SUPERMARKET BILLING SYSTEM PROJECT REPORT

DATA STRUCTURE AND ALGORITHM

By
Fahim Iqbal (13596)
Shayan Turk (13618)
Abdullah Asif (13587)
Usman Umar Khan (13626)
Fahad Zakir

FACULTY IN-CHARGE: Sir. Sibtul Hasan

Abbottabad University Science And Technology

June, 2024

CONTENTS

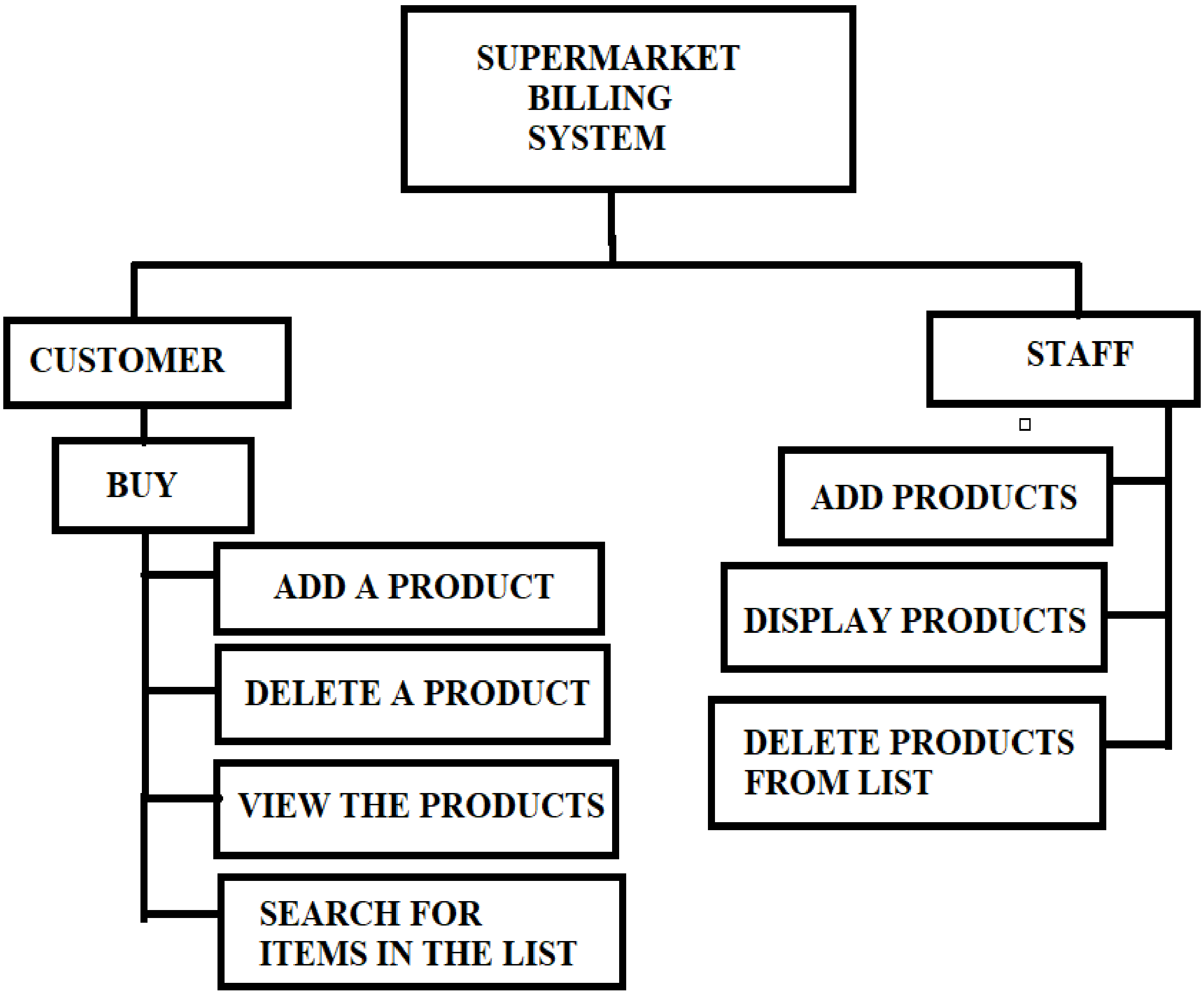
INTRODUCTION

The project is on Supermarket Billing System. Supermarket is where customers come to purchase their everyday utilizing products and pay for that. So, there is a need to calculate what number of products are sold and to create the bill for the customer. Supermarket Billing goes for forming into software that can be utilized at places like shopping Malls, calculating the bill, etc. It is additionally used to convert all the manual work which is time consuming and blunder inclined to completely mechanized system which helps in disposing of all the administrative work, save time, improves customer services. The Supermarket Billing System additionally contains discounts on different products with the goal that the item is offered at discounted price while billing It additionally accelerates different processes such as option of new things to the menu, deletion of things from the menu, modification of details of things and calculation of bills subsequently giving convenience to the workers as well as customers. In this project c++ language is utilized to keep up every one of the information. It gives numerous features like file handling.

ABSTRACT

This project aims to make software quick in preparing, with great user interface so user can transform it and it should be utilized for a very long period without mistake and maintenance. It is utilized to reduced pressure on the work and relieving manpower from repetitive and dull occupation. The primary goal of this project is to reduce the measure of paper work and furthermore to make the system progressively reliable to avoid any ambiguity and furthermore to reduce the cost factor of the system. Supermarket Billing system helps to make the system more feasible and flexible and thus recovery of information becomes too convenient. It is utilized to build the effectiveness and accuracy of the system. This software project consists of an efficient and simple GUI to help the employees in simple bill computation and giving a efficient customer service The Supermarket Billing System is worked to enable supermarkets to compute and display bills and serve the customer in a quicker and efficient way. The system also contains discounts on various products so that the product is offered at discounted price while billing.

WORK FLOW



1. The product will come in the store.
2. The Administrator will enter the information of the product in database and cost and discount accessible for every product.
3. The customer will come and take the basket with him/her and pick the product and take it to the counter.
4. The bill figuring operator will enter the product number then it will show its data and cost and the bill will be determined and complete payment will be appeared.
5. Customer will pay for the products.
6. Every one of the products will be packed and delivered to the customer.

PROBLEM STATEMENT

Supermarket requires stock control system to void out of stock level for each product. A purchasing admin will be able to process an order by entering product numbers and required quantities into the system. The system will display a description, price and available stock. In-stock products will normally be collected immediately by the Customer from the store. If stock is not available the purchasing admin product not available will be printed. Order details for in-stock products will be printed including the quantity, product number and description. The catalogue of available products will be maintained remotely.

DSA CONCEPTS AND SOFTWARES USED

DOUBLY LINKED LIST

- Doubly Linked list acts as a pillar in the development of our program. We have used its unique advantages and ease of insertion, deletion, search to create a highly efficient program.
- In case of insertion and deletion doubly linked lists are highly efficient as the time complexities are O(1) and O(1) respectively.
- In doubly linked list, the traversal can be done using the previous node link or the next node link.
- If we are at a node, then it is possible to go to any node.
- It is easy to reverse the link list.

USES OF DOUBLY LINKED LISTS:

1) In many operating systems, the thread scheduler (what chooses what processes need to keep running at which times) keeps up a doubly-linked list of the considerable number of processes running whenever. This makes it simple to move a process from one queue (say, the list of dynamic processes that need a swing to run) into another queue (say, the list of processes that are blocked and trusting that something will discharge them). The utilization of a doubly-linked list here allows every one of these splices and rewires to run in time O(1) and without any memory allocations, and the doubly-linked-list structure works well for executing the scheduler utilizing queues (where you only need to haul things out from the front.)

2) Doubly linked list is used in constructing MRU/LRU (Most/Least Recently Used) store.

3) A music player which has straightaway and previous buttons.

4) Represent a deck of cards in a diversion.

5) The browser reserve which allows you to hit the BACK-FORWARD pages.

C++-LANGUAGE

• An important advantage of C++ is its ability to extend itself. A C++ program is basically a collection of functions that are supported by the C++ library this makes it easier for us to add our own functions to C++ library. Because of the accessibility of huge number of functions, the programming task becomes easier.

