Super Market Management System

A Project Report submitted for
Data structures and algorithms (UCS - 613)
(Third Year)

Submitted by

Prathit Malik	**(Regd. No. : 101506128)**
Pulkit Bahl	**(Regd. No. : 101506131)**
Rahul Garg	**(Regd. No. : 101506133)**
Saket Panchori	**(Regd. No. : 101506145)**

Group: ECE-7

Supervisor

Dr. Sumit Miglani
Assistant Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUE OF ENGINEERING AND TECHNOLOGY, PATIALA-147004
INDIA

May 2018

Problem Formulation

In this system there are many type of inputs processed into it. All type of inputs is stored into projects for various purposes. The main problems are:-

- Product details like (Product name, Product price, Product Quantity)
- Customer's purchase details
- Products expiry date
- Bill records of customers
- Detailed information about stock
- Addition of new items
- Modify existing product
- Delete items from the inventory

Analysis Of Problem

The operations in supermarket management system includes the procurement of goods, inspection, warehousing, returns, replenishment, sales, promotion, price management and inventory gains and losses.

During sales process, based on inventory gains and losses and the demand custom promotions, the system goes through replenishment and price management, and generates a notice of assessment into the process again. Mark return merchandise and replenishment goods into the return / replenishment process, storing the chargeback track record in the historical database, those which does not complete successfully will re-enter the returns or reported loss treatment and track workflow.

Data Structures and algorithm techniques

The type of data structure used in this project is **LINKED LIST** structure along with some concepts of file handling. The following functions were performed on linked list that was created as a database consisting of details of various products held at a supermarket:

1.Insertion-The process of insertion consists of three parts:

a) insertion at beginning

b) insertion in between

c) insertion at the end

2.Deletion-The process of deletion also consists of three parts:

a) deletion at beginning

b) deletion in between

c) deletion at end

3.Searching-Searching or traversing is done to explore the contents of list in which there are also three cases

a) Best case-O(1)

b) Average case-O(n/2)

c) Worst case-O(n)

Linked list was used because of its numerous advantages over the arrays including:

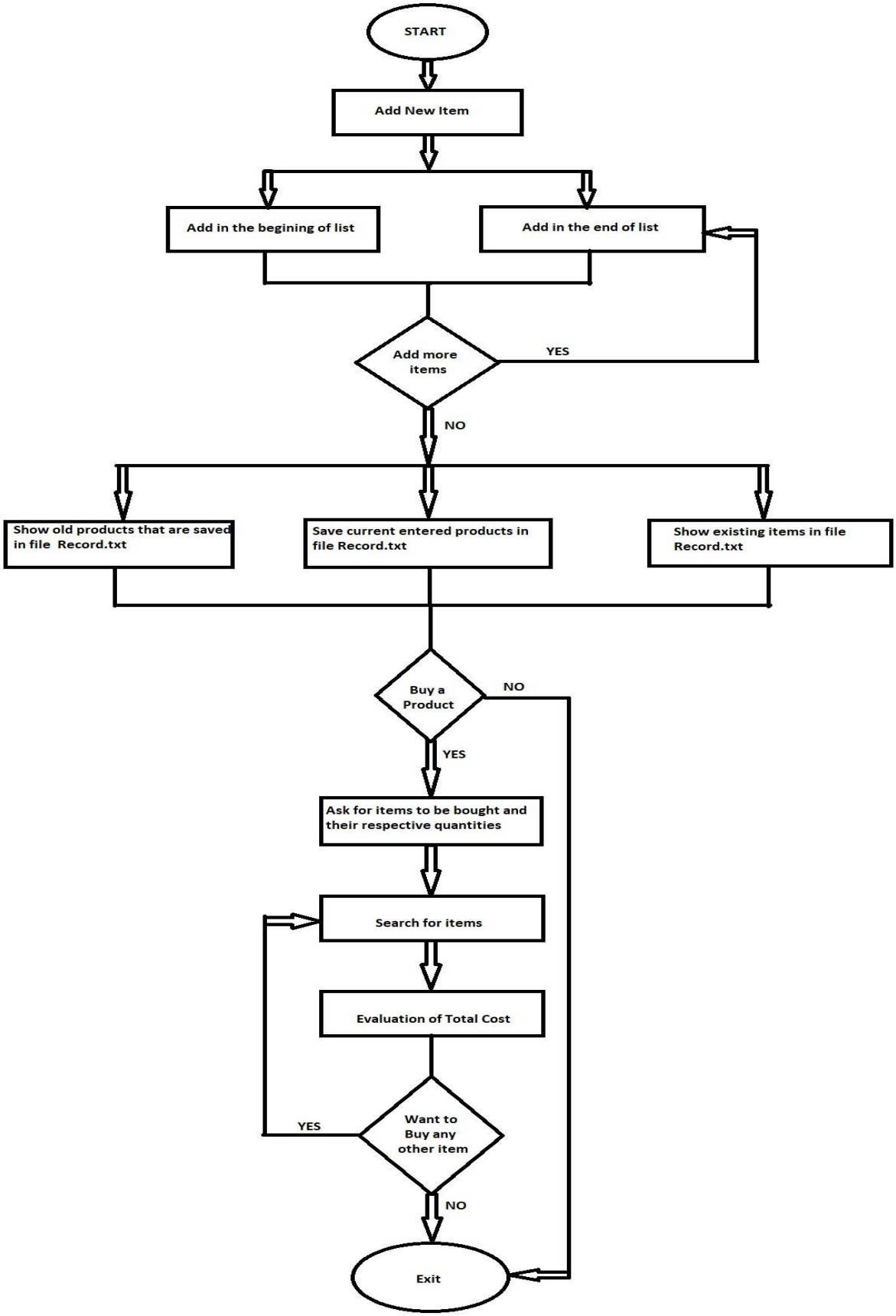
1. Inserting a new element in an array of elements is expensive, because room has to be created for the new elements and to create room existing elements have to shifted.

