GIT Department of Computer Engineering
CSE 222/505 - Spring 2021
Homework-3 # Report

Can Duyar
171044075

PART-1:(It looks like hw1 but I changed some parts on it end explained full design because of my KWLinkedlist,KWArrayList and HybridList implementation instead of arrays)

I used,

-LinkedList in the textbook for information about branches
-ArrayList in the textbook for users of the automation system
-HybridList you implemented for the furnitures

* I applied your rules for my homework completely.
* The whole program works correctly(as I explained in test cases/running command and results).

1) Detailed system requirements:

In this assignment, I designed and implemented the furniture-automation system with linkedlist,arraylist and hybridlist data structures as a difference from homework-1. This automation system has users such as administrators, branch employees and customers. Each of the users has some features that are related with their roles.

Administrators:
-Manage the system by adding and removing branches and branch employees.
-They can also query whether there are any products that need to be supplied.
Branch Employees:
-They can inquire about the products in stock, inform the manager that the product should be purchased when any product is less than the requested amount, add / remove products, make sales, access the information of the previous orders of a customer by using the customer number and add new order to this section.
-They update the customers' previous orders section during their sales from the store.
Customers:
-Subscribes to the system, he/she uses his/her name, surname and e-mail address, and defines a password.
-They log into the system, they can search for products, see the list of products, see which store a product is in, shop online by entering address and phone information, and view their previous orders.

To implement complete automation system with these features; First, we need an interface for users.This interface should include getters and setters for name and surname of these user types. In addition to “Users interface”, we also need a general class named as “Company” to keep data of the furniture company and to provide informations for its subclasses. This Company class should have information about branches,branch-employees and customers.Also, We should have a “Branch class” which is extended from Company class to keep branch informations(name of the branch and branch ID).

I used “Users interface” for “Administrator class”,”BranchEmployee class” and “Customer class”. Private variables of the Administrator class are the name and the surname. We should implement getters and setters for name and surname of administrator.I overrode for them .We need an administrator constructor with two-parameters for this class to initialize the name and surname variables.Also,I implemented some methods that are related with administrator-operations(to add and remove branches and branch employees to/from the “branches”linkedlist and “employees” arraylists(max size = 100 as default) in Company class.). In addition to these, I implemented “productQuery” method to see, there are any products that need to be supplied or not, and implemented “addProduct” function for adding products(products that you explained in the hw-1’s pdf) to “products” hybridlist(MAX_NUMBER = 10 for each node) in Company class.).At the beginning of the program,products were added by administrator because there was no branch employee. When administrator added branch employee then this operation is done by branch employees. Also I implemented “showBranchMenu” method to see branches during using of my interactive menu.

Private variables of the Branch class are branchName and branchID. We should implement getters and setters for branchName and branchID of the branch. We need a Branch constructor with one-parameter for this class to initialize the name of the branch. I also implemented a method named as “isInThere” for checking an Branch object is member of branches or not.

Private variables of the BranchEmployee class are name,surname and ID.We should implement getter and setters for name(overrode),surname(overrode) and ID of the branch employee.We need a Branch constructor with three-parameter for this class to initialize branchname,name and surname of branch-employee. I also implemented a method named as “seeFurnitures”,it shows the list of products if their stock number is not equal to 0. I implemented methods for making sales,adding/removing products,informing the manager,accessing products’ informations.I overrode my “isInThere” method to check an Branch object is member of employees or not.

Private variables of the Customer class are name,surname,email,address,phoneNumber,password, total_customer, previousOrders(MAX_NUMBER = 10,type = Furniture),and boughtProductNum. I implemented required getters and setters for these private variables. I used a no-parameter constructor and a constructor with four parameters to initialize the name,surname,email and password.In addition to them, I used functions to buy,search for products,login control,set and see previous orders of a customer.

Hybridlist is a linkedlist that keeps data as arraylists.Each node is an arraylist for a hybridlist.

I used some static-protected lists(arraylist,linkedlist and hybridlist)(MAX_NUMBER= 100 for arraylists and MAX_NUMBER = 10 for each node of hybridlists), static-protected integers to keep numbers of customers and furnitures, I also used a private Admin object for Company class, it’s a general class to keep data of the automation system. I needed just one no-parameter constructor for Company class. I implemented require getters and setters for variables and implemented methods to show customers and list of products.I also implemented some methods for adding customer,searching for products,checking for login operation,buying furniture,setting previous orders of a customer and viewing these previous orders.

Private variables of the Furniture class are product,model,color,productBranch and stock. I implemented required getters and setters for these private variables. I used a constructor with three parameters and a constructor with four parameters to initalize the product,model,color and productBranch variables.

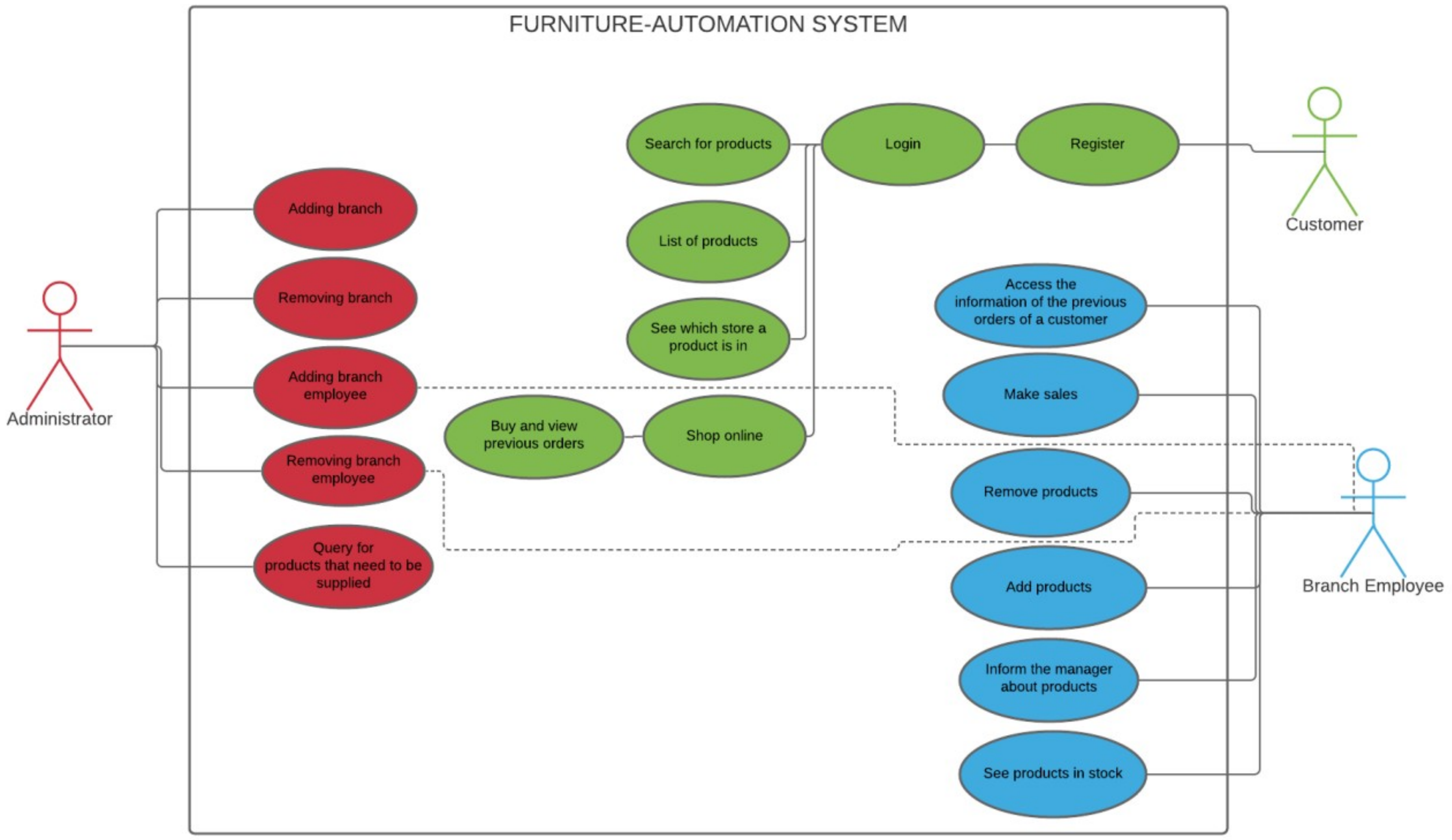
I implemented KWLinkedList and KWArrayList classes from the text book and use them as data structure. I also used both of these data structures together and create a data structure named as HybridList. HybridList class that keeps a LinkedList as a component and the elements stored in the LinkedList are ArrayLists. The number of elements in each ArrayList should be less than MAX_NUMBER. When the number of elements in an ArrayList exceeds MAX_NUMBER a new

ArrayList should be generated in the LinkedList. When there is no element in an ArrayList it should be removed from the LinkedList.

