ABSTRACT

This project aims to carry out the process of placing an online order in a supermarket for items and then calculating the bill for the items requested with the help of Data structures.

This system is already used in many e-commerce websites to take the consumer's requirements, send it to the supplier and making both ends happy by mediating a suitable price.

This program also aims to do the same with the help of Data structures and can be easily modified to include more items.

EXPLANATION

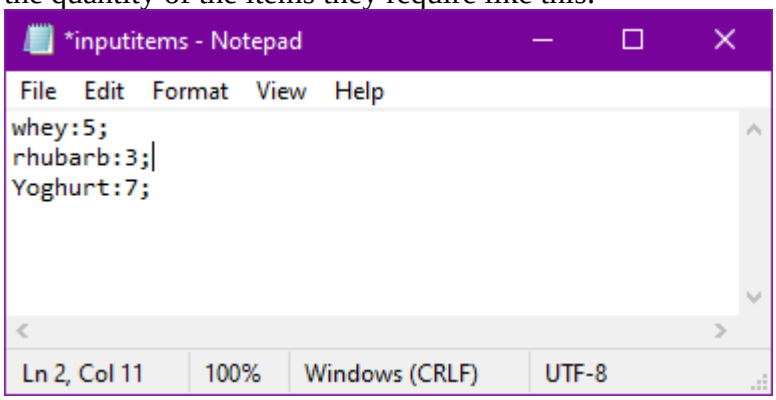
We use the program to calculate the bill for the inputted items by the user
This program uses 2 Data Structures:

1) Doubly Linked List

We Use DLL to store the items which are available for purchase in the store
Each Node of the DLL Stores 3 items along with the price of each item.

```
insertFront("apple","carrot","milk",10,12,18);
insertFront("dates","onion","butter",14,19,30);
insertFront("huito","rhubarb","cheese",20,13,40);
insertFront("pear","sprout","yoghurt",30,27,130);
insertFront("kepel","parsley","cheddar",44,18,140);
insertFront("guava","corncob","ayran",14,20,55);
insertFront("grape","pumpkin","kefir",40,23,46);
insertFront("mango","spinach","malai",24,9,59);
insertFront("prune","brinjal","cream",12,19,90);
insertFront("peach","shallot","whey",20,7,105);
```

After seeing the list, The user inputs the items they want into the text document along with the quantity of the items they require like this:



The program then reads the input file and pushes the Item name and quantity inputted by the user into a stack

2) Stack(cart, unavailableitems)

The Stack contains the input item name and the quantity of the item which was read from the input file.

```
cart.push(item.toLowerCase(),Integer.parseInt(quant));
```

After this the program runs to compare the name of the item in the stack to the names of items in the Doubly linked list in a function which then returns the price of that item.

```
public int billcalc(StackDList cart)
{
    int price=0;
    DLLNode t = DLLhead;
    while(t!=null)
    {
        if(cart.GetItem().compareTo(t.fruit)==0)
        {
            price= t.pricef;
            break;
        }
        else if(cart.GetItem().compareTo(t.vegetable)==0)
        {
            price= t.pricev;
            break;
        }
        else if(cart.GetItem().compareTo(t.diary)==0)
        {
            price= t.priced;
            break;
        }
        else
        {
            price=-5;
        }
        t=t.next;
    }
    return price;
}
```

If the price returned is -5, it means that the item is unavailable and the item is popped from the cart Stack and is pushed into another Stack called Unavailable Items

```
if(market.billcalc(cart)==-5)
{
    unavailableitems.push(cart.GetItem(),cart.GetQuantity());
    cart.pop();
}
```

If the item is a valid item with a valid price, then it gets popped from the stack and inserted into a Singly linked list which contains the following items:

- Item name
- Item quantity
- Item price for 1 quantity
- Item price for n quantity

3)Singly Linked List (bill)
The singly linked list stores the above details which is used to print the bill and calculate the total amount for the bill.

```
else
{
   bill.insertfront(cart.GetItem(),cart.GetQuantity(),market.billcalc(cart),cart.GetQuantity()*market.billcalc(cart));
   billtotal+=cart.GetQuantity()*market.billcalc(cart);
   cart.pop();
}
```

Finally after all these processes, the items are displayed in a billing format with the name and phone number of the customer in a Bill format

```
-----------------------------------------------------------------
|      Item    |   quantity    |       price   |       total   |
-----------------------------------------------------------------
|      yoghurt |       7       |       130     |       910     |
|      rhubarb |       3       |       13      |       39      |
|      whey    |       5       |       105     |       525     |
-----------------------------------------------------------------
                                    Bill total:     1474
```

