                    fout.close();
                    ifstream fin;
 }




            }
```

```
        }



//////////////////////////////////////////////////////////////////////////////////////////

int membership()

{

            string customer;

    cout<<"\n\t\t Enter Your Name :";

            cin>>customer;

    enqueue(customer);

    system("cls");

    display();

            string products[20];

              //   for display sold items

            int pay=0,no,c=0,price,id,i=1;


    if(head==NULL)

    {

        cout<<"\n<<<<There is no items to buy>>>>\n\n";

    }

        else

    {

            cout<<"How many items you want to buy!\n";

            cin>>no;
```

```
int count=display();          //   for store no of nodes  in c
            while (i<=no)
{
                    struct node *cur=head;
    int quant,cho;   a:          //   quant   for quantity  and cho for choice
    cout<<"Enter id of item that you want to buy: ";
    int id,pos=0;
    cin>>id;
    if(id==-1)
                    {
                    system("cls");
                    }
          pos=search(id);
          if(pos<=count)
{
                      //     item is available in store
             while(cur->ID!=id)
{
                    cur=cur->next;
             }
   cout<<"How many quantities you want:";
         cin>>quant;
         if(cur->quantity<quant)
   {
```

```
            cout<<"\n\n\t\t\t----The Quantity You Entered is not available---"<<endl;
            cout<<"\n\t\t\t-----(Press -1 for Back to Main Menu)------"<<endl;
            goto a;

          }
          products[c]=cur->proName;   // this will conatin the items buy names in array;
                    c++;

              pay=pay+(cur->prePrice*quant);         //     calculate Bill
              cur->quantity=cur->quantity-quant;           //    change quantity
              i++;

             }
            else
        {

                  cout<<"\n<<<<<<<<<This item is not available in our store at this
time>>>>\n\n";

}
}

            system("cls");
        cout<<"\n\n\n\n\t\t\tYou have bought : ";
        for(int i=0;i<no;i++)
```

```
    {             //    show that item you have bought
      cout<<products[i]<<" ";
     }
            price=pay*(0.80);          //    with 20% discount as member
            cout<<"\n\nOriginal price : "<<pay;
      cout<<"\n with 20% discount: "<<price;
   }


      if(pay>=3000){
                                                                    //    coupon
      cout<<"\n You have won coupon worth rupees 500 "<< "\nThank you for
shopping !\n\n";
                  }
       else{
       cout<<"\n shop more to win gift coupons"<< "\nThank you for shopping !\n\n";
                  }


}


 ////////////////////////////////////////////////////////////////////////////////////////////////
  int newmembership()
 {
                  string customer;
      cout<<"\n\t\t Enter Your Name :";
                  cin>>customer;
```

```
enqueue(customer);
system("cls");
display();
        string products[20];
          //   for display sold items
        int pay=0,no,c=0,price,id,i=1;


if(head==NULL)
{
    cout<<"\n<<<<There is no items to buy>>>>\n\n";
}
    else
{
        cout<<"How many items you want to buy!\n";
        cin>>no;
int count=display();        //   for store no of nodes  in c
        while (i<=no)
{
                struct node *cur=head;
  int quant,cho;   a:        //   quant   for quantity  and cho for choice
  cout<<"Enter id of item that you want to buy: ";
  int id,pos=0;
  cin>>id;
  if(id==-1)
```

```
                    {
                    system("cls");
                    }
        pos=search(id);
        if(pos<=count)
{
                    //     item is available in store
          while(cur->ID!=id)
{
                    cur=cur->next;
          }
   cout<<"How many quantities you want:";
        cin>>quant;
        if(cur->quantity<quant)
   {
      cout<<"\n\n\t\t\t----The Quantity You Entered is not available---"<<endl;
      cout<<"\n\t\t\t-----(Press -1 for Back to Main Menu)------"<<endl;
      goto a;


   }
   products[c]=cur->proName;   // this will conatin the items buy names in array;
            c++;


        pay=pay+(cur->prePrice*quant);         //     calculate Bill
```

```
                cur->quantity=cur->quantity-quant;          //    change quantity
                i++;


              }
             else
        {


                    cout<<"\n<<<<<<<<<<This item is not available in our store at this
time>>>>\n\n";


}
}


             system("cls");
        cout<<"\n\n\n\n\t\t\tYou have bought : ";
        for(int i=0;i<no;i++)
       {              //    show that item you have bought
        cout<<products[i]<<" ";
        }
             price=(pay+1000);          //    amount  + membership (will get discount from
next time)
             cout<<"\n\nOriginal price : "<<pay;
        cout<<"\n with purchase of membership card: "<<price; //adding price of
membership card
}
```

```
        cout<<"\n You will get membership discount from next time "<< "\nThank you
for shopping !\n\n";



}

 ////////////////////////////////////////////////////////////////////////////////////////////////////
   int stock()
          {
                    system("cls");
                              int ch;



          do {


                //            choice for below message




          cout<<"\t\t=========================================="<<endl;
          cout<<"\t\t|                Stock                    |"<<endl;
          cout<<"\t\t=========================================="<<endl;


          cout<<"\t\t     Enter 1 for ADD a new product    "<<endl;
          cout<<"\t\t     Enter 2 to display all products   "<<endl;
```