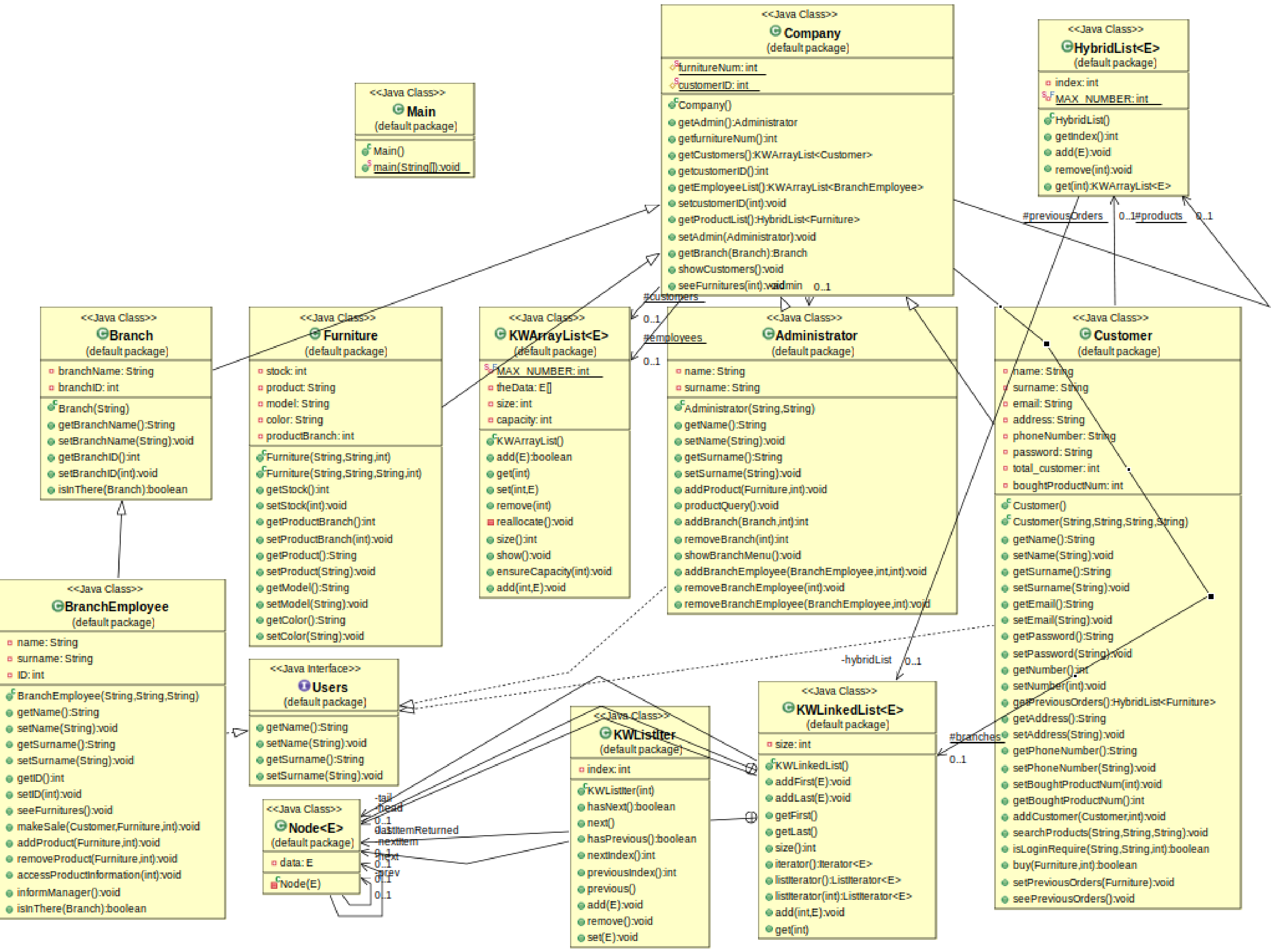
I also needed a main class to test my all automation system so I also implemented it as interactive menu in addition to driver test functions(some of them are tested in interactive menu).As a non-functional requirements, my system does not register the data anywhere so if you close the system the data are lost. I checked every error. If automation system waits for integers then you shouldn't enter a string or character!!!

*I used Ubuntu 20.04-Focal Fossa and compile the Main.java with javac 11.0.10 for this automation system.

2) The Project use case diagrams (extra points):



3) Class diagrams:



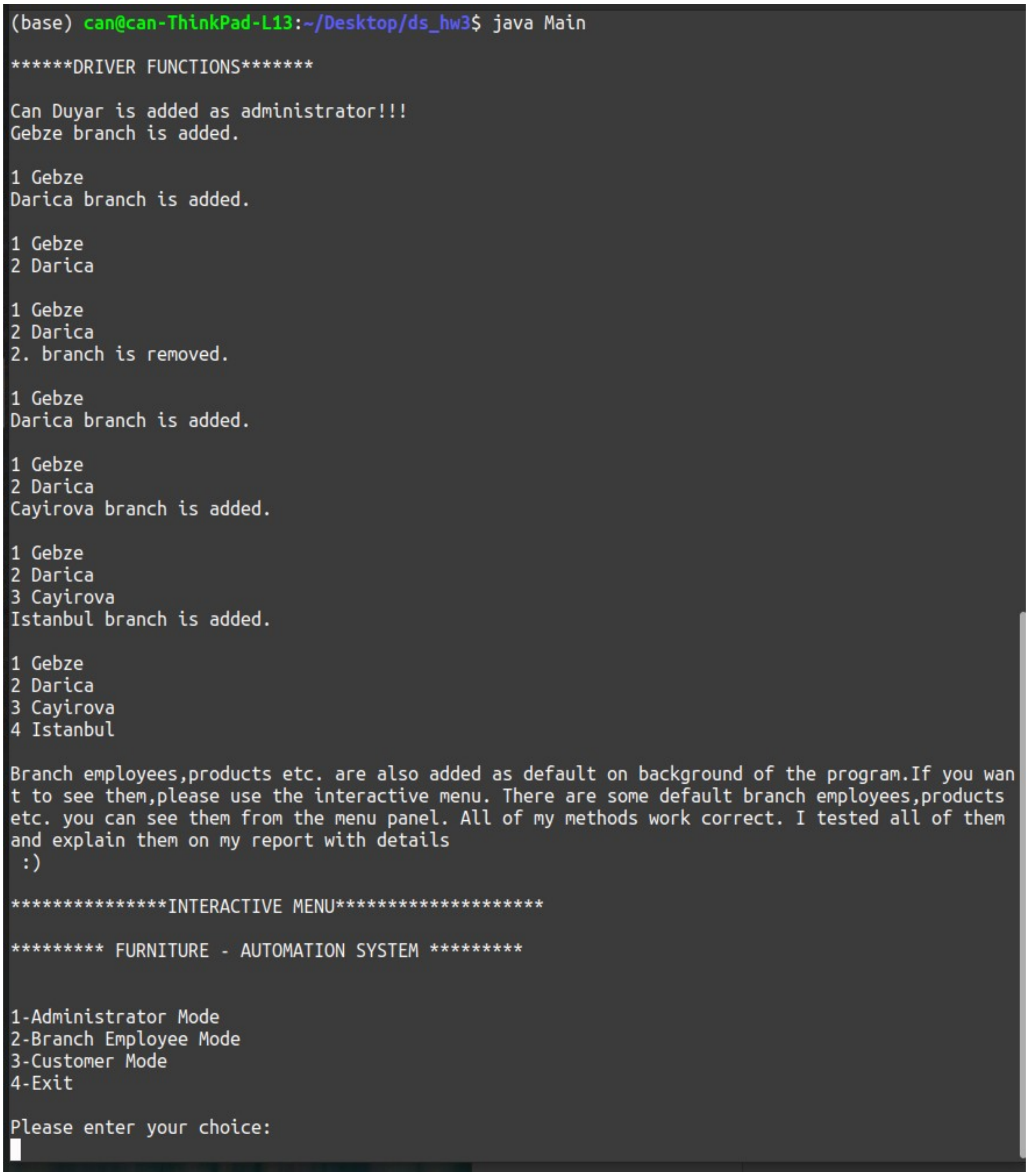
4)Problem solutions approach:

My approach for problem solving is based entirely on object-oriented programming.I planned the design of the entire program for two days before I started coding to combine my old design with KWArrayList,KWLinkedList and HybridList .I created some classes for all program. I used polimorphism,method overriding and inheritance as you explained in the pdf-grading part. I also write an interface for Users.I needed this interface because there was some common methods for all of the Users and I overrode these methods for each of the Users. I used inheritance to create a relation between class structures. I applied error handling for all parts of my automation system as I explained in test cases/running command and results. When I handled the data that are related with branches,employees,customers and products, I used linkedlist class and its listiterator(to solve the traverse problem of linkedlist) class for branch operations,arraylist for operations(administrator,branch-employee,customer) and HybridList for furniture operations . That's why my program design is based on arraylist,linkedlist and hybridlist.