CODE

```
package DSA1;
import java.io.*;
import java.util.Scanner;
class Endsem
{
   public static void main(String args[])
   {
      try
      {   Scanner sc=new Scanner(System.in);
         Endsem obj=new Endsem();
         DLLendsem market=new DLLendsem();
         StackDList cart=new StackDList();
         StackDList unavailableitems=new StackDList();
         SLIST bill=new SLIST();
         market.ins();


         System.out.println("\t\t\tWELCOME TO AMRIRA SUPERMARKET");
         System.out.println("\nORDER PLACEMENT TAB");
         System.out.print("\n\nName: ");

         String name=sc.nextLine();
         System.out.print("\nPhone Number: ");
         long phoneno=sc.nextLong();
         System.out.println("\n");
         System.out.println("Fruits\t\t\t\tVegetables\t\t\tDiary products");
         System.out.println(" ---------------------------------------------------------------------------
");
         market.display();
         System.out.println("\n");
         System.out.println("\nWrite down any of the above products in the file and its
quantity seperated by a (:) and ending with a semicolon(;)");
         System.out.println("example:\nproductname1:quantitiy;\nproductname2:quantitiy;");
          System.out.println("Once done please enter 'Y'");

         while(true)
         {
            if(sc.nextLine().compareTo("Y")==0)
               break;
            else
               continue;
         }
         File inp=new File("D:\\2nd Sem\\Data Structures & Algorithms
1\\EndSem\\inputitems.txt");

         //Thread.sleep(10000);
         Scanner sc1=new Scanner(inp);
         int i;
```

```
String quant="";
String input;
String item="";
while(sc1.hasNextLine())
{
   input=sc1.nextLine();
   item="";
   quant="";
   for(i=0;i<input.length();i++)
   {
      if(input.charAt(i)!=':')
      {
         item+=input.charAt(i);
      }
      else
      {
         i=i+1;
         while(input.charAt(i)!=';')
         {
            quant+=input.charAt(i);
            i=i+1;
         }
      }
   }
   cart.push(item.toLowerCase(),Integer.parseInt(quant));
}
int billtotal=0;
while(cart.GetSize()!=0)
{
   if(market.billcalc(cart)==-5)
   {
      unavailableitems.push(cart.GetItem(),cart.GetQuantity());
      cart.pop();
   }
   else
   {

bill.insertfront(cart.GetItem(),cart.GetQuantity(),market.billcalc(cart),cart.GetQuantity()*mar
ket.billcalc(cart));
      billtotal+=cart.GetQuantity()*market.billcalc(cart);
      cart.pop();
   }
}

System.out.print('\u000C');
System.out.println("\t\t\tORDER SUMMARY");
System.out.println("Name:        "+name);
System.out.println("Phone number: "+phoneno);
System.out.println("\n");
if(unavailableitems.GetSize()!=0)
```

```
            {
               System.out.println("The following items are not available:");
               unavailableitems.display();
               System.out.println("\nApologies for any inconvenience caused.\n\n");
            }
            System.out.println(" ------------------------------------------------------------------");
            System.out.println(" |\tItem\t|   quantity    |\tprice\t|\ttotal\t|");
            System.out.println(" ------------------------------------------------------------------");
            bill.display();
            System.out.println(" ------------------------------------------------------------------");
            System.out.println("\t\t\t\t\tBill total:     "+billtotal);
         }
         catch(Exception e)
         {
            System.out.println("An error occurred.");
         }
      }
   }

   //for market items
   class DLLNode
   {
      String fruit,vegetable,diary;
      int pricef,pricev,priced;
      DLLNode prev;
      DLLNode next;
      DLLNode()
      {
         fruit="";
         vegetable="";
         diary="";
         pricef=0;
         pricev=0;
         priced=0;
         prev = null;
         next = null;
      }
      DLLNode(String f,String v,String d,int pf,int pv,int pd)
      {
         fruit=f;
         vegetable=v;
         diary=d;
         pricef=pf;
         pricev=pv;
         priced=pd;
         prev = null;
         next = null;
      }
      DLLNode(DLLNode n1, String f,String v,String d,int pf,int pv,int pd, DLLNode n2)
      {
```

```
        fruit=f;
        vegetable=v;
        diary=d;
        pricef=pf;
        pricev=pv;
        priced=pd;
        prev = n1;
        next = n2;
    }
}
class DLLendsem
{
    DLLNode DLLhead;
    DLLNode DLLtail;
    StackNode Stackhead;
    StackNode Stacktail;
    DLLendsem()
    {
        DLLhead = null;
        DLLtail = null;
    }
    DLLendsem(String f,String v,String d,int pf,int pv,int pd)
    {
        DLLhead= new DLLNode(f,v,d,pf,pv,pd);
        DLLtail = DLLhead;
    }
    void insertFront(String f,String v,String d,int pf,int pv,int pd)
    {
        DLLNode t = new DLLNode(f,v,d,pf,pv,pd);
        if (DLLhead == null)
        {
            DLLhead = DLLtail = t;
        }
        else
        {
            DLLhead.prev = t;
            t.next = DLLhead;
            DLLhead = t;
        }
    }
    void display()
    {
        DLLNode t = DLLhead;
        while(t != null)
        {
            System.out.println(t.fruit+"\t\t\t\t"+t.vegetable+"\t\t\t\t"+t.diary);
            t = t.next;
        }
    }
    public int billcalc(StackDList cart)
```

```
    {
        int price=0;
        DLLNode t = DLLhead;
        while(t!=null)
        {
            if(cart.GetItem().compareTo(t.fruit)==0)
            {
                price= t.pricef;
                break;
            }
            else if(cart.GetItem().compareTo(t.vegetable)==0)
            {
                price= t.pricev;
                break;
            }
            else if(cart.GetItem().compareTo(t.diary)==0)
            {
                price= t.priced;
                break;
            }
            else
            {
                price=-5;
            }
            t=t.next;
        }
        return price;
    }
    void ins()
    {
            insertFront("apple","carrot","milk",10,12,18);
            insertFront("dates","onion","butter",14,19,30);
            insertFront("huito","rhubarb","cheese",20,13,40);
            insertFront("pear","sprout","yoghurt",30,27,130);
            insertFront("kepel","parsley","cheddar",44,18,140);
            insertFront("guava","corncob","ayran",14,20,55);
            insertFront("grape","pumpkin","kefir",40,23,46);
            insertFront("mango","spinach","malai",24,9,59);
            insertFront("prune","brinjal","cream",12,19,90);
            insertFront("peach","shallot","whey",20,7,105);
    }
}

//for cart items
class StackNode
{
    String item;
    int quantity;
    StackNode prev;
    StackNode next;
```