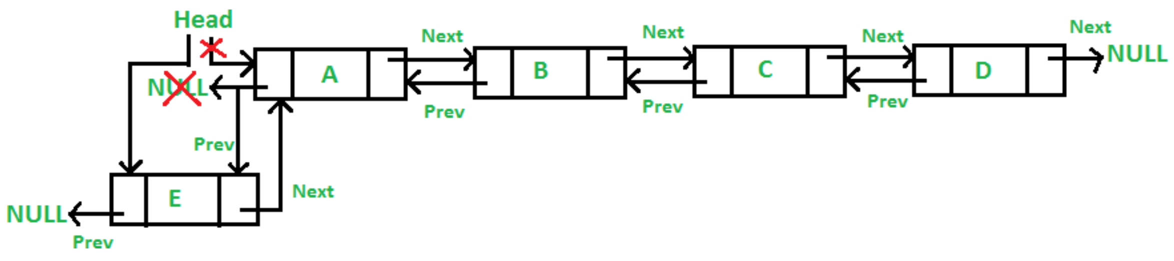
• In C++ language, it is easier for us to think of a problem in terms of function modules or blocks. We can use the collection of these modules to make a complete program. This modular structure makes program debugging, testing and maintenance simpler.

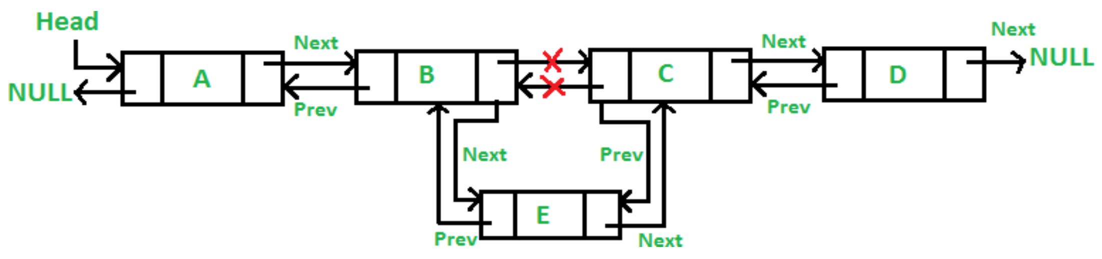
HOW DLL IS IMPLEMENTED

To conclude when manipulation operations (insert, delete, search) are frequent on data doubly linked list should be used. In a supermarket Store one thing that needs to be done frequent is to make entries of the purchased items by customers and generate their bills. So, in a Supermarket store we are implementing a billing system through the use of doubly linked list.

1. INSERT

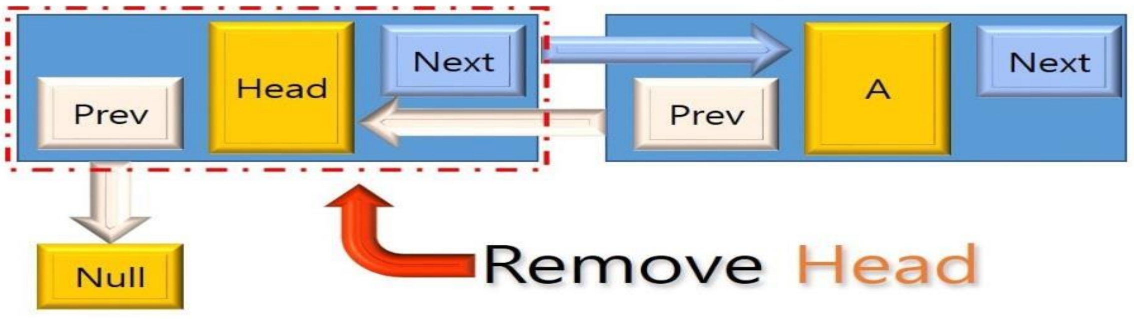
- Add Product
- Add Product at first position
- Add Product after a given position

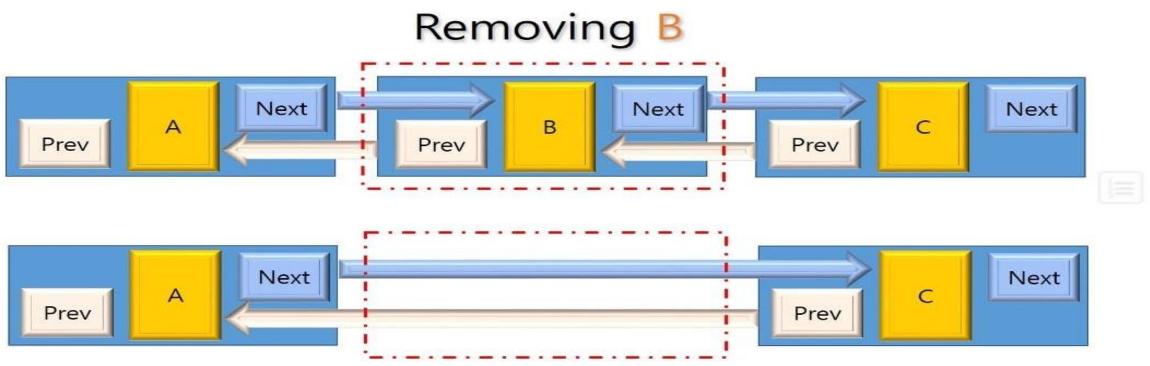


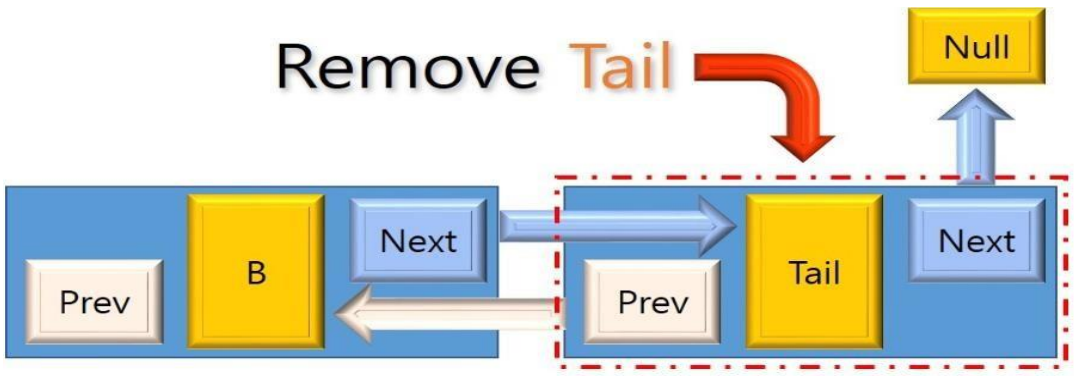


2. DELETE

- Delete Product at last position
- Delete Product at first position
- Delete Product at after a given position







3. DISPLAY

4. BUY

5. SEARC

CODE

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

string check(int);
int  display1();
int search(int);

struct node{
node  *prev;
int data;
int x,y,sum;     //x == quantity
string name,nam; //y == price
node *next;
};

struct node *start=NULL;
node* create_node(){
node *n = new node;
cout<<"ENTER PRODUCT ID: "<<endl;
cin>>n->data;
cout<<"ENTER PRODUCT NAME: "<<endl;
cin>>n->name;
cout<<"ENTER QUANTITY: "<<endl;
cin>>n->x;
cout<<"ENTER PRICE OF EACH PRODUCT: "<<endl;
cin>>n->y;
n->next = NULL;
n->prev = NULL;
return n;
}

void insert_node(){
node *temp;
temp = create_node();
if(start == NULL){
    start = temp;
}
else{
    node *traverse;
    traverse = start;
    while(traverse->next != NULL){
        traverse = traverse->next;
    }
    traverse->next=temp;
```

```
   temp->prev=traverse;
}
}

void add_begin(){
node *temp;
temp = create_node();
temp->next = start;
start->prev = temp;
start=temp;
}

void add_after(){
node *temp;
temp= create_node();
int value;
cout<<"Enter the product id after which the product has to be inserted :"<<endl;
cin>>value;
node *traverse;
traverse = start;
while(traverse->data!=value){
   traverse = traverse->next;
}
temp->prev = traverse;
temp->next=traverse->next;
traverse->next->prev = temp;
traverse->next = temp;

}

void display(){
node *traverse;
traverse = start;
system("cls");
cout<<"\n\n\t\t\t**********************ALL                                MART
STORE*****************************"<<endl;
cout<<"\nPRODUCT ID\t"<<"PRODUCT NAME\t"<<"QUANTITY\t"<<"PRICE"<<endl;
while(traverse!=NULL){
   cout<<traverse->data<<"\t\t";
   cout<<traverse->name<<"\t\t";
   cout<<traverse->x<<"\t\t";
   cout<<traverse->y<<"\n";


   traverse = traverse->next;