These list structures' types are Furniture,Customer,BranchEmployee and Branch so each of elements of these lists are object. I solved keeping data problem with this way. Each of object has their own methods(I explained each of methods in system requirements part) so I handled operations with this way. In test part, I designed a menu which is interactive so we can test all of the program with this menu. There is a main menu and some submenus to handle operations of user-types. I also write some driver test methods except my interactive menu to see some results before using the interactive menu. I solved test problems with using both of them. I used inheritance for my classes to avoid from the rewrite codes.

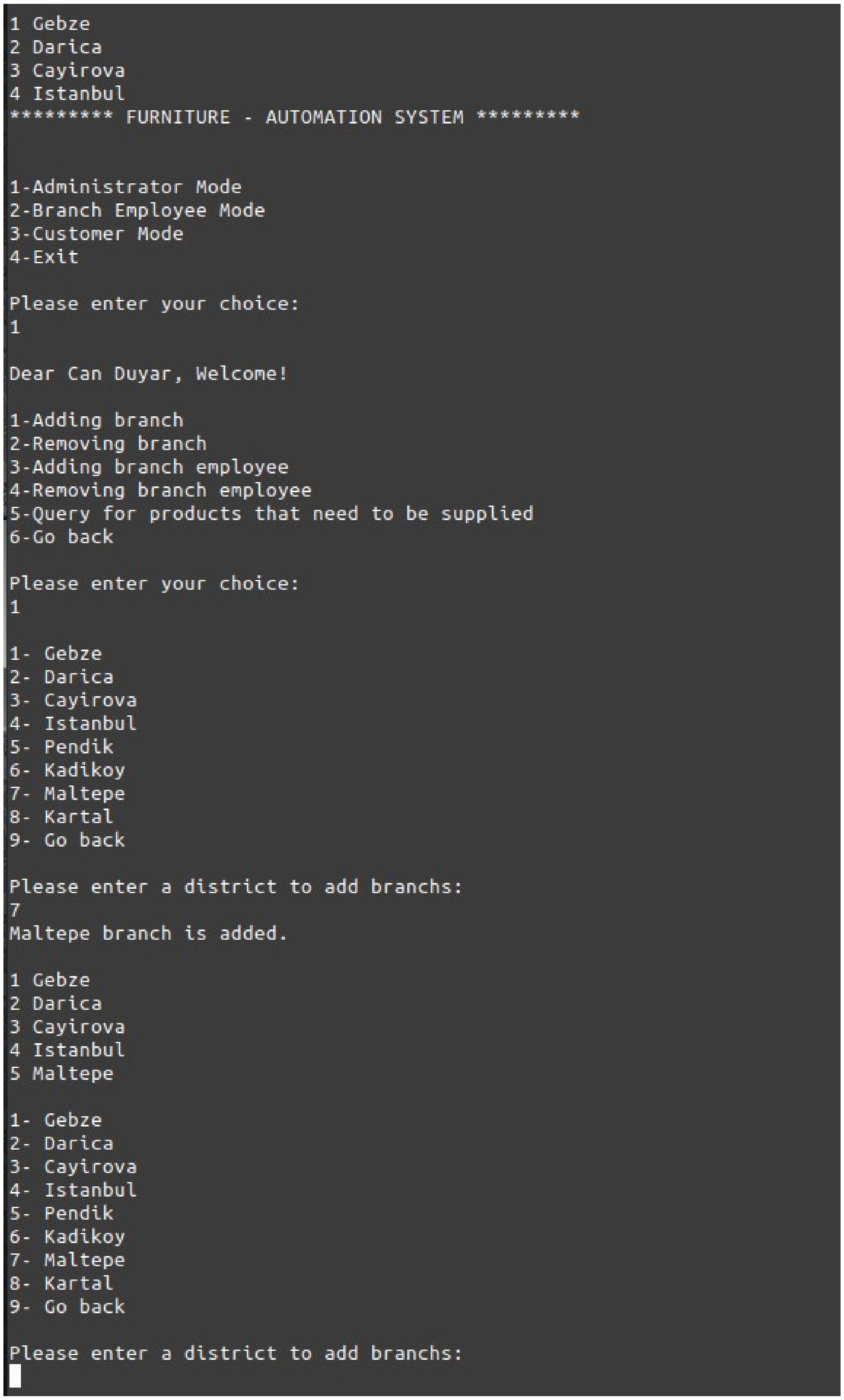
5) Test cases/Running command and results: (I used same tests for my linkedlist,arraylist and hybridlist based automation system-program)

*At the beginning of the program, some branches are added automatically because there are 4 branches(In my example they are Gebze,Darıca,Cayirova,Istanbul) according to pdf instructions.
*At the beginning of the program administrator is added as “Can Duyar” and also products(from different colors and models) are added automatically according to pdf instructions.

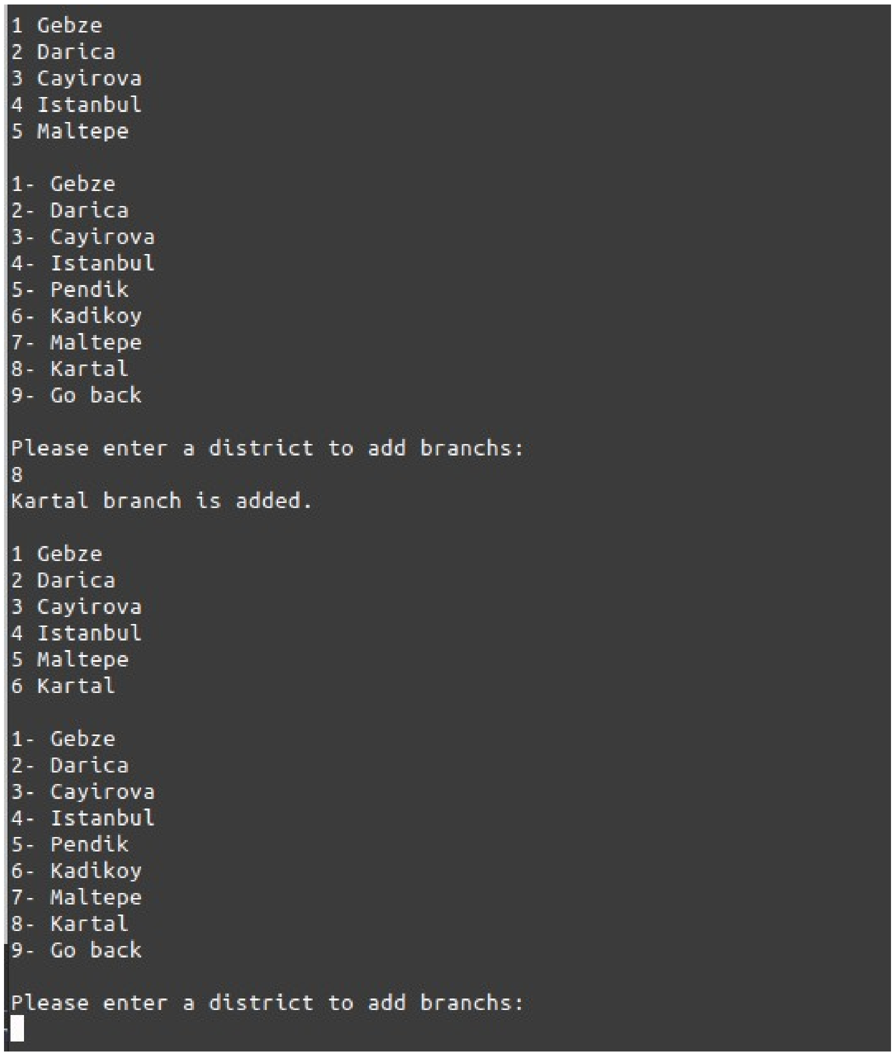


```
(base) can@can-ThinkPad-L13:~/Desktop/ds_hw3$ java Main

******DRIVER FUNCTIONS*******

Can Duyar is added as administrator!!!
Gebze branch is added.

1 Gebze
Darica branch is added.

1 Gebze
2 Darica

1 Gebze
2 Darica
2. branch is removed.

1 Gebze
Darica branch is added.

1 Gebze
2 Darica
Cayirova branch is added.

1 Gebze
2 Darica
3 Cayirova
Istanbul branch is added.

1 Gebze
2 Darica
3 Cayirova
4 Istanbul

Branch employees,products etc. are also added as default on background of the program.If you wan
t to see them,please use the interactive menu. There are some default branch employees,products
etc. you can see them from the menu panel. All of my methods work correct. I tested all of them
and explain them on my report with details
 :)

***************INTERACTIVE MENU********************

********* FURNITURE - AUTOMATION SYSTEM *********


1-Administrator Mode
2-Branch Employee Mode
3-Customer Mode
4-Exit

Please enter your choice:
```