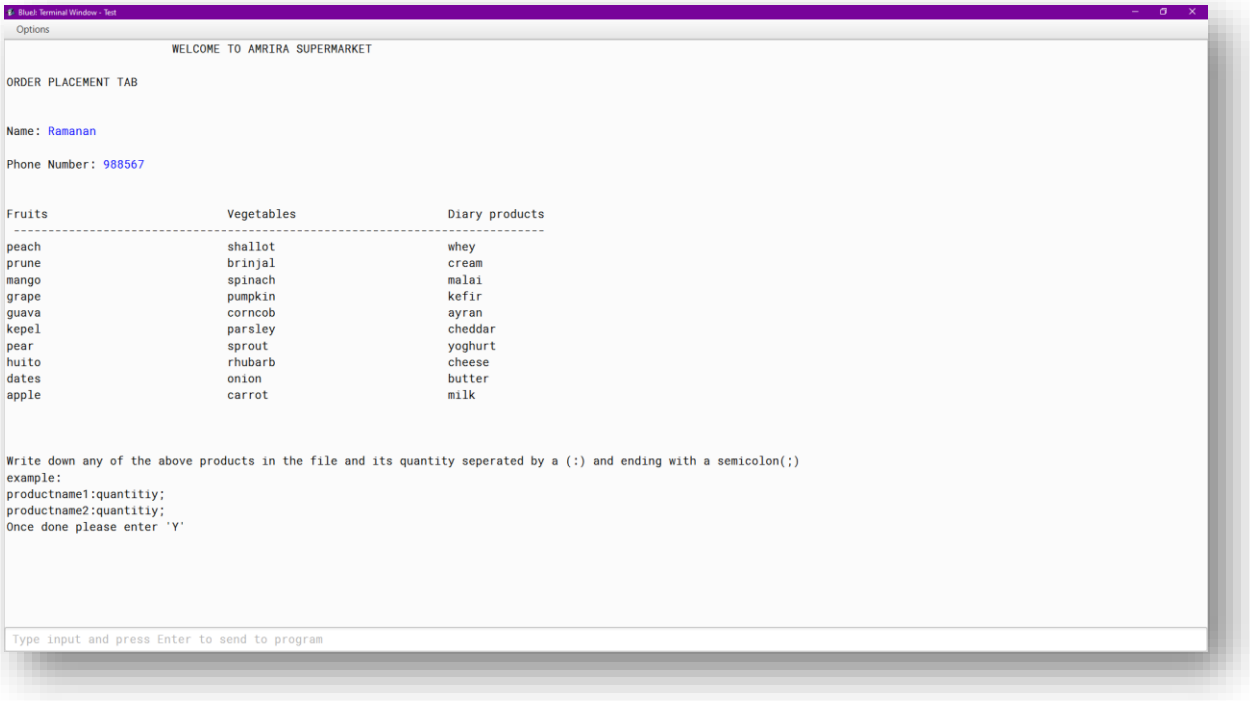
```
    StackNode()
    {
       item="";
       quantity=0;
       prev = null;
       next = null;
    }
    StackNode(String i,int q)
    {
       item=i;
       quantity=q;
       prev = null;
       next = null;
    }
    StackNode(StackNode n1,String i,int q, StackNode n2)
    {
       item=i;
       quantity=q;
       prev = n1;
       next = n2;
    }
}
class StackDList
{
    StackNode Stackhead;
    StackNode Stacktail;
    StackDList()
    {
       Stackhead = null;
       Stacktail = null;
    }
    StackDList(String i,int q)
    {
       Stackhead= new StackNode(i,q);
       Stacktail = Stackhead;
    }
    void push(String i,int q)
    {
       if(Stackhead == null)
       {
          Stackhead = new StackNode(i,q);
          Stacktail = Stackhead;
       }
       else
       {
          StackNode t = new StackNode(i,q);
          Stacktail.next = t;
          t.prev = Stacktail;
          Stacktail = t;
       }
```

```
}
void pop()
{
   if(Stackhead == null)
   {
      return;
   }
   else
   {
      if(Stackhead != Stacktail)
      {
         Stackhead = Stackhead.next;
      }
      else
      {
         Stackhead = Stacktail = null;
      }
   }
}
void display()
{
   StackNode t = Stackhead;
   while(t != null)
   {
      System.out.println(t.item);
      t = t.next;
   }
}
int GetQuantity()
{
   return Stackhead.quantity;
}
String GetItem()
{
   return Stackhead.item;
}
int GetSize()
{
   int count=0;
   StackNode t=Stackhead;
   if(t == null)
   {
      return 0;
   }
   else
   {
      while(t!=null)
      {
         t=t.next;
         count++;
```