2. The size of the arrays is fixed: So we must know the upper limit on the number of elements in advance. Also, generally, the allocated memory is equal to the upper limit irrespective of the usage, and in practical uses, upper limit is rarely reached.

3. Deletion is also expensive with arrays until unless some special techniques are used. For example, to delete 1010 in id[], everything after 1010 has to be moved.

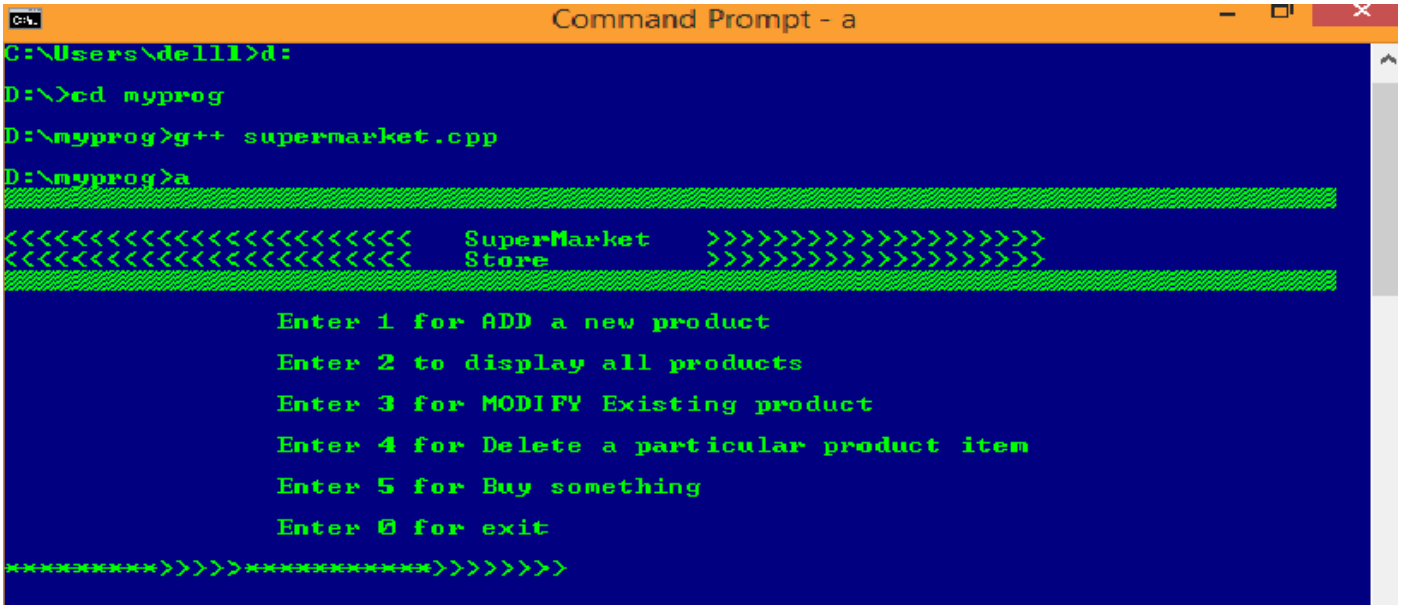
So summarizing all the above points in general linked list provides following two advantages over arrays:
1)Dynamicsize
2) Ease of insertion/deletion