Test Case-1: Adding a new branch (Admin): passed

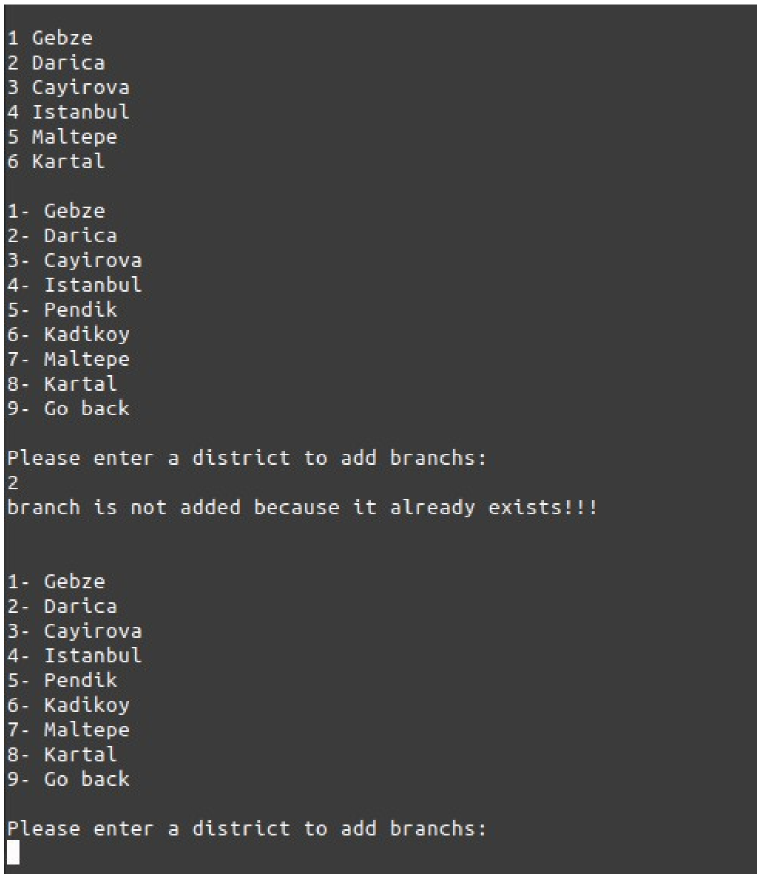


```
1 Gebze
2 Darica
3 Cayirova
4 Istanbul
********* FURNITURE - AUTOMATION SYSTEM *********


1-Administrator Mode
2-Branch Employee Mode
3-Customer Mode
4-Exit

Please enter your choice:
1

Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
1

1- Gebze
2- Darica
3- Cayirova
4- Istanbul
5- Pendik
6- Kadikoy
7- Maltepe
8- Kartal
9- Go back

Please enter a district to add branchs:
7
Maltepe branch is added.

1 Gebze
2 Darica
3 Cayirova
4 Istanbul
5 Maltepe

1- Gebze
2- Darica
3- Cayirova
4- Istanbul
5- Pendik
6- Kadikoy
7- Maltepe
8- Kartal
9- Go back

Please enter a district to add branchs:
```



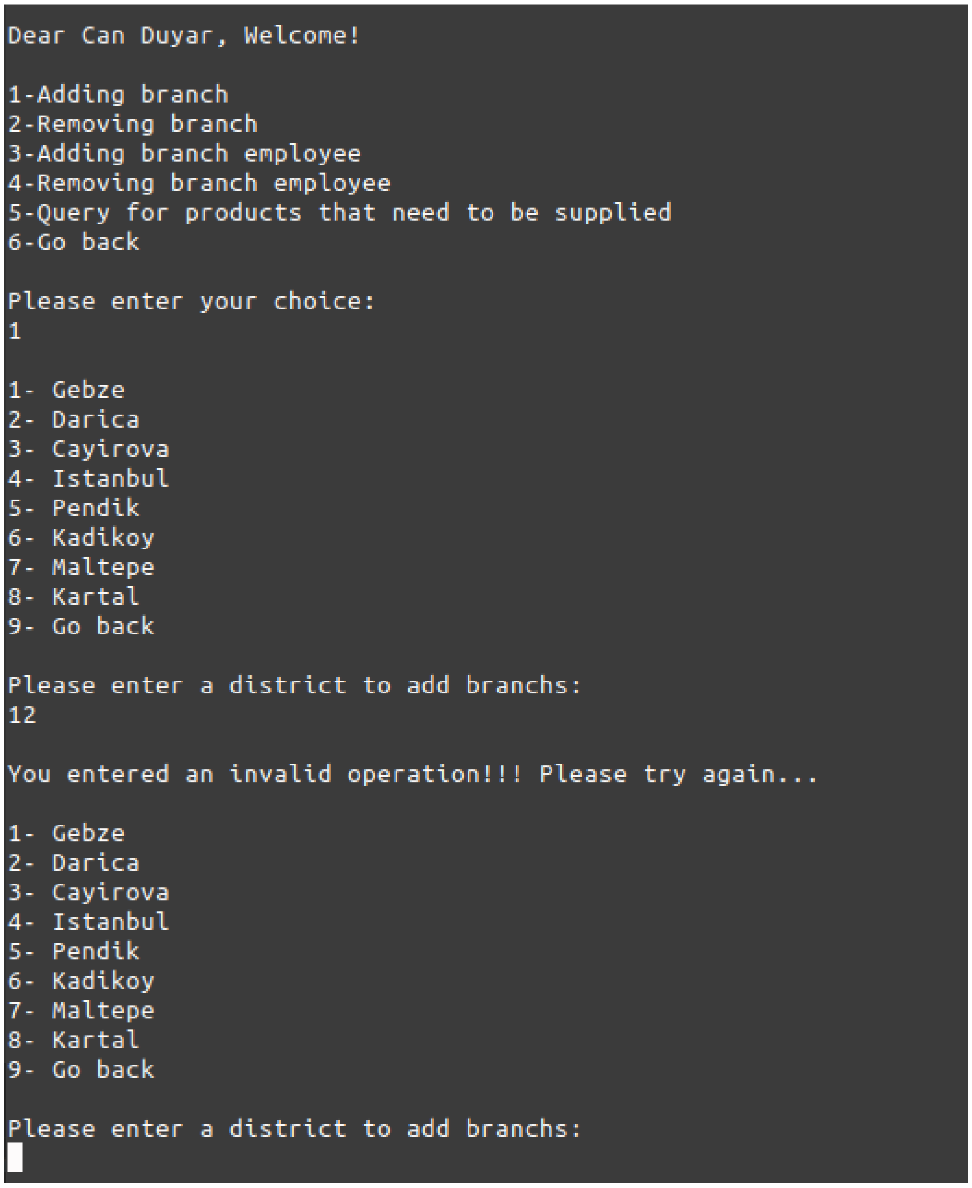
```
1 Gebze
2 Darica
3 Cayirova
4 Istanbul
5 Maltepe

1- Gebze
2- Darica
3- Cayirova
4- Istanbul
5- Pendik
6- Kadikoy
7- Maltepe
8- Kartal
9- Go back

Please enter a district to add branchs:
8
Kartal branch is added.

1 Gebze
2 Darica
3 Cayirova
4 Istanbul
5 Maltepe
6 Kartal

1- Gebze
2- Darica
3- Cayirova
4- Istanbul
5- Pendik
6- Kadikoy
7- Maltepe
8- Kartal
9- Go back

Please enter a district to add branchs:
```