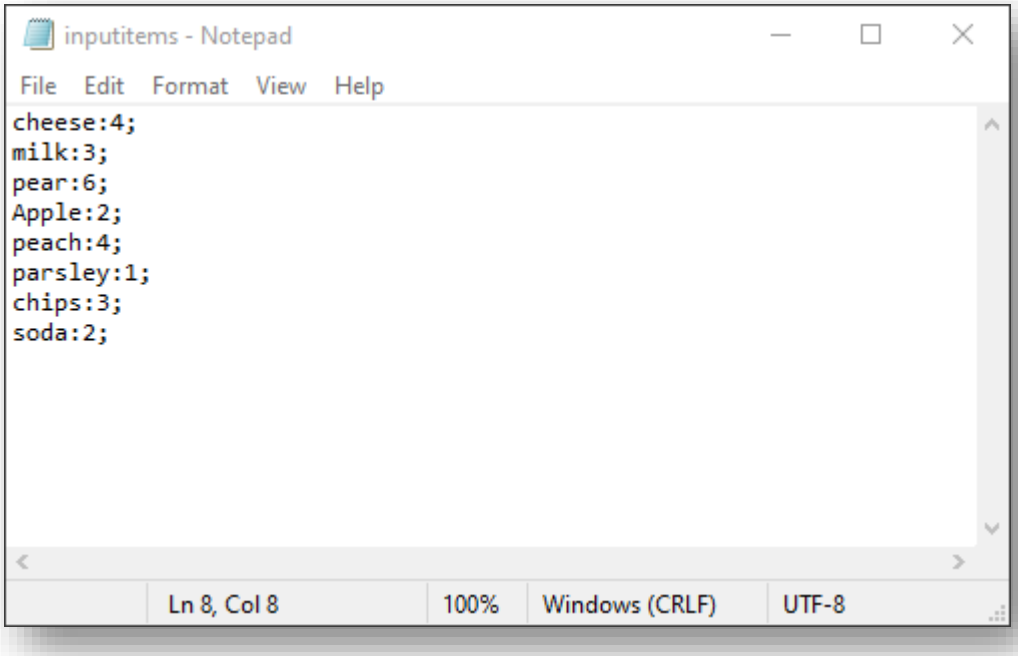
```
            }
        }
            return count;
    }
}


//for displaying billamt
class SLLNode
{
    String itemname;
    int quantity;
    int price;
    int totalprice;

    SLLNode next;
    SLLNode()
    {
        itemname="";
        quantity=0;
        price=0;
        totalprice=0;
        next = null;
    }
    SLLNode(String i,int q,int p,int tp)
    {
        itemname=i;
        quantity=q;
        price=p;
        totalprice=tp;
        next = null;
    }
    SLLNode(String i,int q,int p,int tp, SLLNode n)
    {
        itemname=i;
        quantity=q;
        price=p;
        totalprice=tp;
        next = n;
    }
}
class SLIST
{
    SLLNode SLLhead;
    SLIST()
    {
        SLLhead = null;
    }
    SLIST(String i,int q,int p,int tp)
    {
```

```
        SLLhead = new SLLNode(i,q,p,tp);
    }
    void insertfront(String i,int q,int p,int tp)
    {
        SLLNode temp = new SLLNode(i,q,p,tp);
        temp.next = SLLhead;
        SLLhead = temp;
    }
    void display()
    {
        SLLNode temp = SLLhead;
        if(temp == null)
        {
            System.out.println("List is empty");
        }
        else
        {
            while(temp!=null)
            {
                System.out.println("
|\t"+temp.itemname+"\t|\t"+temp.quantity+"\t|\t"+temp.price+"\t|\t"+temp.totalprice+"\t|");
                temp=temp.next;
            }
        }
    }
}
```