```
        cout<<"\t\t     Enter 3 for MODIFY Existing product"<<endl;
        cout<<"\t\t     Enter 4 for Delete a particular product item"<<endl;
        cout<<"\t\t     Enter 0 for Main Menu"<<endl;



        cout<<"\nEnter Your choice >>>";   cin>>ch;
        switch(ch){
        case 1:
        beg();
        break;
case 2:
        system("cls");
   display();
   break;
case 3:
        modify();
        system("cls");
        break;
case 4:
        delPro();
cout<<"\n-------Product is Deleted-------\n";
        break;


default: system("cls");
```

```
        }
}
while(ch!=0)  ;
}

//////////////////////////////////////////////////
void administator()
        {
                int ch;
                system("cls");




        do {

            //          choice for below message




        cout<<"\t\t=============================================="<<endl;
        cout<<"\t\t|         Administator Portal             |"<<endl;
        cout<<"\t\t=============================================="<<endl;

        cout<<"\t\t     Enter 1 to display all products   "<<endl;
```

```
        cout<<"\t\t     Enter 2 for Customers List "<<endl;
        cout<<"\t\t     Enter 3 for Dequeue customer"<<endl;
        cout<<"\t\t     Enter 4 for Generate hash"<<endl;
        cout<<"\t\t     Enter 0 for Main Menu"<<endl;



        cout<<"\nEnter Your choice >>>";   cin>>ch;
        switch(ch){
case 1:
        system("cls");
   display();
   break;
case 2:
          system("cls");
     cout<<"|============CUSTOMERS NAMES LIST==============|"<<endl;
          displayQueue();
           break;
case 3:
           system("cls");
          cout<<"|============CUSTOMERS NAMES
LIST==============|"<<endl;
          dequeue();
     displayQueue();
        break;
case 4:
```

```
            int x,n;
           cout << "Enter element to generate hash = ";
                        cin >> x;  cout<<"Of total list number : "; cin>>n;
                        cout << "Hash of " << x << " is = " << hash(x,n );

                        break;



default: system("cls");


          }
}
while(ch!=0)  ;
}


////////////////////////////////////////////////////////////////////////////////

int main()
{
```

```
 for(int i=0;i<=51;i++)
 {

      push(i);
       }



                system("color 0C");     //  for console color
      gotoxy(17,5);
      cout<<"--------------------------------------------------"<<endl;
      gotoxy(17,7);
      cout<<"||            Super Market Project             ||"<<endl;
      gotoxy(17,9);
      cout<<"--------------------------------------------------"<<endl;
      gotoxy(17,11);
      cout<<"|Subject Teacher ->> Gayathri.P MAM <<-- |\n"<<endl;
      gotoxy(17,13);
      cout<<">>>----Project Implemented By-----<<<"<<endl;
gotoxy(22,15);
      cout<<"JAGIRDAR ROHIT (19BCE0763)"<<endl;
      gotoxy(22,16);
      cout<<"KANDULA MONA REDDY  (19BCE0814)"<<endl;
      gotoxy(22,17);
      cout<<"KANDRA KSHEERAJ   (19BCE0829)"<<endl;
```

```
        gotoxy(22,18);
        cout<<"LIKHITHA MODUGULA  (19BCT0032)"<<endl;
        gotoxy(22,19);
        cout<<"KOTHA BRINDA VIVEK  (19BDS0070)"<<endl<<endl;
        system("pause");
        system("cls");
        system("color Fc");
        ///////////////////////////
        int ps,profit=0;
        cout<<"\n\t\t| Super Market Login |\n";
        cout<<"\n\t\tEnter Password: ";
        cin>>ps;
        if(ps==1161)
        {
                cout<<"\t\tWelcome \n\n";

        ///////////////////////////
    int ch;

        while(ch!=6){

                        //          choice for below message

    cout<<"\n\t\t|--------<Main Menu>-----------|";
```

```
cout<<"\n\n";
    cout<<"\t\t 1)Stock                    \n";
    cout<<"\t\t 2)Staff details           \n";
    cout<<"\t\t 3)Customer                \n";
    cout<<"\t\t 4)Membership Customer    \n";
    cout<<"\t\t 5)Admin                   \n";
    cout<<"\t\t 6)Exit                   \n";




    cout<<"\nEnter Your choice >>>";cin>>ch;
    switch(ch){
    case 1:

            stock();

            break;

    case 2:

            {
    ofstream fout;
            string line;
            fout.open("Staff_Data.txt");
```

```
            while (fout) {
                        getline(cin,line);
                        if (line == "-1")
                        break;
                        fout << line << endl;
            }
                        fout.close();
                        ifstream fin;
//                      fin.open("Data.txt");

//          while (fin) {
//                      getline(fin, line);
//                      cout << line << endl;
//          }