Test Case-2: Trying to add existing branch (Admin)- It gives error: passed



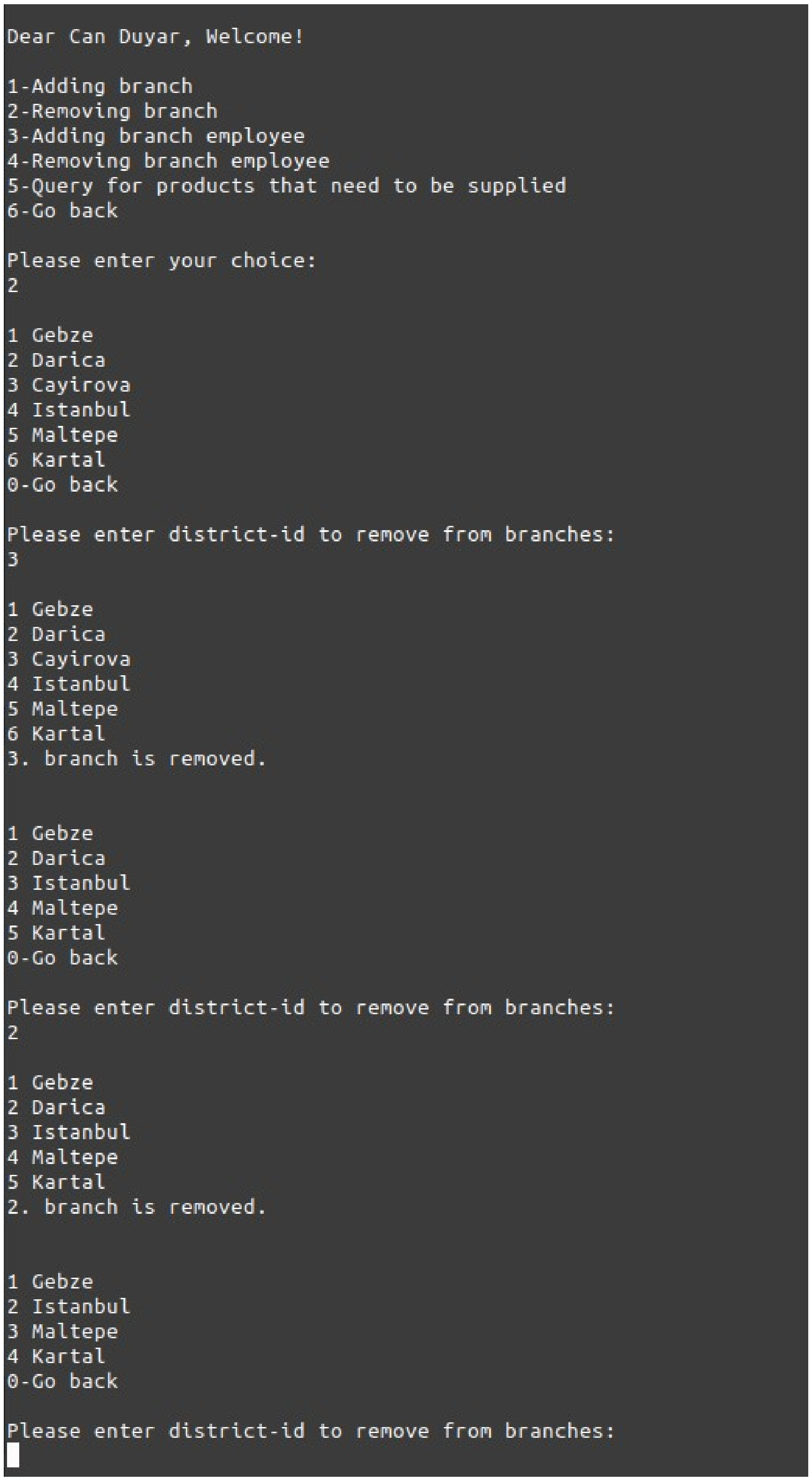
```
1 Gebze
2 Darica
3 Cayirova
4 Istanbul
5 Maltepe
6 Kartal

1- Gebze
2- Darica
3- Cayirova
4- Istanbul
5- Pendik
6- Kadikoy
7- Maltepe
8- Kartal
9- Go back

Please enter a district to add branchs:
2
branch is not added because it already exists!!!


1- Gebze
2- Darica
3- Cayirova
4- Istanbul
5- Pendik
6- Kadikoy
7- Maltepe
8- Kartal
9- Go back

Please enter a district to add branchs:
```

Test Case-3: Trying to add a branch that has out of branch ids (Admin)- It gives error: passed



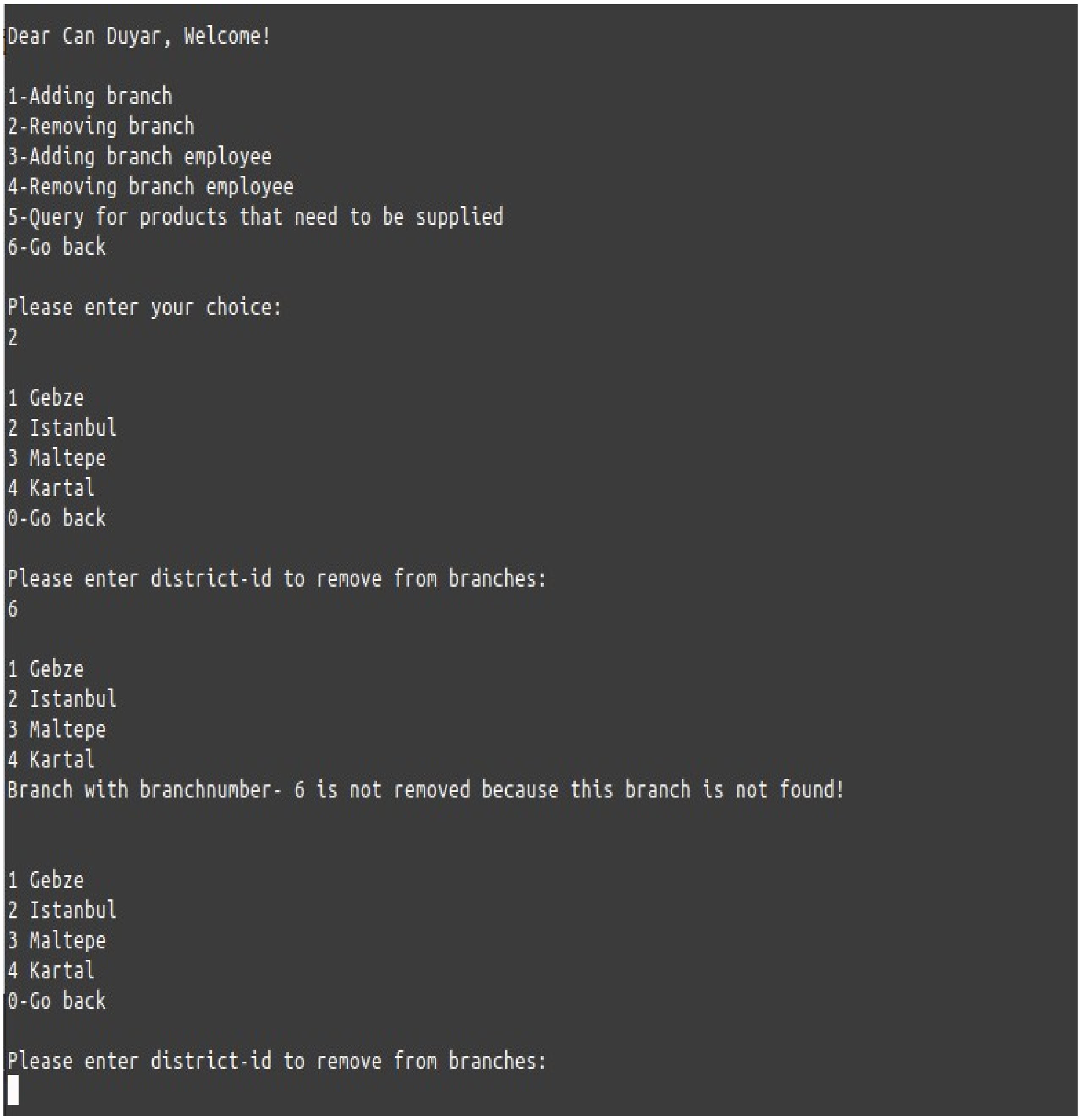
```
Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
1

1- Gebze
2- Darica
3- Cayirova
4- Istanbul
5- Pendik
6- Kadikoy
7- Maltepe
8- Kartal
9- Go back

Please enter a district to add branchs:
12

You entered an invalid operation!!! Please try again...

1- Gebze
2- Darica
3- Cayirova
4- Istanbul
5- Pendik
6- Kadikoy
7- Maltepe
8- Kartal
9- Go back

Please enter a district to add branchs:
```