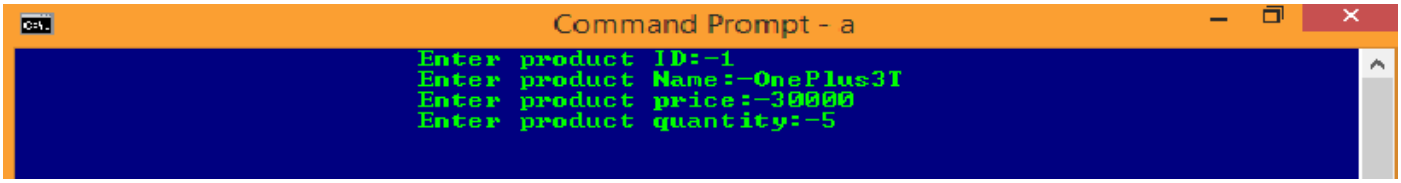
Methodology in form of Flowchart

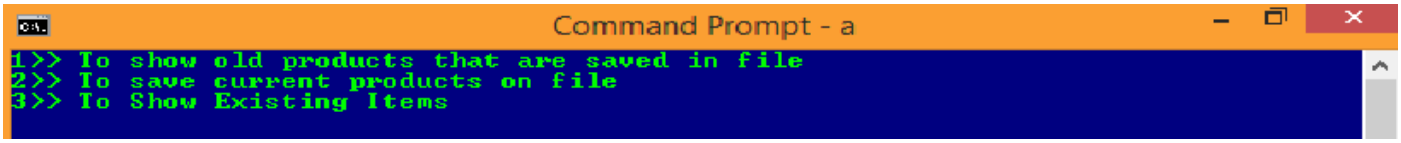


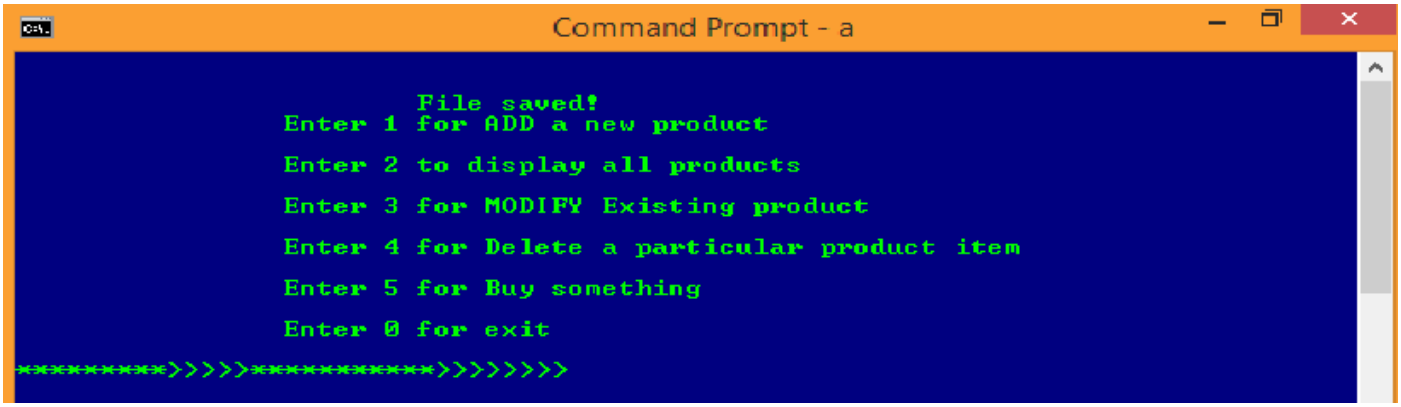
Result

The project was compiled and executed.The following outputs were obtained:

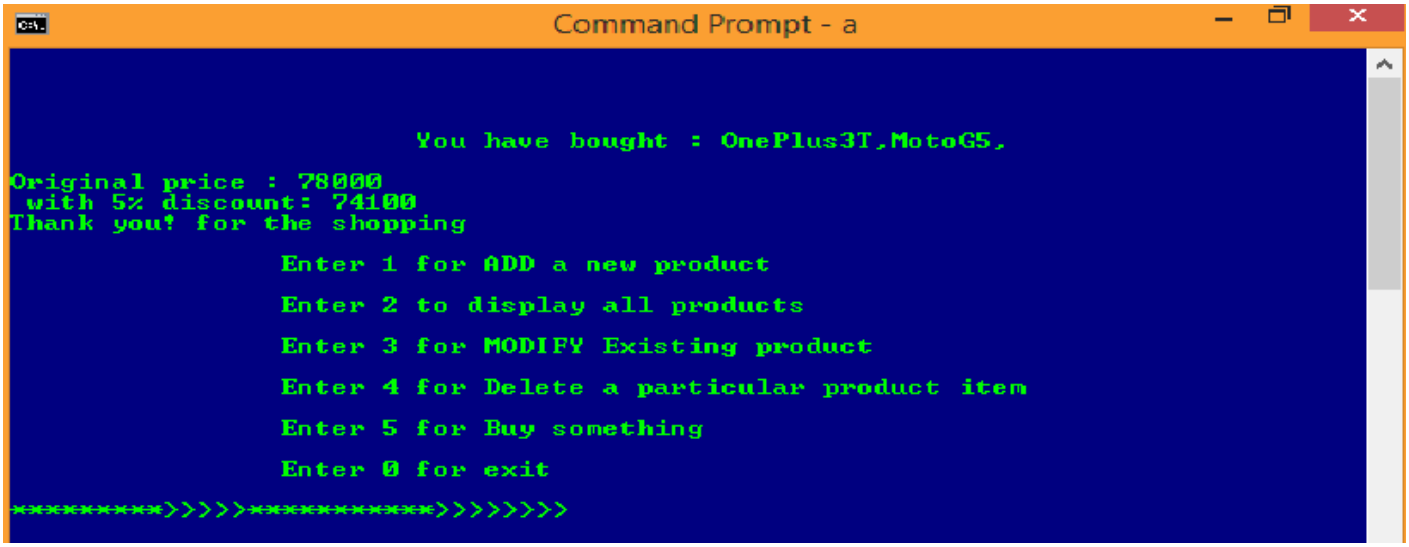









Command Prompt - a
Existing products are:
ID Product Name Price Quantity
1 OnePlus3T 30000 5
2 MotoG5 18000 3
3 samsungJ6 12000 6
Total products in our store is : 3
Enter 1 for ADD a new product
Enter 2 to display all products
Enter 3 for MODIFY Existing product
Enter 4 for Delete a particular product item
Enter 5 for Buy something
Enter 0 for exit


Command Prompt - a
You have bought : OnePlus3T,MotoG5,
Original price : 78000
with 5% discount: 74100
Thank you! for the shopping
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 5 for Buy something
Enter 0 for exit