//                      fin.close();
//                      return(0);
}

    break;

case 3:
            cout<<endl<<endl;
```

```
            bpop();
            system("pause");
        buy();
        break;



    case 4:


            int choice;
            while(choice!=2){
                    cout<<"\t\t 1)Existing Member                    \n";
                    cout<<"\t\t 2)New Member Register              \n";


                            cout<<"\nEnter Your choice >>>";cin>>choice;
            switch(choice){
            case 1:
                    membership();
                    break;
            case 2:
                    newmembership();
                    break;
            }
            break;
    }
```

```
            break;
    case 5:
            administator();
            break;



     case 6:
            cout<<"\n\n\t\t\t\t\tThank You\t\t\t\t";
            break;


            }


}
return 0;



}
            else{
            cout<<"\t\tWrong password \n\n";
            }
}
```

Results and Discussion (Working and Outputs of code):

As soon as we run the code we get the details of this project and further we can continue into working by entering any key.

```
C:\Users\ACER\Documents\projectdsa.exe

                ------------------------------------------------
                ||            Super Market Project            ||
                ------------------------------------------------
                |Subject Teacher ->> Gayathri.P MAM <<-- |
                >>>----Project Implemented By-----<<<
                     JAGIRDAR ROHIT (19BCE0763)
                     KANDULA MONA REDDY  (19BCE0814)
                     KANDRA KSHEERAJ   (19BCE0829)
                     LIKHITHA MODUGULA  (19BCT0032)
                     KOTHA BRINDA VIVEK  (19BDS0070)
Press any key to continue . . .
```

To secure the details we set a password to verify that only eligible person can access it.

```
C:\Users\ACER\Documents\projectdsa.exe

                |       Super Market Login      |
                Enter Password: 1161
                Welcome

                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>
```

If a non-eligible person tries to access the program automatically terminate.

```
C:\Users\ACER\Documents\projectdsa.exe

                |       Super Market Login      |

                Enter Password: 1056
                Wrong password


Process returned 0 (0x0)   execution time : 15.660 s
Press any key to continue.
```

MODULE 1(STOCK):

If we select option 1 then it leads us to stock module operations.

```
C:\Users\ACER\Documents\projectdsa.exe
                ============================================
                |                Stock                     |
                ============================================
                     Enter 1 for ADD a new product
                     Enter 2 to display all products
                     Enter 3 for MODIFY Existing product
                     Enter 4 for Delete a particular product item
                     Enter 0 for Main Menu

Enter Your choice >>>
```

Addition of products to stock.

```
C:\Users\ACER\Documents\projectdsa.exe
                        Enter product ID:-1
                        Enter product Name:-TEA
                        Enter product price:-150
                        Enter product quantity:-100
```

If the product is inserted then it prints “product is inserted” message. It again shows all operations in stock module.

```
C:\Users\ACER\Documents\projectdsa.exe

                    This product is Inserted!


          ==============================================
          |                  Stock                     |
          ==============================================
               Enter 1 for ADD a new product
               Enter 2 to display all products
               Enter 3 for MODIFY Existing product
               Enter 4 for Delete a particular product item
               Enter 0 for Main Menu

Enter Your choice >>>
```

Addition of other five products into the stock to perform other operations in stock and other modules.

```
C:\Users\ACER\Documents\projectdsa.exe
                    Enter product ID:-2
                    Enter product Name:-SUNSILK
                    Enter product price:-180
                    Enter product quantity:-100
```

```
C:\Users\ACER\Documents\projectdsa.exe
                   Enter product ID:-3
                   Enter product Name:-HORLICKS
                   Enter product price:-420
                   Enter product quantity:-100
```

```
C:\Users\ACER\Documents\projectdsa.exe
                    Enter product ID:-4
                    Enter product Name:-ARIEL
                    Enter product price:-250
                    Enter product quantity:-100
```

```
C:\Users\ACER\Documents\projectdsa.exe
                    Enter product ID:-5
                    Enter product Name:-DAAWAT
                    Enter product price:-220
                    Enter product quantity:-100
```

```
C:\Users\ACER\Documents\projectdsa.exe
                    Enter product ID:-6
                    Enter product Name:-FORTUNE
                    Enter product price:-130
                    Enter product quantity:-100
```

All the products that were inserted in the stock can be retrieved and displayed by this operation.

```
C:\Users\ACER\Documents\projectdsa.exe
Existing products are:
ID              Product Name            Price           Quantity
================================================================|
1               TEA                     150                     100
2               SUNSILK                 180                     100
3               HORLICKS                        420                     100
4               ARIEL                   250                     100
5               DAAWAT                  220                     100
6               FORTUNE                 130                     100

Total products in our store is : 6


                ==============================================
                |                Stock                       |
                ==============================================
                     Enter 1 for ADD a new product
                     Enter 2 to display all products
                     Enter 3 for MODIFY Existing product
                     Enter 4 for Delete a particular product item
                     Enter 0 for Main Menu

Enter Your choice >>>
```

Modification of any details of the inserted product can be done by this operation.