Test Case-4:Removing a branch (Admin): passed

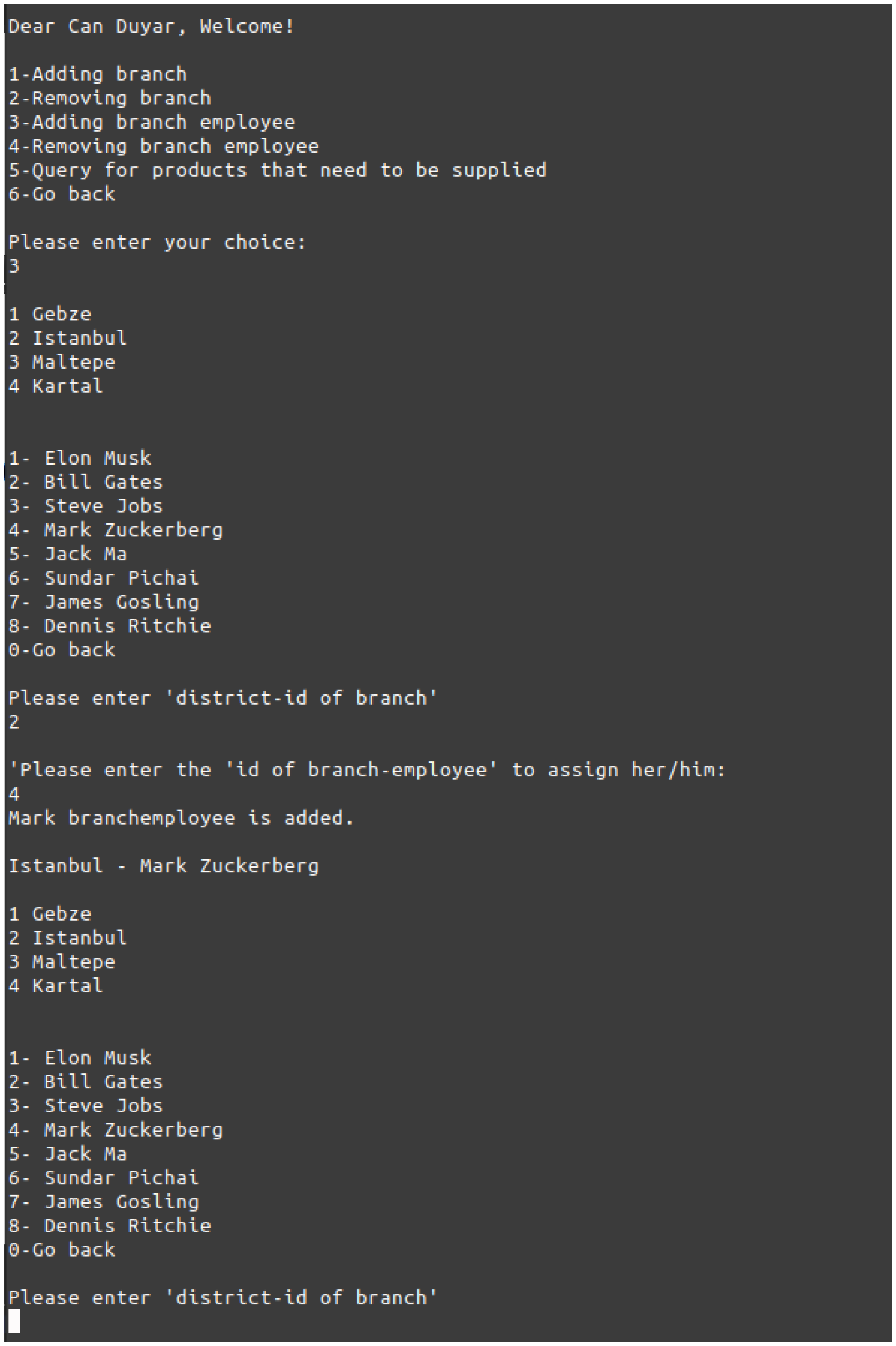


```
Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
2

1 Gebze
2 Darica
3 Cayirova
4 Istanbul
5 Maltepe
6 Kartal
0-Go back

Please enter district-id to remove from branches:
3

1 Gebze
2 Darica
3 Cayirova
4 Istanbul
5 Maltepe
6 Kartal
3. branch is removed.


1 Gebze
2 Darica
3 Istanbul
4 Maltepe
5 Kartal
0-Go back

Please enter district-id to remove from branches:
2

1 Gebze
2 Darica
3 Istanbul
4 Maltepe
5 Kartal
2. branch is removed.


1 Gebze
2 Istanbul
3 Maltepe
4 Kartal
0-Go back

Please enter district-id to remove from branches:
```

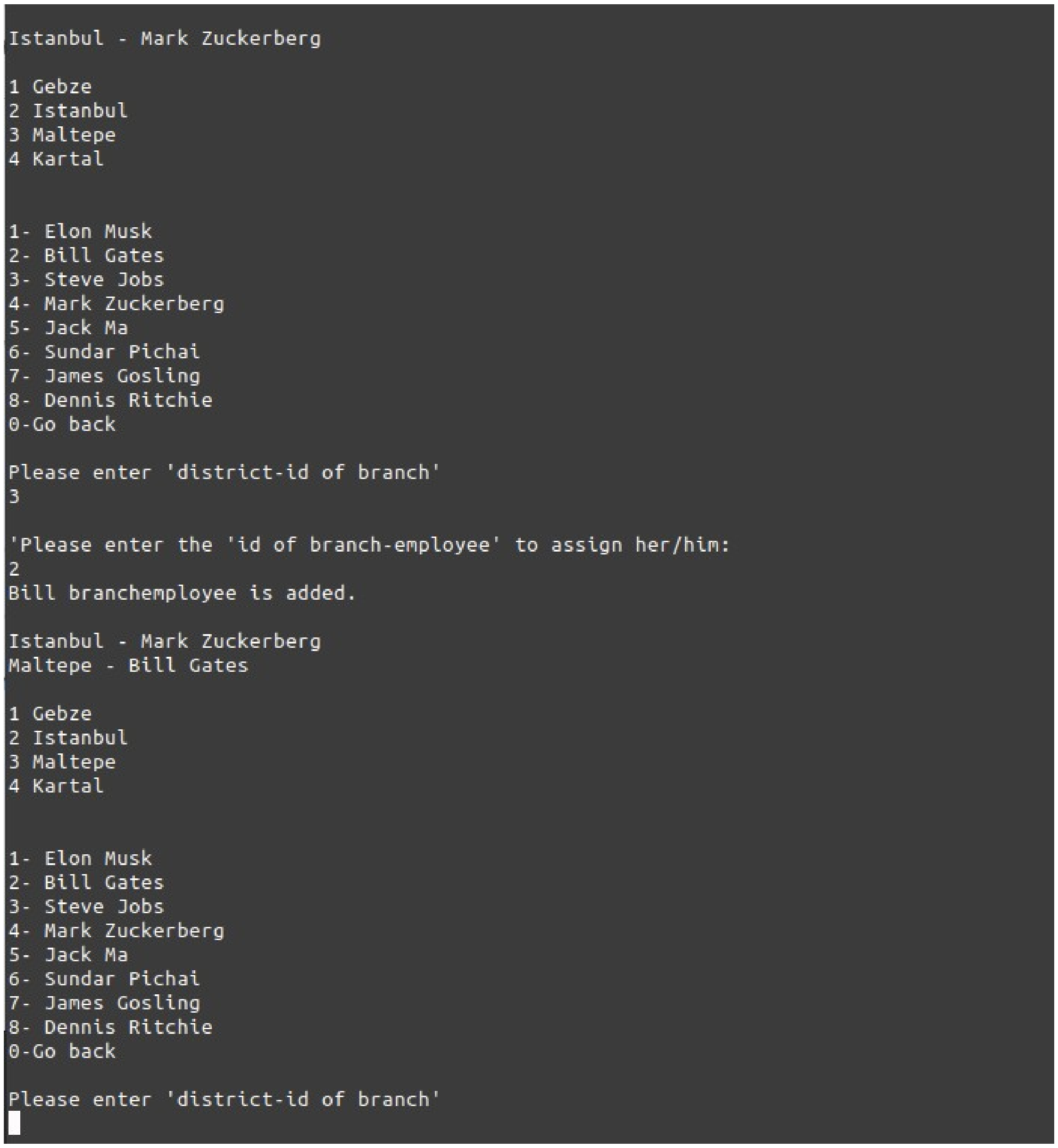
Test Case-5: Trying to remove non-existing branch (Admin)- It gives error: passed


```
Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
2

1 Gebze
2 Istanbul
3 Maltepe
4 Kartal
0-Go back

Please enter district-id to remove from branches:
6

1 Gebze
2 Istanbul
3 Maltepe
4 Kartal
Branch with branchnumber- 6 is not removed because this branch is not found!


1 Gebze
2 Istanbul
3 Maltepe
4 Kartal
0-Go back

Please enter district-id to remove from branches:
```