}
}

void delete_beg(){
```

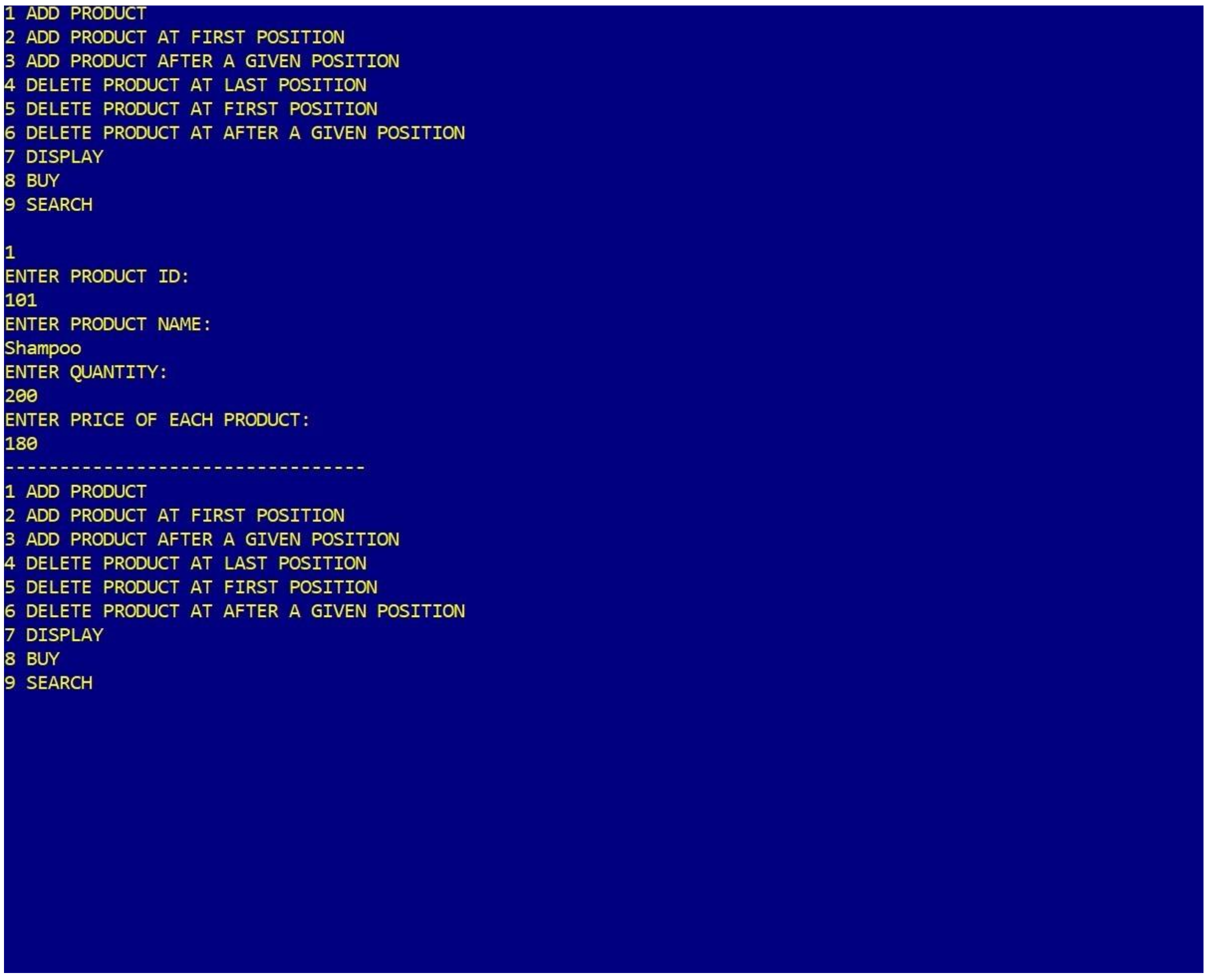
```
node *temp;
temp = start;
start = start->next;
cout<<temp->data<<" has been deleted"<<endl;
delete(temp);
}

void delete_end(){
node *traverse;
traverse = start;
while(traverse->next != NULL){
traverse = traverse->next;
}
traverse->prev->next = NULL;
cout<<traverse->data<<" has been deleted"<<endl;
delete(traverse);
}

void delete_after(){
int value;
cout<<"Enter the product after which the node has to be deleted"<<endl;
cin>>value;
node *traverse;
traverse = start;
while(traverse->data != value){
    traverse = traverse->next;
}
node *temp;
temp = traverse->next;
traverse->next = temp->next;
temp->next->prev = traverse;
cout<<temp->data<<" has been deleted"<<endl;
delete(temp);

}




        void buy(){
                system("cls");
                string products[20];
                int pay=0,no,c=0,price,id,i=1;
                if(start==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= display1();
```

```
            while (i<=no){
                        struct node *cur=start;

                  int quant,cho;
            cout<<"Enter id of item that you want to buy: \n";
      int id,pos=0;
            cin>>id;
            pos=search(id);

            if(pos<=count){

                  while(cur->data!=id){
                        cur=cur->next;
                  }

            cout<<"How many quantities you want: \n";
            cin>>quant;
            products[c]=cur->name; c++;
            pay=pay+(cur->y*quant);
            cur->x=cur->x-quant;
            i++;
            }
            else{
                  cout<<"\n<<<<<<<<<This item is not available in our store at this
      time>>>>\n\n";
              }
      }

            //system("cls");
      cout<<"\n\n\n\t\t\tYou have bought : ";
      for(int i=0;i<no;i++){
            cout<<products[i]<<",";
      }
      price=pay*(0.95);
            cout<<"\n\nOriginal price : "<<pay;
        cout<<"\n with 5% discount: "<<price<<"\nThank you! for the shopping\n\n";
      }
      }
            int search(int id)
       {
            int count=1;
            struct node *p=start;
            while(p!=NULL)
            {
                  if(p->data==id)
                  {

                        break;    }
                  else
                        count++;
                        p=p->next;
            }

            return count;
 }

int display1(){
            //          system("cls");
                        int c=0;
                        struct node *p=start;
                        cout<<"Existing products are:\n";
                        cout<<"ID\t\tProduct Name\t\tPrice\t\tQuantity\n";
                        while(p!=NULL)
                        {
                                    cout<<p->data<<"\t\t"<<p->name<<"\t\t\t"<<p->y<<"\t\t"<<check(p-
>x)<<"\n";
                                    p=p->next;
                                    c=c+1;
                        }
                        cout<<"\nTotal products in our store is : "<<c<<"\n\n\n";
                        return c;
                }
                string check(int quant){
                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";
                        }
```

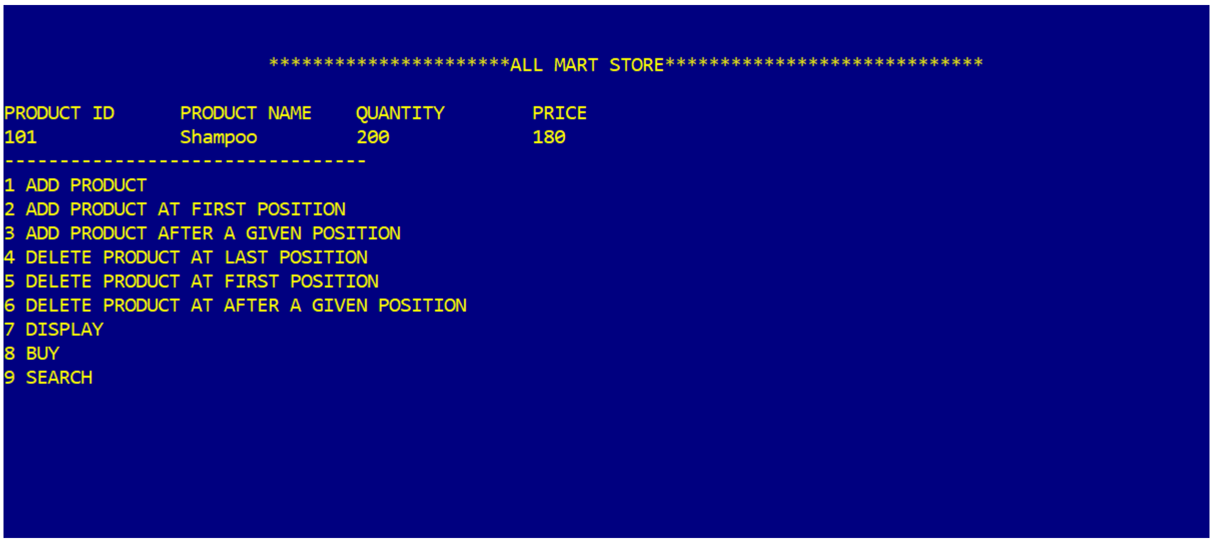
```
   void srch()
   {
      system("cls");
      cout<<"\n\n\t\t\t***********************ALL                                        MART
STORE*****************************\n\n\n"<<endl;
      string nam;
      cout<<"Enter Product Name "<<endl;
      cin>>nam;
int f=0;

         struct node *p=start;
         while(p!=NULL)
         {
                  if(nam==p->name)
                  {
                     f=1;
                           break;
                              }