```
C:\Users\ACER\Documents\projectdsa.exe
Existing products are:
ID              Product Name            Price           Quantity
=================================================================|
1               TEA                     150                     100
2               SUNSILK                 180                     100
3               HORLICKS                        420                     100
4               ARIEL                   250                     100
5               DAAWAT                  220                     100
6               FORTUNE                 130                     100

Total products in our store is : 6


Old ID : 3

Old Name : HORLICKS

Old Price : 420

Old Quantity : 100


Enter new ID:8
Enter new Name:HORLICKS(MALT)
Enter new Price:400
Enter new Quantity:120
```

Modifications can be observed and verified by again displaying products in the stock.

```
C:\Users\ACER\Documents\projectdsa.exe
Existing products are:
ID              Product Name            Price           Quantity
=================================================================|
1               TEA                     150                     100
2               SUNSILK                 180                     100
8               HORLICKS(MALT)                  400                     120
4               ARIEL                   250                     100
5               DAAWAT                  220                     100
6               FORTUNE                 130                     100

Total products in our store is : 6


                ============================================
                |                Stock                     |
                ============================================
                     Enter 1 for ADD a new product
                     Enter 2 to display all products
                     Enter 3 for MODIFY Existing product
                     Enter 4 for Delete a particular product item
                     Enter 0 for Main Menu

Enter Your choice >>>
```

If you have to remove any product from the stock, deletion operation can be used. If we have to delete horlicks(malt) from the stock then we have to enter this product's id(8)

```
C:\Users\ACER\Documents\projectdsa.exe
Existing products are:
ID              Product Name            Price           Quantity
=================================================================|
1               TEA                     150                     100
2               SUNSILK                 180                     100
8               HORLICKS(MALT)                  400                     120
4               ARIEL                   250                     100
5               DAAWAT                  220                     100
6               FORTUNE                 130                     100

Total products in our store is : 6



Enter ID to delete that product:

8
```

Item deleted

```
C:\Users\ACER\Documents\projectdsa.exe

<<item is deleted>>

-------Product is Deleted-------
                ============================================
                |                Stock                     |
                ============================================
                     Enter 1 for ADD a new product
                     Enter 2 to display all products
                     Enter 3 for MODIFY Existing product
                     Enter 4 for Delete a particular product item
                     Enter 0 for Main Menu

Enter Your choice >>>
```

To return to main menu enter 0.

MODULE 2(STAFF DETAILS STORAGE BY USING FILES):

At the beginning of the day all the staff details such as empid, in time can be stored in a file named Staff_Data to maintain records of all employees in the supermarket.

If any others should be added (later) then we can add it later but before that we should save the previous details in any other file.

To end the storage details enter -1 in a new line.

```
C:\Users\ACER\Documents\projectdsa.exe

                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>2
krishna  234  8:25
tulsi    345  8:25
aakash   123  8:26
jodha    187  8:26
arjun    237  8:26
rukmini  90   8:26
jahnavi  567  8:27
tejaswi  341  8:27
likhitha 231  8:28
japneet  78   8:30
-1

                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>
```

Open file named Staff_Data in the folder where code is present

C:\Users\ACER\Documents\Staff_Data.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

change.log | sample.html | proj.h | recent_customer.txt | Staff_Data.txt

```
krishna  234  8:25
tulsi    345  8:25
aakash   123  8:26
jodha    187  8:26
arjun    237  8:26
rukmini  90   8:26
jahnavi  567  8:27
tejaswi  341  8:27
likhitha 231  8:28
japneet  78   8:30
```

MODULE 3(CUSTOMER):

First a trolley is allocated to the customer by using stack.

C:\Users\ACER\Documents\projectdsa.exe

```
                1)Stock
                2)Staff details
                3)Customer
                4)Membership Customer
                5)Admin
                6)Exit

Enter Your choice >>>2
krishna  234  8:25
tulsi    345  8:25
aakash   123  8:26
jodha    187  8:26
arjun    237  8:26
rukmini  90   8:26
jahnavi  567  8:27
tejaswi  341  8:27
likhitha 231  8:28
japneet  78   8:30
-1

                |--------<Main Menu>-----------|

                1)Stock
                2)Staff details
                3)Customer
                4)Membership Customer
                5)Admin
                6)Exit