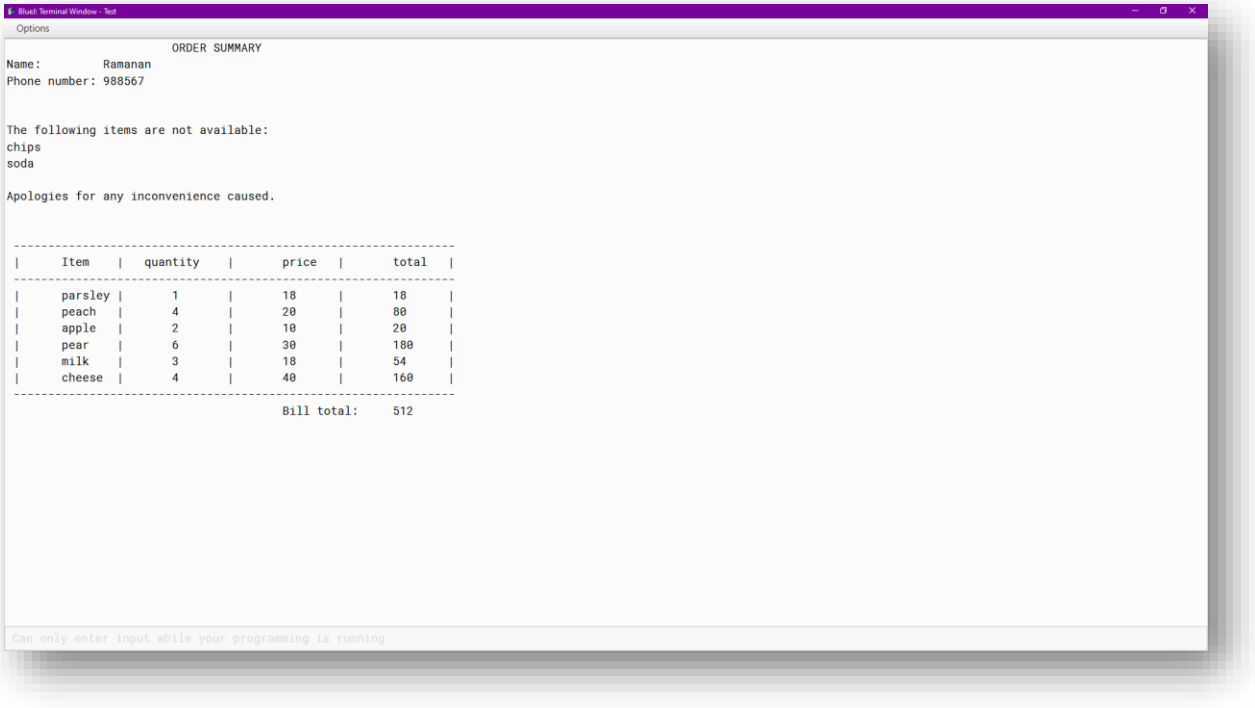
OUTPUT



The items inputted are:



The bill format for the following items are:



As you can see all the available items are processed in the bill whereas the items which are not available are displayed as unavailable.

---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---x---