                           p=p->next;
         }

         if(f==1)
   {
         cout<<"Product is available"<<endl;
   }
   else{
         cout<<"Out of Stock"<<endl;

   }
   }

int main(){

system("color 1E");
int option;
do{
cout<<"..............................................."<<endl;
cout<<"1 ADD PRODUCT"<<endl;
cout<<"2 ADD PRODUCT AT FIRST POSITION"<<endl;
cout<<"3 ADD PRODUCT AFTER A GIVEN POSITION"<<endl;
cout<<"4 DELETE PRODUCT AT LAST POSITION"<<endl;
cout<<"5 DELETE PRODUCT AT FIRST POSITION"<<endl;
cout<<"6 DELETE PRODUCT AT AFTER A GIVEN POSITION"<<endl;
cout<<"7 DISPLAY"<<endl;
cout<<"8 BUY"<<endl;
cout<<"9 SEARCH"<<endl;
cin>>option;
  switch(option){
  case 1:insert_node();
       break;
  case 2: add_begin();
        break;
  case 3: add_after();
        break;
  case 4: delete_end();
        break;
  case 5: delete_beg();
        break;
  case 6: delete_after();
        break;
  case 7: display();
        break;
  case 8: buy();
        break;
  case 9: srch();
         break;

  }

}while(option != 0);
return 0;}
```

SCREENSHOTS

1) ADD PRODUCT OPERATION

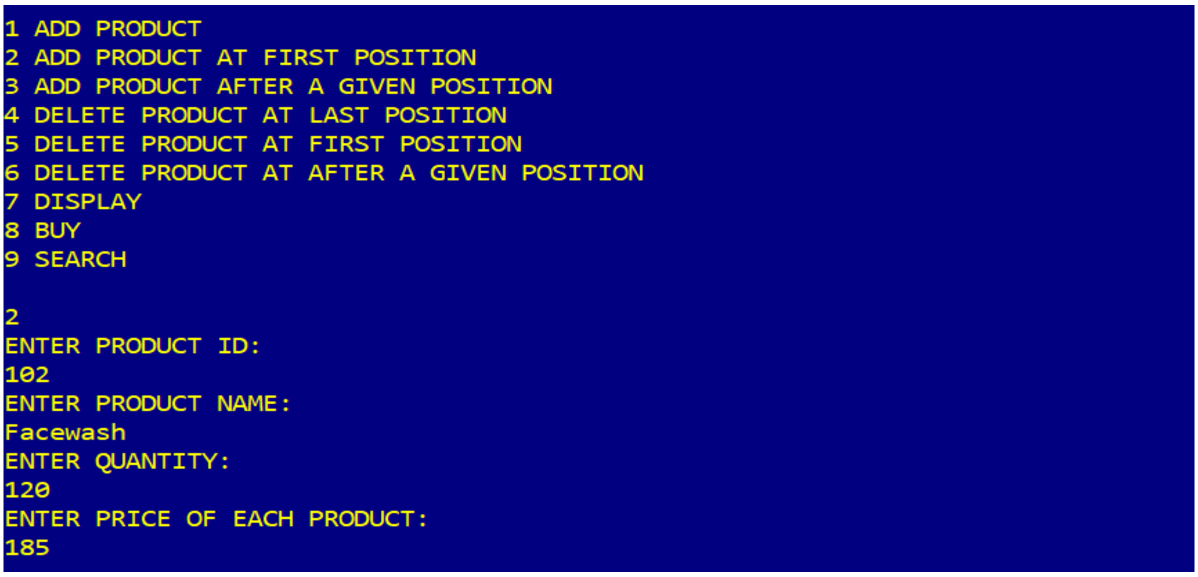


```
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH

1
ENTER PRODUCT ID:
101
ENTER PRODUCT NAME:
Shampoo
ENTER QUANTITY:
200
ENTER PRICE OF EACH PRODUCT:
180
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```

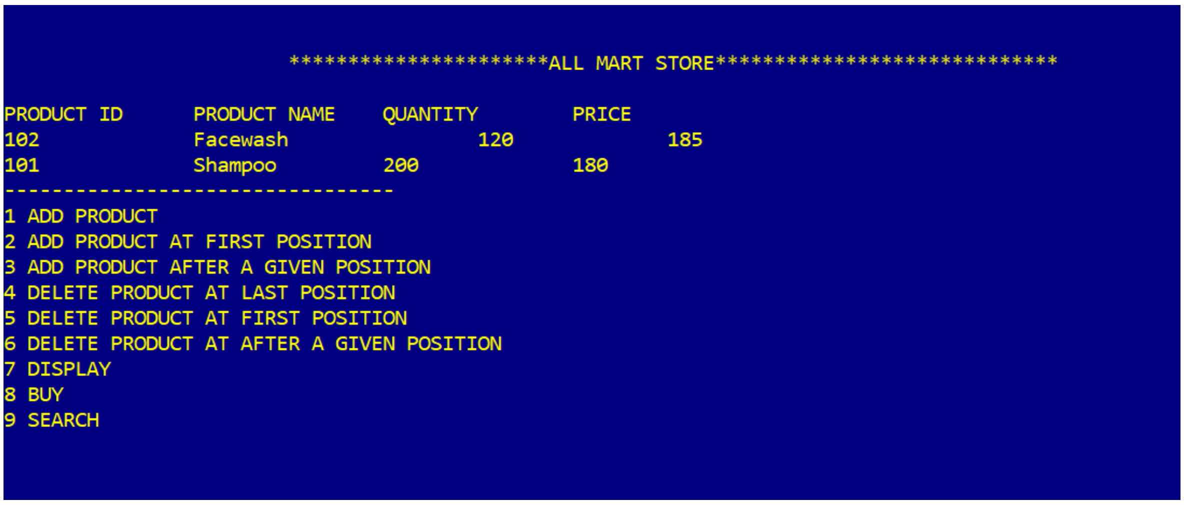


```
                    *********************ALL MART STORE****************************

PRODUCT ID      PRODUCT NAME    QUANTITY        PRICE
101             Shampoo         200             180
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```

2) <u>ADD PRODUCT AT FIRST POSITION OPERATION</u>

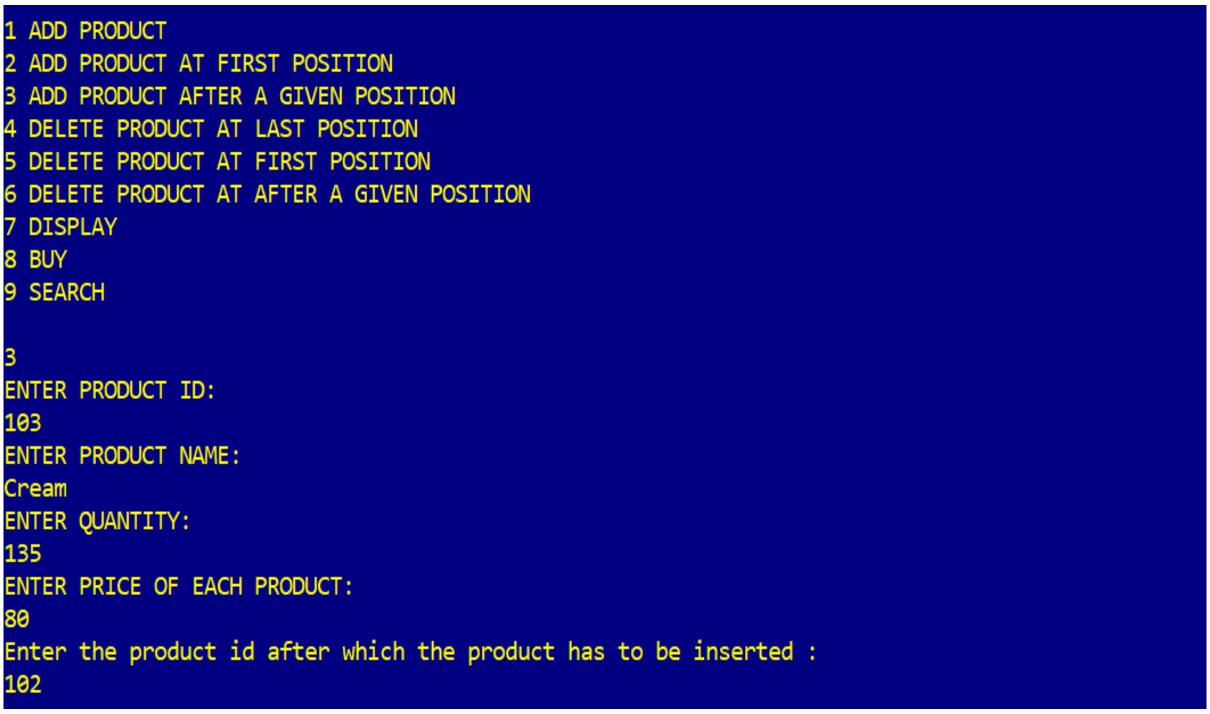