Enter Your choice >>>3


Your Trolli No is :50
                 ___
                /  |
  _____________/   --
 |___/__ /___/_|
 |__/___/___/__|
 |_/___/___/___|
    ______/
     O   O
Press any key to continue . . .
```

Enter no. of products to be bought, id's of that product, customer name.

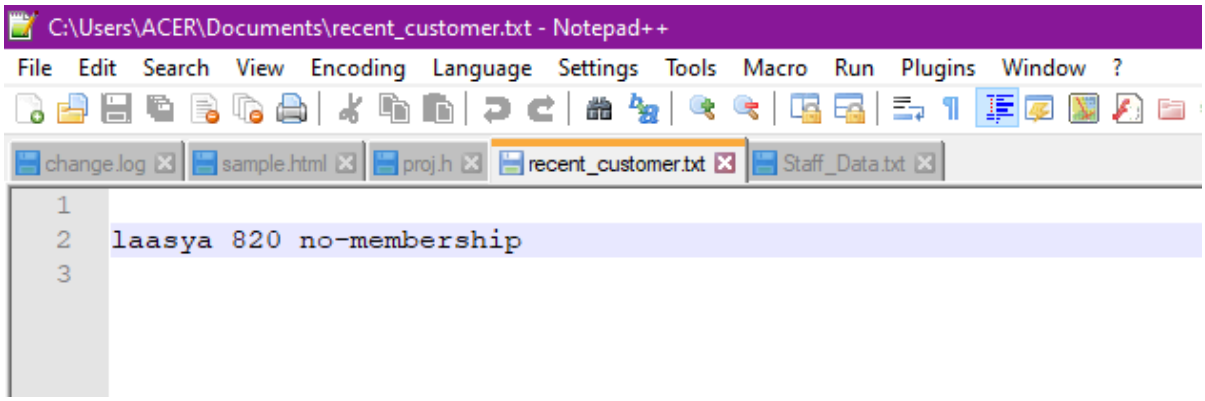
If id is not valid it shows that product unavailable.

```
C:\Users\ACER\Documents\projectdsa.exe
Existing products are:
ID              Product Name            Price           Quantity
=================================================================|
1               TEA                     150                     100
2               SUNSILK                 180                     100
4               ARIEL                   250                     100
5               DAAWAT                  220                     100
6               FORTUNE                 130                     100

Total products in our store is : 5


Enter id of item that you want to buy: 1
How many quantities you want:1
Enter id of item that you want to buy: 2
How many quantities you want:3
Enter id of item that you want to buy: 3

<<<<<<<<<This item is not available in our store at this time>>>>

Enter id of item that you want to buy: 6
How many quantities you want:1

                 Enter Your Name :laasya_
```

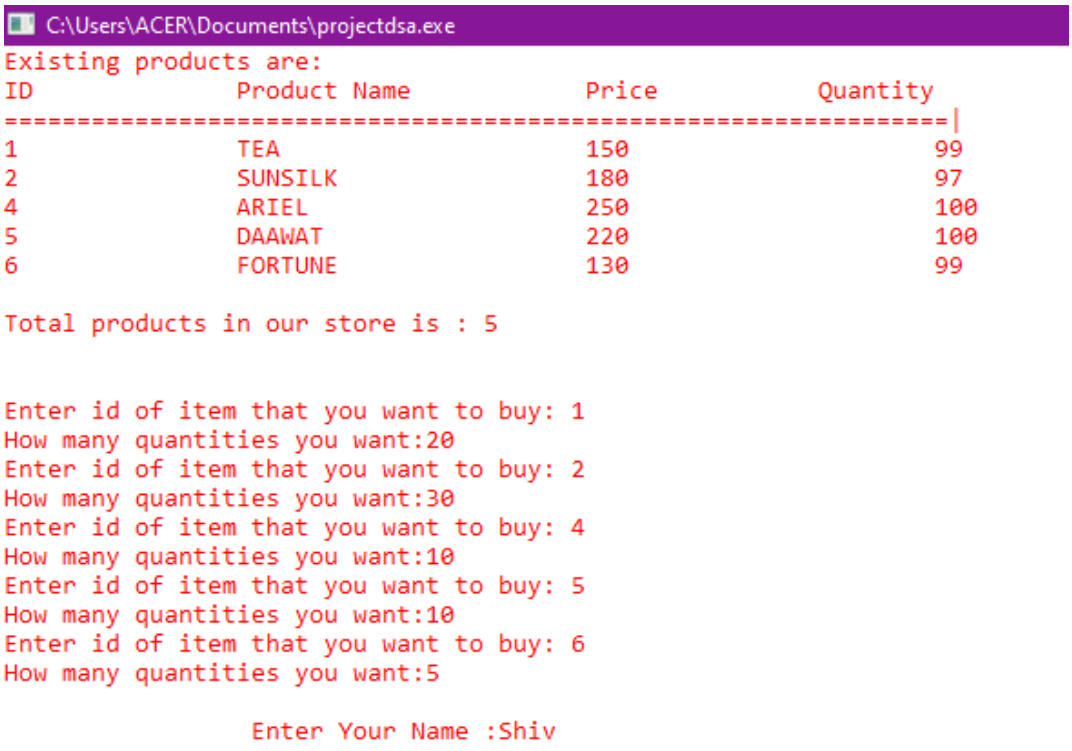
Final amount and enter customer details:

```
C:\Users\ACER\Documents\projectdsa.exe



                       You have bought : TEA SUNSILK FORTUNE
 Bill Amount: 820
Thank you for shopping !

laasya 820 no-membership
-1

                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>
```

Open the file recent_customer :



Customer who purchases more than 3000



Discount of 10%

```
C:\Users\ACER\Documents\projectdsa.exe



                        You have bought : TEA SUNSILK ARIEL DAAWAT FORTUNE

Original price : 13750
 with 10% discount: 12375
Thank you for shopping !