Test Case-6: Adding a branch employee (Admin): passed

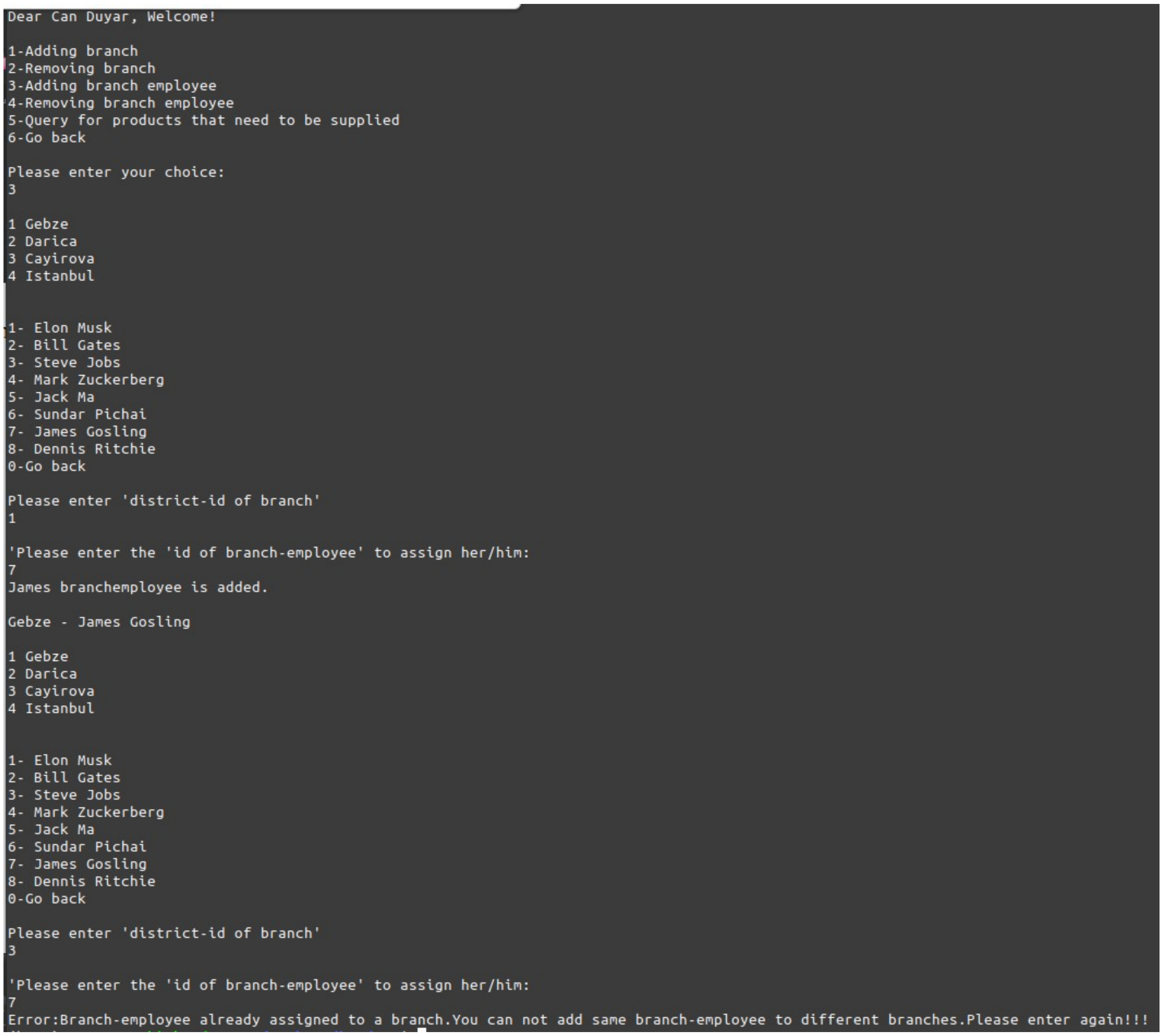


```
Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
3

1 Gebze
2 Istanbul
3 Maltepe
4 Kartal


1-  Elon Musk
2-  Bill Gates
3-  Steve Jobs
4-  Mark Zuckerberg
5-  Jack Ma
6-  Sundar Pichai
7-  James Gosling
8-  Dennis Ritchie
0-Go back

Please enter 'district-id of branch'
2

'Please enter the 'id of branch-employee' to assign her/him:
4
Mark branchemployee is added.

Istanbul - Mark Zuckerberg

1 Gebze
2 Istanbul
3 Maltepe
4 Kartal


1-  Elon Musk
2-  Bill Gates
3-  Steve Jobs
4-  Mark Zuckerberg
5-  Jack Ma
6-  Sundar Pichai
7-  James Gosling
8-  Dennis Ritchie
0-Go back

Please enter 'district-id of branch'
```



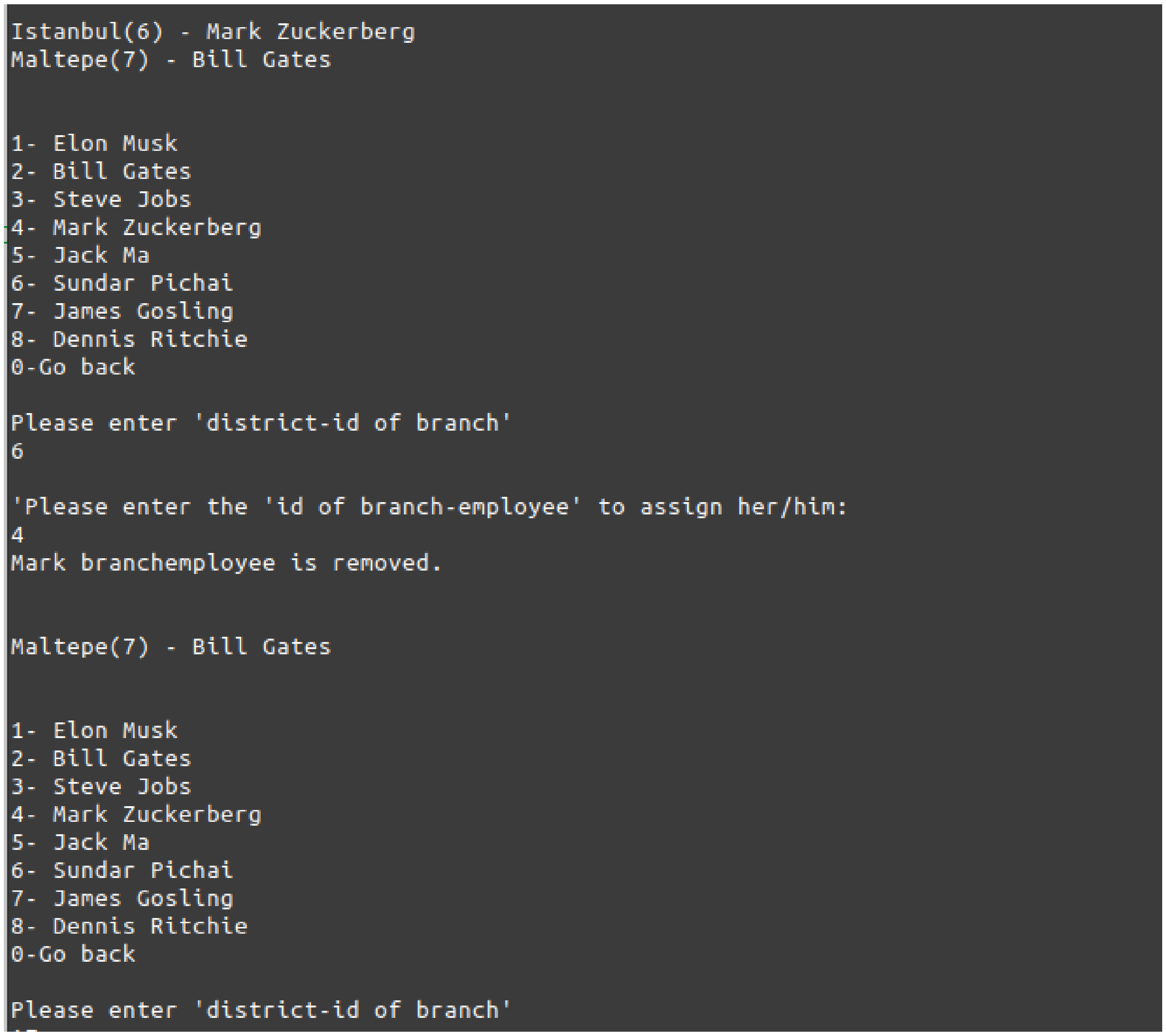
```
Istanbul - Mark Zuckerberg

1 Gebze
2 Istanbul
3 Maltepe
4 Kartal


1- Elon Musk
2- Bill Gates
3- Steve Jobs
4- Mark Zuckerberg
5- Jack Ma
6- Sundar Pichai
7- James Gosling
8- Dennis Ritchie
0-Go back

Please enter 'district-id of branch'
3

'Please enter the 'id of branch-employee' to assign her/him:
2
Bill branchemployee is added.

Istanbul - Mark Zuckerberg
Maltepe - Bill Gates

1 Gebze
2 Istanbul
3 Maltepe
4 Kartal


1- Elon Musk
2- Bill Gates
3- Steve Jobs
4- Mark Zuckerberg
5- Jack Ma
6- Sundar Pichai
7- James Gosling
8- Dennis Ritchie
0-Go back

Please enter 'district-id of branch'
```

Test Case-7: Trying to add same branch employee to different branches (Admin) – it gives error: passed