```
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH

2
ENTER PRODUCT ID:
102
ENTER PRODUCT NAME:
Facewash
ENTER QUANTITY:
120
ENTER PRICE OF EACH PRODUCT:
185
```

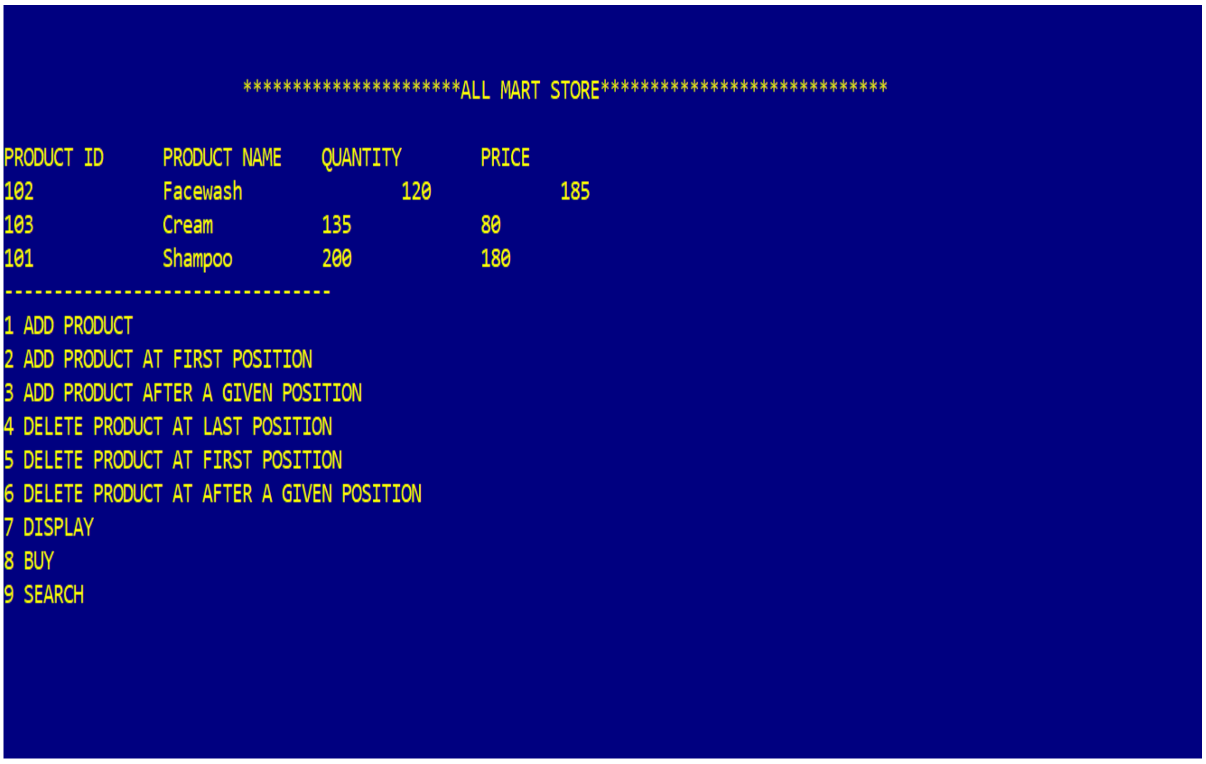


```
                    *********************ALL MART STORE****************************

PRODUCT ID      PRODUCT NAME    QUANTITY        PRICE
102             Facewash                120             185
101             Shampoo         200             180
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```

3) ADD PRODUCT AFTER A GIVEN POSITION OPERATION

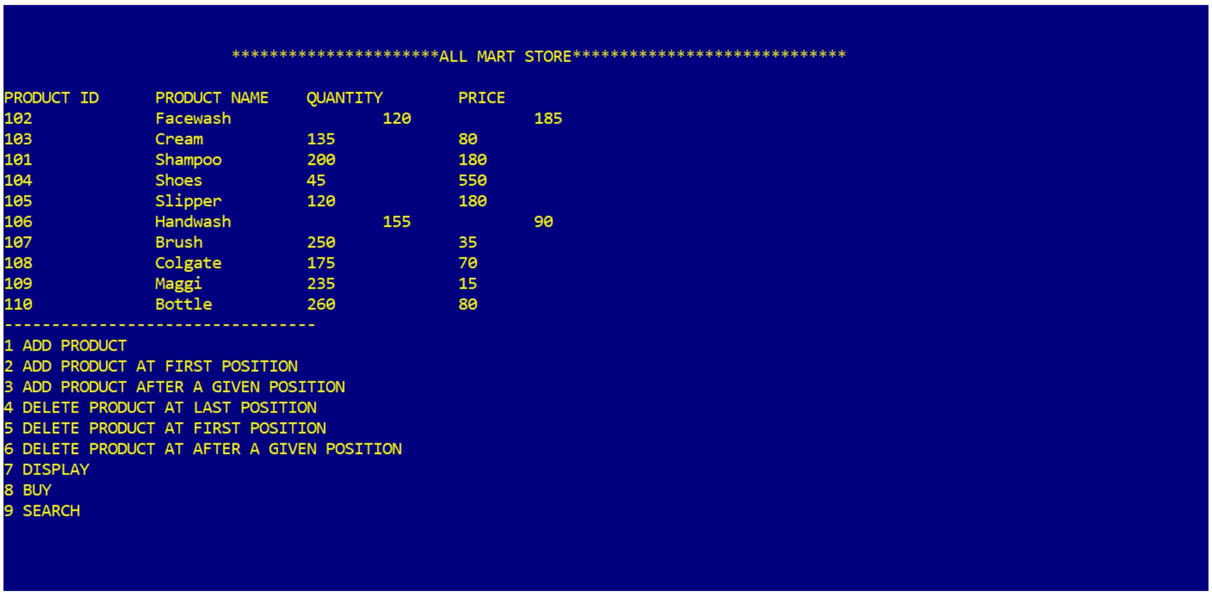


```
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH

3
ENTER PRODUCT ID:
103
ENTER PRODUCT NAME:
Cream
ENTER QUANTITY:
135
ENTER PRICE OF EACH PRODUCT:
80
Enter the product id after which the product has to be inserted :
102
```