shiv 12375 non-membership
-1
```

MODULE 4(MEMBERSHIP CUSTOMER):

Case 1(existing membership card)

Will get a compulsory discount of 20%

```
C:\Users\ACER\Documents\projectdsa.exe



                        You have bought : TEA SUNSILK ARIEL DAAWAT FORTUNE

Original price : 13750
 with 10% discount: 12375
Thank you for shopping !

shiv 12375 non-membership
-1

                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>4
                 1)Existing Member
                 2)New Member Register

Enter Your choice >>>1
```

Name input

```
C:\Users\ACER\Documents\projectdsa.exe

                        You have bought : TEA SUNSILK ARIEL DAAWAT FORTUNE

Original price : 13750
 with 10% discount: 12375
Thank you for shopping !

shiv 12375 non-membership
-1

                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>4
                 1)Existing Member
                 2)New Member Register

Enter Your choice >>>1

                 Enter Your Name :pooja
```

Purchasing less than 3000 will not have any gift coupons

```
C:\Users\ACER\Documents\projectdsa.exe

                        You have bought : DAAWAT

Original price : 660
 with 20% discount: 528
 shop more to win gift coupons
Thank you for shopping !

                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>
```

```
C:\Users\ACER\Documents\projectdsa.exe

                        You have bought : DAAWAT

Original price : 660
 with 20% discount: 528
 shop more to win gift coupons
Thank you for shopping !


                 |--------<Main Menu>-----------|

                  1)Stock
                  2)Staff details
                  3)Customer
                  4)Membership Customer
                  5)Admin
                  6)Exit

Enter Your choice >>>4
                  1)Existing Member
                  2)New Member Register

Enter Your choice >>>1

                  Enter Your Name :lalitya_
```

```
C:\Users\ACER\Documents\projectdsa.exe
Existing products are:
ID              Product Name            Price           Quantity
=================================================================|
1               TEA                     150                     79
2               SUNSILK                 180                     67
4               ARIEL                   250                     90
5               DAAWAT                  220                     87
6               FORTUNE                 130                     94

Total products in our store is : 5


Enter id of item that you want to buy: 1
How many quantities you want:20
Enter id of item that you want to buy: 2
How many quantities you want:20
Enter id of item that you want to buy: 3

<<<<<<<<<This item is not available in our store at this time>>>>

Enter id of item that you want to buy: 4
How many quantities you want:40
Enter id of item that you want to buy: 6
How many quantities you want:10
```

Shopping amount >3000 we get gift coupons of worth 500 Rs.

```
C:\Users\ACER\Documents\projectdsa.exe



                        You have bought : TEA SUNSILK ARIEL FORTUNE

Original price : 17900
 with 20% discount: 14320
 You have won coupon worth rupees 500
Thank you for shopping !


                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>
```

Case 2(new membership card):

```
C:\Users\ACER\Documents\projectdsa.exe



                        You have bought : TEA SUNSILK ARIEL FORTUNE

Original price : 17900
 with 20% discount: 14320
 You have won coupon worth rupees 500
Thank you for shopping !


                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>4
                 1)Existing Member
                 2)New Member Register

Enter Your choice >>>2

                 Enter Your Name :amal
```

```
C:\Users\ACER\Documents\projectdsa.exe
Existing products are:
ID              Product Name            Price           Quantity
=================================================================|
1               TEA                     150                     59
2               SUNSILK                 180                     47
4               ARIEL                   250                     50
5               DAAWAT                  220                     87
6               FORTUNE                 130                     84

Total products in our store is : 5


Enter id of item that you want to buy: 4
How many quantities you want:5
Enter id of item that you want to buy: 6
How many quantities you want:30
```

Bill to be paid by new membership card holder is amount purchased+1000.

Discounts and coupons will be given for their next shopping.

```
C:\Users\ACER\Documents\projectdsa.exe



                        You have bought : ARIEL FORTUNE

Original price : 5150
 with purchase of membership card: 6150
 You will get membership discount from next time
Thank you for shopping !


                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>
```

MODULE 5(ADMIN):

Admin portal opens as soon as we select this module.

```
C:\Users\ACER\Documents\projectdsa.exe
                    ==============================================
                    |          Administator Portal               |
                    ==============================================
                         Enter 1 to display all products
                         Enter 2 for Customers List
                         Enter 3 for Dequeue customer
                         Enter 4 for Generate hash
                         Enter 0 for Main Menu

Enter Your choice >>>
```

Display of all the products in the stock , because admin should know so as to add items if they are out of stock.