References

1.Data structure,algorithms and applications in C++ by Sartaaj Sahni

2.Data Structure by Seymour Lipschutz

3.Object oriented programming with C++ by E Balagurusamy

CODE

```
#include<iostream>
#include<string>
#include <fstream>
#include <sstream>
//   declarations of functions
using namespace std;
int search(int);
int display();
string check(int);       //   for checking quantity
 //   build a node
struct node {
      int ID;
      string proName;
      double prePrice;            //   product price
      int quantity;
struct    node* next;
};
struct node *head=NULL;
//      for entering(saving) 1st record in list
 void beg()
{
      system("cls");
      int id,quant;           //  quant    for quantity
      string name;
      double pre;            //  pre for price
      struct node *t=new node;
      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;
      t->next=head;
      head=t;
      system("cls");
            cout<<"\n\n\t\t\t\tThis product is Inserted!\n\n\n";
      }
// for entering(saving) 2nd and onward records in list
      void end()
      {
      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;
            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";
      }
      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;       //   pName for new name
            if (head == NULL)
    {
      system("cls");
         cout<<"List is empty"<<endl;
    }else
```

```
      {
            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 Name : "<<cur->proName;
            cout<<"\nOld Sallary : "<<cur->prePrice<<endl;
            cout<<"Enter new Name:";
            cin>>pName;
            cur->proName=pName;
            cout<<"Enter new Price:";
            cin>>pre;
            cur->prePrice=pre;
      }else{
            cout<<id<<" is <<<Not found>>\n\n";
      }
      }
}
      void saveOnFile()
      {
      system("cls");

  ofstream myfile ("Products Record.txt");

  if (myfile.is_open())
  {
    myfile << "ID \t\t Product Name \t\t\t\t\t Price \t\t\t Quantity \n";
    struct node *p=head;
    int c=0;          //   c for count products
            while(p!=NULL)
            {
                  myfile<<p->ID<<"\t\t"<<p->proName<<"\t\t\t\t"<<p-
>prePrice<<"\t\t\t"<<check(p->quantity)<<"\n";
                  p=p->next;
                  c=c+1;
            }
            myfile<<"\nTotal products in our store is : "<<c;
            }
    myfile.close();
    system("cls");
    cout<<"\n\n\t\t\tFile saved!\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\t Quantity\n";
            while(p!=NULL)
            {
```

```
                cout<<p->ID<<"\t\t"<<p->proName<<"\t\t\t\t"<<p-
>prePrice<<"\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 OldProducts(){
           system("cls");
           string line;
        ifstream myfile ("Products Record.txt");
        if (myfile.is_open())
  {
    while ( getline (myfile,line) )
    {
      cout << line << '\n';
    }
    cout<<"\n\n";
    myfile.close();
  }
  else cout << "Unable to open file\n\n";
           }

     void buy(){
           system("cls");
           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 wanna 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;              //   quant   for quantity  and cho for
choice
     cout<<"Enter id of item that you want to buy: ";
int id,pos=0;
     cin>>id;
     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;
      products[c]=cur->proName; 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*(0.95);           //    with 5% discount
      cout<<"\n\nOriginal price : "<<pay;
    cout<<"\n with 5% discount: "<<price<<"\nThank you! for the shopping\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;
 }
                                 //        Main function
int main(){
            system("color 1A");      //  for console color
      cout<<"\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\x
B2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\
xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2
\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB
2\xB2\xB2\n\n";
      cout<<"<<<<<<<<<<<<<<<<<<<<<<<   SuperMarket   >>>>>>>>>>>>>>>>>>>\n";
//  \xB2  is for styling
      cout<<"<<<<<<<<<<<<<<<<<<<<<<<   Store         >>>>>>>>>>>>>>>>>>>\n";

cout<<"\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2
\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB
2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\x
B2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\
xB2\n\n";              //  on console
```

```
      int temp=1;
      while(true){
      int ch;                        //            choice for below message
      cout<<"\t\tEnter 1 for ADD a new product \n\n\t\tEnter 2 to display all
products \n\n\t\tEnter 3 for MODIFY Existing product\n\n";
      cout<<"\t\tEnter 4 for Delete a particular product item\n\n\t\tEnter 5
for Buy something\n\n\t\tEnter 0 for exit\n\n";
      cout<<"*********>>>>>>>>>>>>>>>>>>>>>>>>";   cin>>ch;
      switch(ch){
      case 1:
      if (temp==0){              //    Second time and on ward this is only
executed
      end();
      }
      if(temp==1){               //    this will be executed only one time
      beg();
      temp=0;
      }
      break;
case 2:
      system("cls");
      cout<<"1>> for show old products that are saved in file\n";
     cout<<"2>> for save current products on file\n";
     cout<<"3>> for Show Existing Items\n";
     int c; cin>>c;
     if(c==1) OldProducts();
     if(c==2) saveOnFile();
     if(c==3) display();
     break;
case 3:
      modify();
      break;
case 4:
      delPro();
      break;
case 5:
      buy();
      break;
case 0:
        cout<<"Exiting..."<<end;
          exit(true);                    //   exit from while loop
        break;
default: system("cls");
         cout<<"\t\t<<<Wrong choice>>>\n\n";
      }
}
}
```