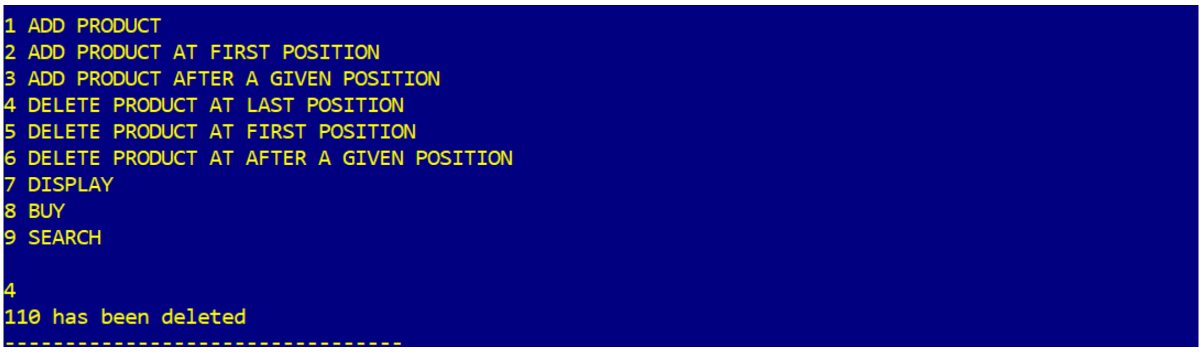
```
                    *********************ALL MART STORE****************************

PRODUCT ID      PRODUCT NAME    QUANTITY        PRICE
102             Facewash                120             185
103             Cream           135             80
101             Shampoo         200             180
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```

4) ADD SOME MORE PRODUCTS AND DISPLAY OPERATION

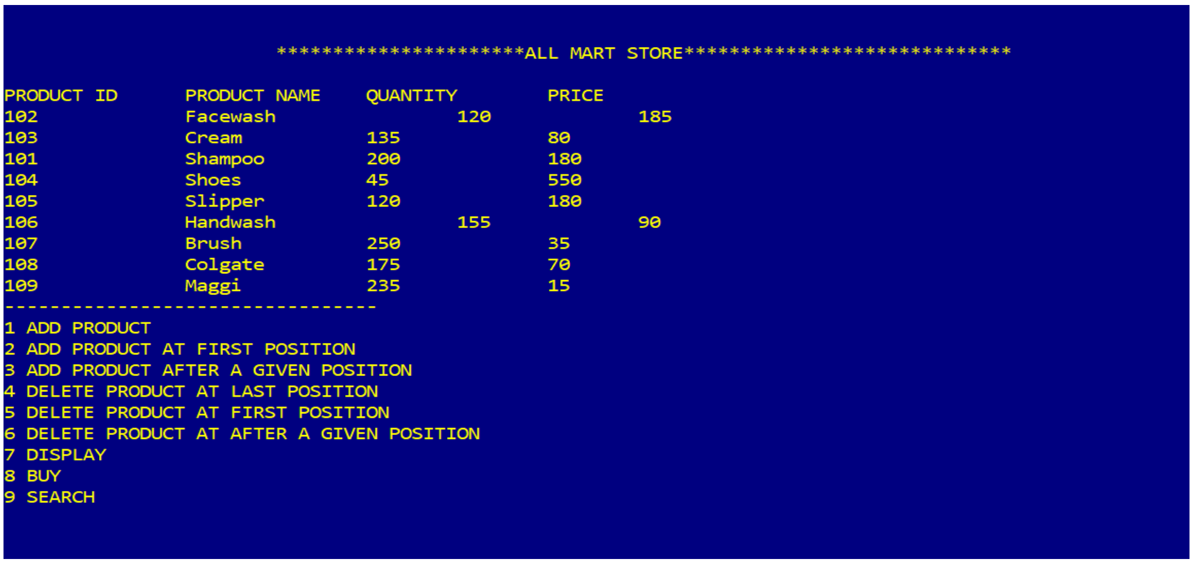


```
                      *********************ALL MART STORE*****************************

PRODUCT ID      PRODUCT NAME    QUANTITY        PRICE
102             Facewash                120             185
103             Cream           135             80
101             Shampoo         200             180
104             Shoes           45              550
105             Slipper         120             180
106             Handwash                155             90
107             Brush           250             35
108             Colgate         175             70
109             Maggi           235             15
110             Bottle          260             80
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```

5) DELETE PRODUCT AT LAST POSITION OPERATION

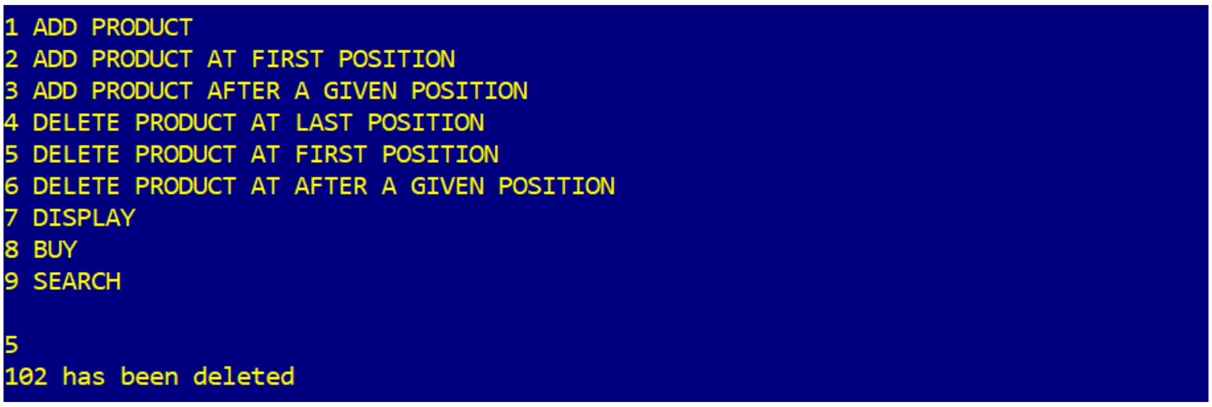


```
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH

4
110 has been deleted
---------------------------------
```

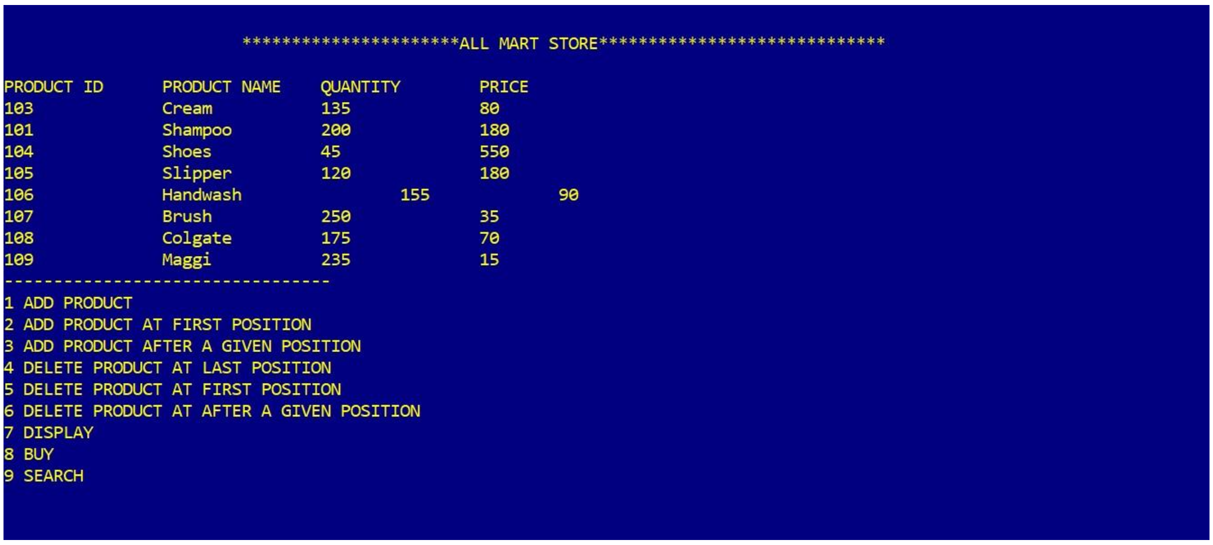


```
                      *********************ALL MART STORE*****************************

PRODUCT ID      PRODUCT NAME    QUANTITY        PRICE
102             Facewash                120             185
103             Cream           135             80
101             Shampoo         200             180
104             Shoes           45              550
105             Slipper         120             180
106             Handwash                155             90
107             Brush           250             35
108             Colgate         175             70
109             Maggi           235             15
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```

6) DELETE PRODUCT AT FIRST POSITION OPERATION

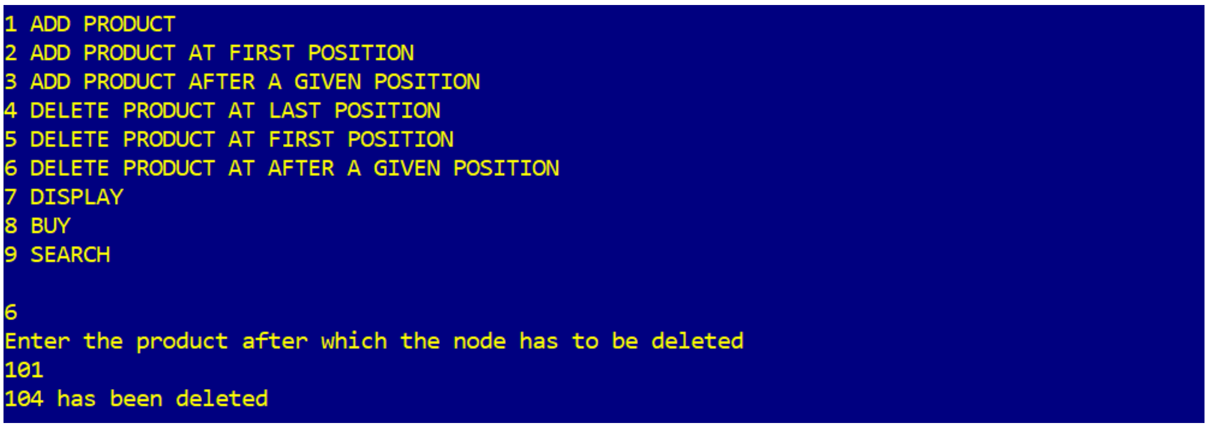