```
C:\Users\ACER\Documents\projectdsa.exe
Existing products are:
ID              Product Name            Price           Quantity
==============================================================|
1               TEA                     150                     59
2               SUNSILK                 180                     47
4               ARIEL                   250                     45
5               DAAWAT                  220                     87
6               FORTUNE                 130                     54

Total products in our store is : 5


                ==============================================
                |          Administator Portal               |
                ==============================================
                     Enter 1 to display all products
                     Enter 2 for Customers List
                     Enter 3 for Dequeue customer
                     Enter 4 for Generate hash
                     Enter 0 for Main Menu

Enter Your choice >>>
```

Customer’s list:

```
C:\Users\ACER\Documents\projectdsa.exe
|============CUSTOMERS NAMES LIST==============|
        laasya
        Shiv
        pooja
        lalitya
        amal
                ============================================
                |          Administator Portal             |
                ============================================
                     Enter 1 to display all products
                     Enter 2 for Customers List
                     Enter 3 for Dequeue customer
                     Enter 4 for Generate hash
                     Enter 0 for Main Menu

Enter Your choice >>>
```

Dequeue any customer from the line (waiting)

```
C:\Users\ACER\Documents\projectdsa.exe
|============CUSTOMERS NAMES LIST==============|
        Shiv
        pooja
        lalitya
        amal
                ============================================
                |          Administator Portal             |
                ============================================
                     Enter 1 to display all products
                     Enter 2 for Customers List
                     Enter 3 for Dequeue customer
                     Enter 4 for Generate hash
                     Enter 0 for Main Menu

Enter Your choice >>>
```

Hash function to make sure no crowding occurs at the counter.

```
C:\Users\ACER\Documents\projectdsa.exe
|============CUSTOMERS NAMES LIST==============|
        Shiv
        pooja
        lalitya
        amal
                ============================================
                |          Administator Portal             |
                ============================================
                     Enter 1 to display all products
                     Enter 2 for Customers List
                     Enter 3 for Dequeue customer
                     Enter 4 for Generate hash
                     Enter 0 for Main Menu

Enter Your choice >>>4
Enter element to generate hash = 1
Of total list number : 10
Hash of 1 is = 1                ============================================
                |          Administator Portal             |
                ============================================
                     Enter 1 to display all products
                     Enter 2 for Customers List
                     Enter 3 for Dequeue customer
                     Enter 4 for Generate hash
                     Enter 0 for Main Menu

Enter Your choice >>>
```

Enter 6 to exit from this stop this program.

```
C:\Users\ACER\Documents\projectdsa.exe

                |--------<Main Menu>-----------|

                 1)Stock
                 2)Staff details
                 3)Customer
                 4)Membership Customer
                 5)Admin
                 6)Exit