```
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH

5
102 has been deleted
```

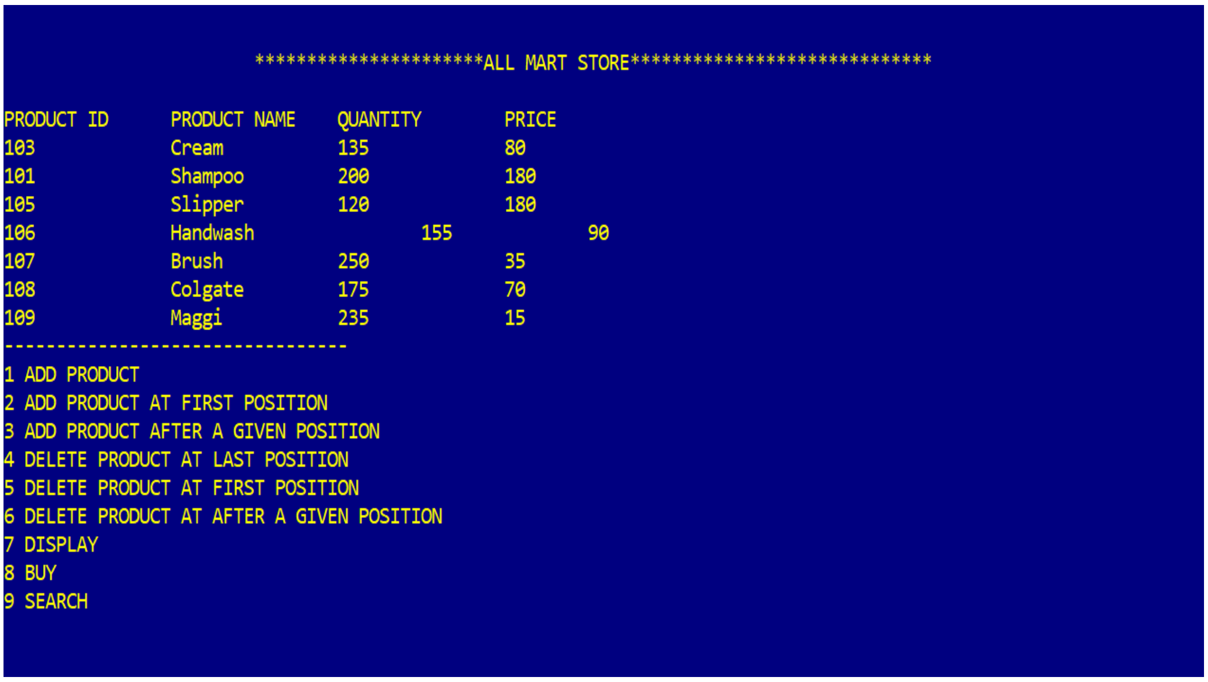


```
                        *********************ALL MART STORE****************************

PRODUCT ID      PRODUCT NAME    QUANTITY        PRICE
103             Cream           135             80
101             Shampoo         200             180
104             Shoes           45              550
105             Slipper         120             180
106             Handwash                155             90
107             Brush           250             35
108             Colgate         175             70
109             Maggi           235             15
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```

7) DELETE PRODUCT AT AFTER A GIVEN POSITION OPERATION

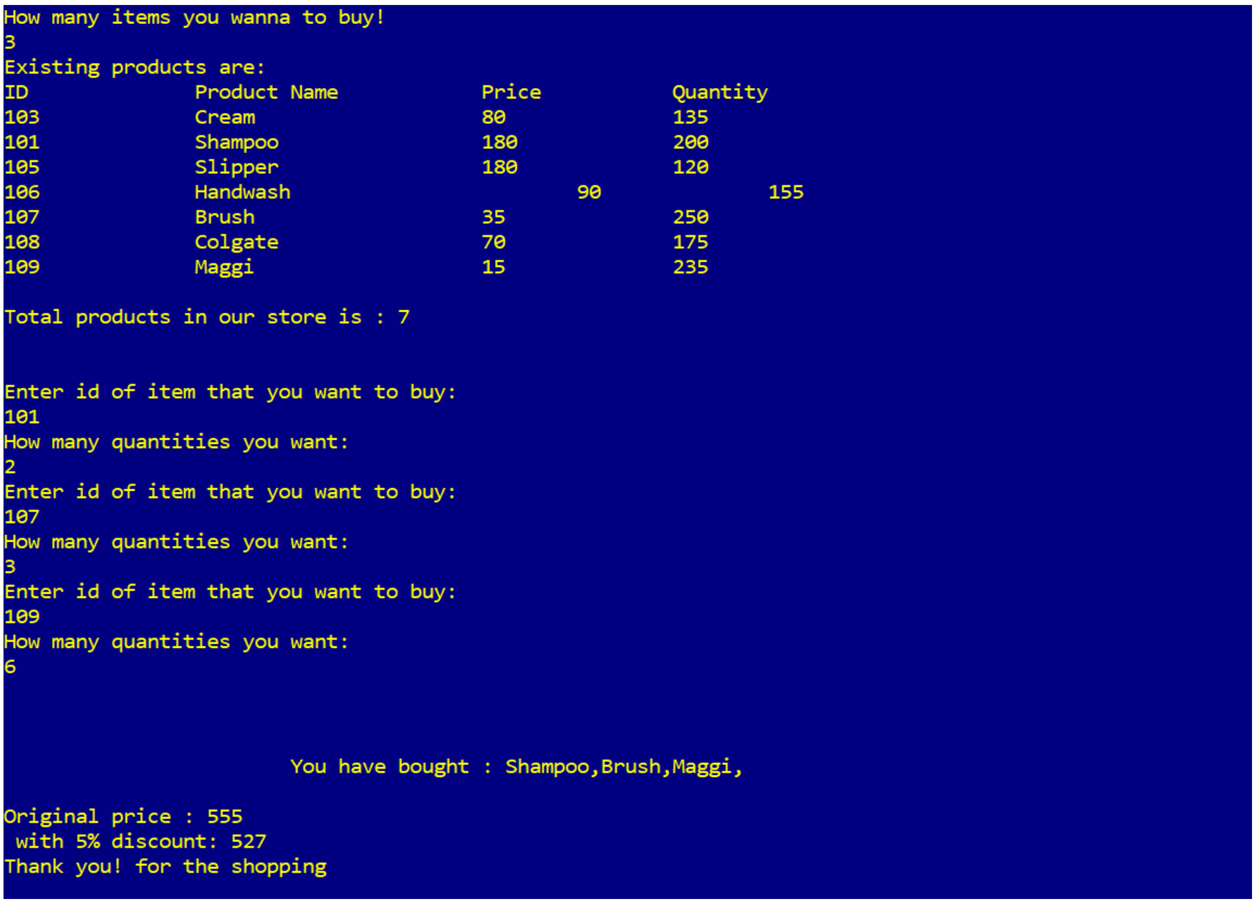


```
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH

6
Enter the product after which the node has to be deleted
101
104 has been deleted
```



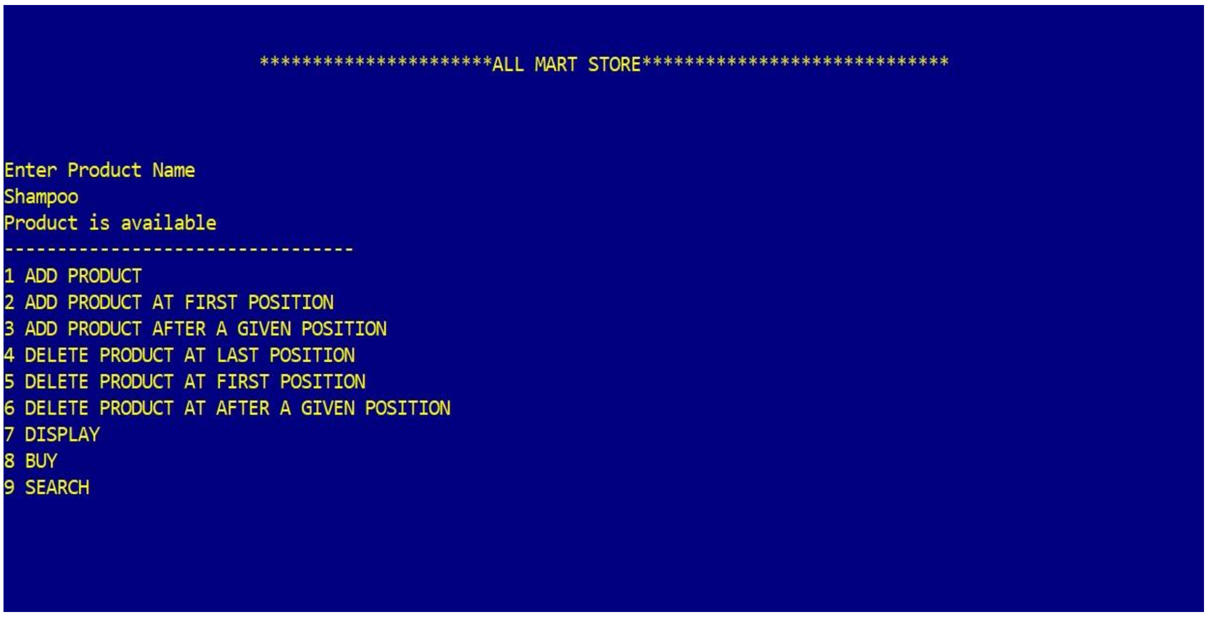
```
                        *********************ALL MART STORE****************************

PRODUCT ID      PRODUCT NAME    QUANTITY        PRICE
103             Cream           135             80
101             Shampoo         200             180
105             Slipper         120             180
106             Handwash                155             90
107             Brush           250             35
108             Colgate         175             70
109             Maggi           235             15
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```

8) <u>BUY OPERATION</u>

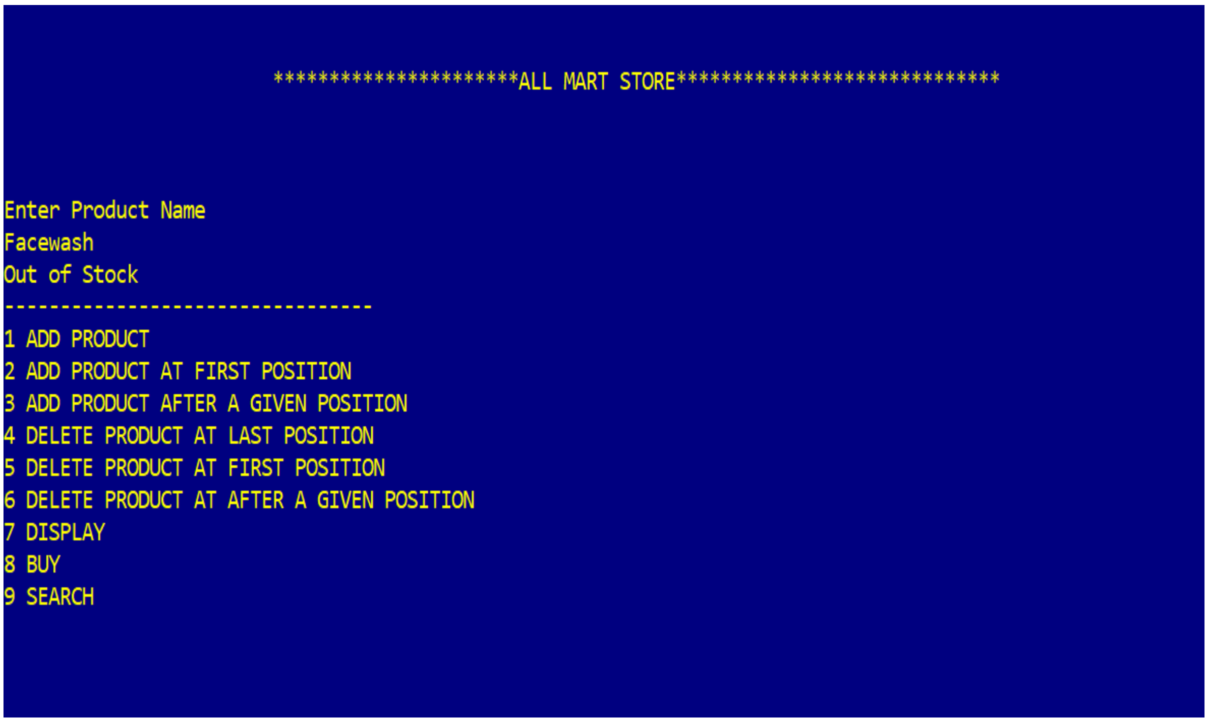


```
How many items you wanna to buy!
3
Existing products are:
ID              Product Name            Price           Quantity
103             Cream                   80              135
101             Shampoo                 180             200
105             Slipper                 180             120
106             Handwash                        90              155
107             Brush                   35              250
108             Colgate                 70              175
109             Maggi                   15              235

Total products in our store is : 7


Enter id of item that you want to buy:
101
How many quantities you want:
2
Enter id of item that you want to buy:
107
How many quantities you want:
3
Enter id of item that you want to buy:
109
How many quantities you want:
6



                        You have bought : Shampoo,Brush,Maggi,

Original price : 555
 with 5% discount: 527
Thank you! for the shopping
```

9) SEARCH OPERATION



```
                         *********************ALL MART STORE****************************


Enter Product Name
Shampoo
Product is available
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```



```
                         *********************ALL MART STORE****************************


Enter Product Name
Facewash
Out of Stock
---------------------------------
1 ADD PRODUCT
2 ADD PRODUCT AT FIRST POSITION
3 ADD PRODUCT AFTER A GIVEN POSITION
4 DELETE PRODUCT AT LAST POSITION
5 DELETE PRODUCT AT FIRST POSITION
6 DELETE PRODUCT AT AFTER A GIVEN POSITION
7 DISPLAY
8 BUY
9 SEARCH
```

IMPORTANCE

1. The system reduces much of human efforts in calculating bill especially for huge products.
2. Saves money and resources of organization and excludes of use of paper or sheets in making bill.
3. It can detect the product information and their price instantaneously using RFID technology.
4. Saves time.
5. It provides accuracy and faultless in billing calculations.
6. The system is designed having attractive GUI and with detailed description.
7. It is flexible and user-friendly.
8. It also notifies customers through sending an electronic bill via email.

CONCLUSION

In conclusion, Supermarket Billing System has to do with making appropriate effort to stop the rising problem to all manual supermarket operation in order to enhance the operation of such supermarket. In this project, the software or system that can be used to aid all supermarkets that is still operating manually have been successfully developed. The software can be implementing in all types of supermarket as mentioned in the second chapter. The software has a large memory of storing all the goods in the supermarket and also keeping record it is highly effective and accurate.

REFERENCES

1. Zhu, X., Shi, Z., & Zhang, Y. (2013). Improvement of the Supermarket Management System based on Commercial Category.
2. Sainath, S., Surender, K., & Vikram Arvind, V. (2014). Automated Shopping Trolley for Super Market Billing System. *International Journal of Computer Applications*, *975*, 8887.
3. Prof. S.H. Patil1, Mayur Chaudhari2, Amit Gore3, Rajendra Kale4, Prof. H.A. Hingoliwala5, A Survey on Technologies Used for Billing System in Supermarkets;2015
4. Prerana, T., Ranjan, S., & Kaushik, P. (2017). Smart Shopping Cart for Automatic Billing In Supermarket.