Enter Your choice >>>6


                                        Thank You
Process returned 0 (0x0)   execution time : 6122.053 s
Press any key to continue.
```

Time complexity

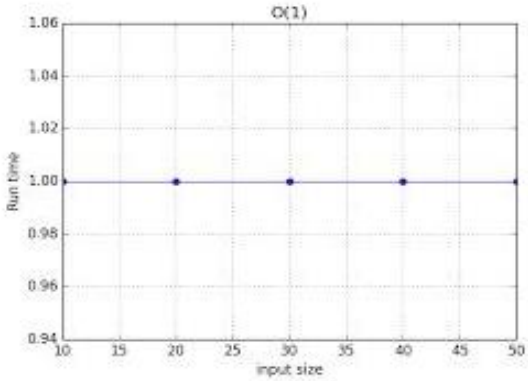
Time complexity is a very important aspect. Individuals prefer faster services due to the rapid development of technology. This demands faster operations from this program compared to other programs which were developed for the same purpose. Hence, time complexity has been calculated and the same has been explained below.

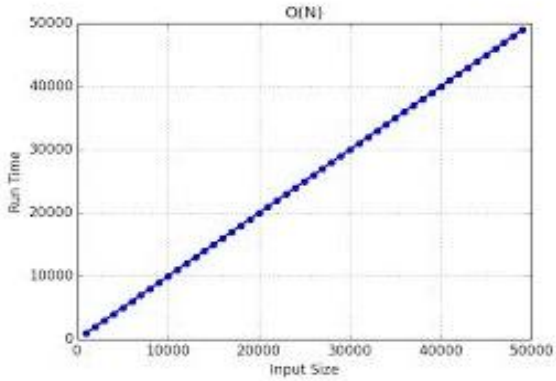
Stack

The worst case of the insertion (trolley allocation) and deletion (deleting the trolley allocated after the customer has shopped) operations of stacks is O(1).

Linked list

The worst case of the insertion and deletion operations of linked lists is O(1). The search operation is O(n).



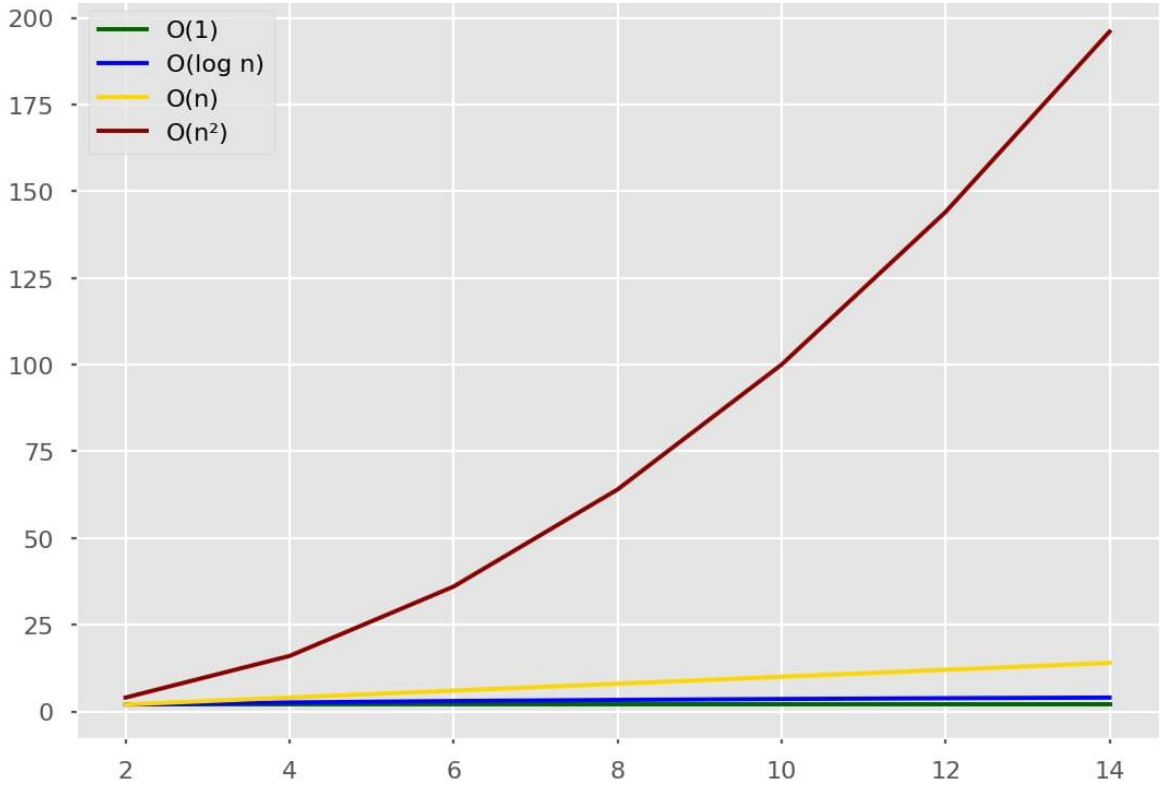


Array

The worst case of the insertion and deletion operations of array is O(n). The search operation is O(n).

Queues

The worst case of the insertion and deletion operations of queue is O(1). The search operation is O(n).



Our Words:

At the end of the project we have practically implemented the data structures that we have learnt in the class and also tried learning and implementing new data structure (Files). First, we referred to different articles and online resources to gain knowledge about the working of this system and tried to extend our knowledge to implement using data structures. We got to know the problems faced there and tried to emulate the real time working of it in efficient ways. We learnt animations like changing colour of output terminal, colour of text, changing the cursor positions and cursor to underscore(_).

Github Link for the project:

https://github.com/ksheeraj1161/Super-Market-Billing-system

References

1) Sara Baase, Allen Van Gelder.(1999).Computer Algorithms, Introduction to Design and Analysis,3rd edition.Wesley Longman Publishing.
2) Sahni, Sartaj. (2005).Data Structures, Algorithms and Applications in C++,2nd ed. Universities Press
3) Thomas, H & Cormen Charles, E & Leiserson Ronald,L&Rivest Clifford, Stein.(May 2001). Introduction_ to_ Algorithm. The MIT Press, McGrawHill Book Company.
4) Reema,Thareja.(2014).Data structures using C,2nd edition. Oxford University Press.
5) Ellis,Horowitz.(1983). Fundamentals of Data Structures in C,2nd Edition.Computer Science Press.
6) Narasimha, Karumanchi .(2014).Data Structures and Algorithms Made Easy,2nd ed.Atlantic Publishors and Distributers.
7) Wegner, Peter; Reilly, Edwin D. (2003-08-29). Encyclopedia of Computer Science. Chichester, UK: John Wiley and Sons

8) Retrieved from https://www.geeksforgeeks.org/pointers-andreferences-in-c/
9) Krishnamoorthy, R., & Kumaravel, G. I. (2010). Data structures using C. New Delhi: Tata Mcgraw Hill education Private.
10) Lipschutz, S. (101-). Data structures (SOS) (Revised first edition). McGraw-Hill Education.
11) Pointers in C and C++. (2020, July 14). JournalDev. Retrieved from : https://www.journaldev.com/30481/pointers-in-c-and-c-plus-plus
12) Basics of Hash Tables Tutorials & Notes: Data Structures. Retrieved from:https://www.hackerearth.com/practice/data-structures/hashtables/basics-of-hash-tables/tutorial/
13) Hash functions. Retrieved from: https://algs4.cs.princeton.edu/34hash/
14) Frank , Steeneken & Dave ,Ackley. (January 2012). A Complete Model of Supermarket Business.Retrieved from https://www.bptrends.com/publicationfiles/01-03-2012-ARTSupermarket%20Article-steeneken-Ackley%20111226.pdf.BP Trends.
15) Vanessa Cross.(September 26, 2017). How to Run a Supermarket Business.Retrieved from https://bizfluent.com/how-6808955-runsupermarket-business.html.