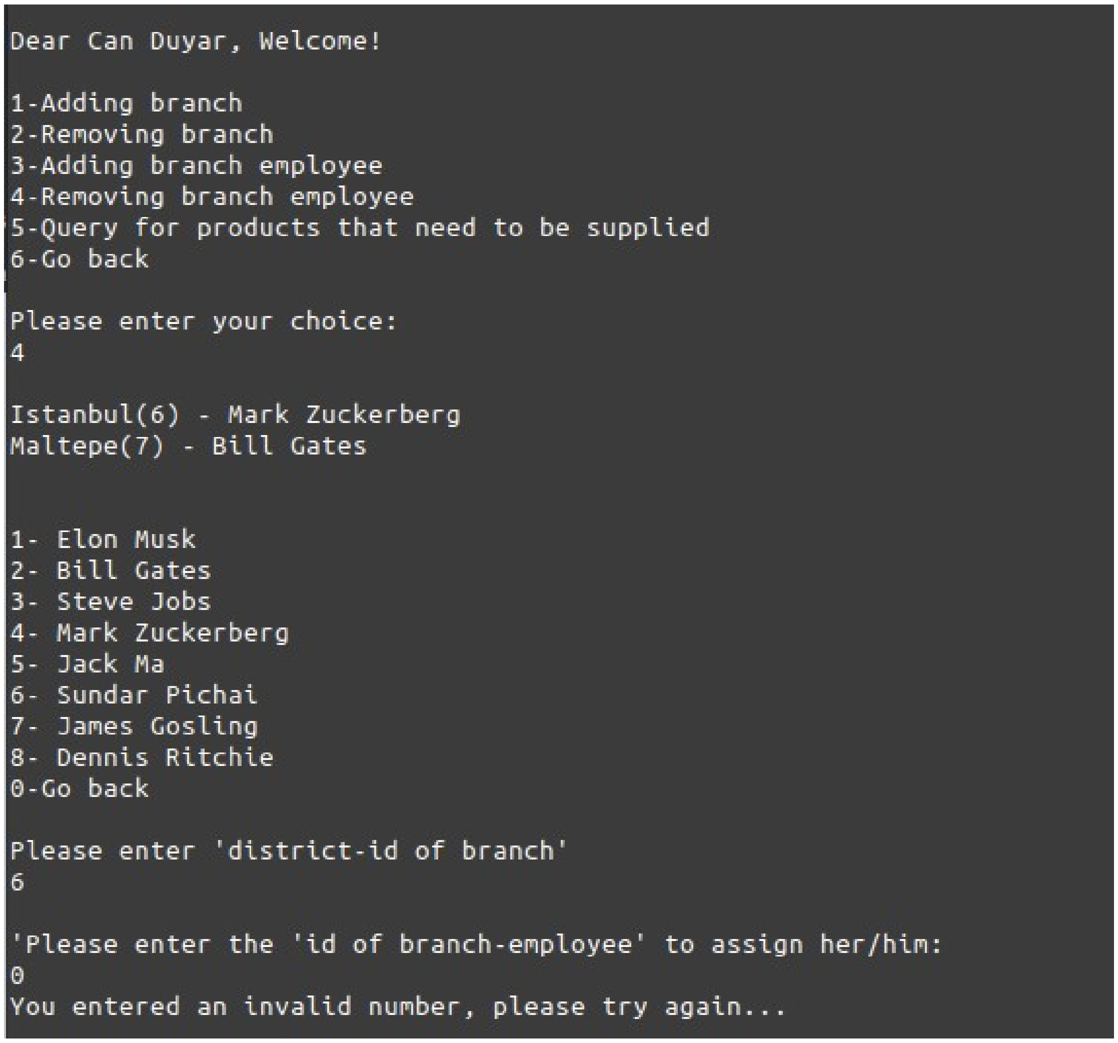
```
Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
3

1 Gebze
2 Darica
3 Cayirova
4 Istanbul


1- Elon Musk
2- Bill Gates
3- Steve Jobs
4- Mark Zuckerberg
5- Jack Ma
6- Sundar Pichai
7- James Gosling
8- Dennis Ritchie
0-Go back

Please enter 'district-id of branch'
1

'Please enter the 'id of branch-employee' to assign her/him:
7
James branchemployee is added.

Gebze - James Gosling

1 Gebze
2 Darica
3 Cayirova
4 Istanbul


1- Elon Musk
2- Bill Gates
3- Steve Jobs
4- Mark Zuckerberg
5- Jack Ma
6- Sundar Pichai
7- James Gosling
8- Dennis Ritchie
0-Go back

Please enter 'district-id of branch'
3

'Please enter the 'id of branch-employee' to assign her/him:
7
Error:Branch-employee already assigned to a branch.You can not add same branch-employee to different branches.Please enter again!!!
```

Test Case-8: Removing a branch employee (Admin): passed



```
Istanbul(6) - Mark Zuckerberg
Maltepe(7) - Bill Gates


1- Elon Musk
2- Bill Gates
3- Steve Jobs
4- Mark Zuckerberg
5- Jack Ma
6- Sundar Pichai
7- James Gosling
8- Dennis Ritchie
0-Go back

Please enter 'district-id of branch'
6

'Please enter the 'id of branch-employee' to assign her/him:
4
Mark branchemployee is removed.


Maltepe(7) - Bill Gates


1- Elon Musk
2- Bill Gates
3- Steve Jobs
4- Mark Zuckerberg
5- Jack Ma
6- Sundar Pichai
7- James Gosling
8- Dennis Ritchie
0-Go back

Please enter 'district-id of branch'
```

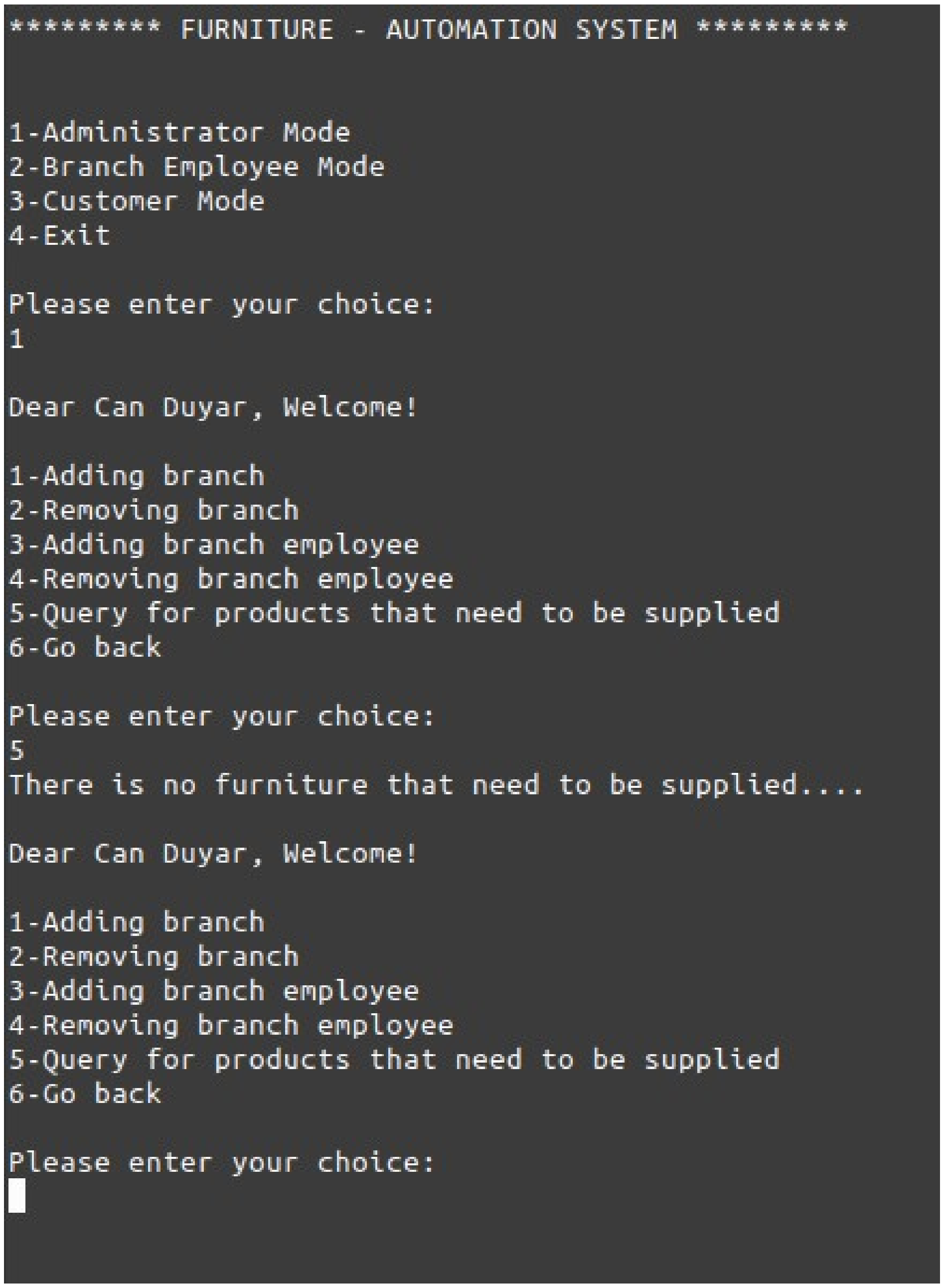
Test Case-9: Trying to remove a branch employee who is not exist(Admin) – it gives error: passed



```
Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
4

Istanbul(6) - Mark Zuckerberg
Maltepe(7) - Bill Gates


1- Elon Musk
2- Bill Gates
3- Steve Jobs
4- Mark Zuckerberg
5- Jack Ma
6- Sundar Pichai
7- James Gosling
8- Dennis Ritchie
0-Go back

Please enter 'district-id of branch'
6

'Please enter the 'id of branch-employee' to assign her/him:
0
You entered an invalid number, please try again...
```

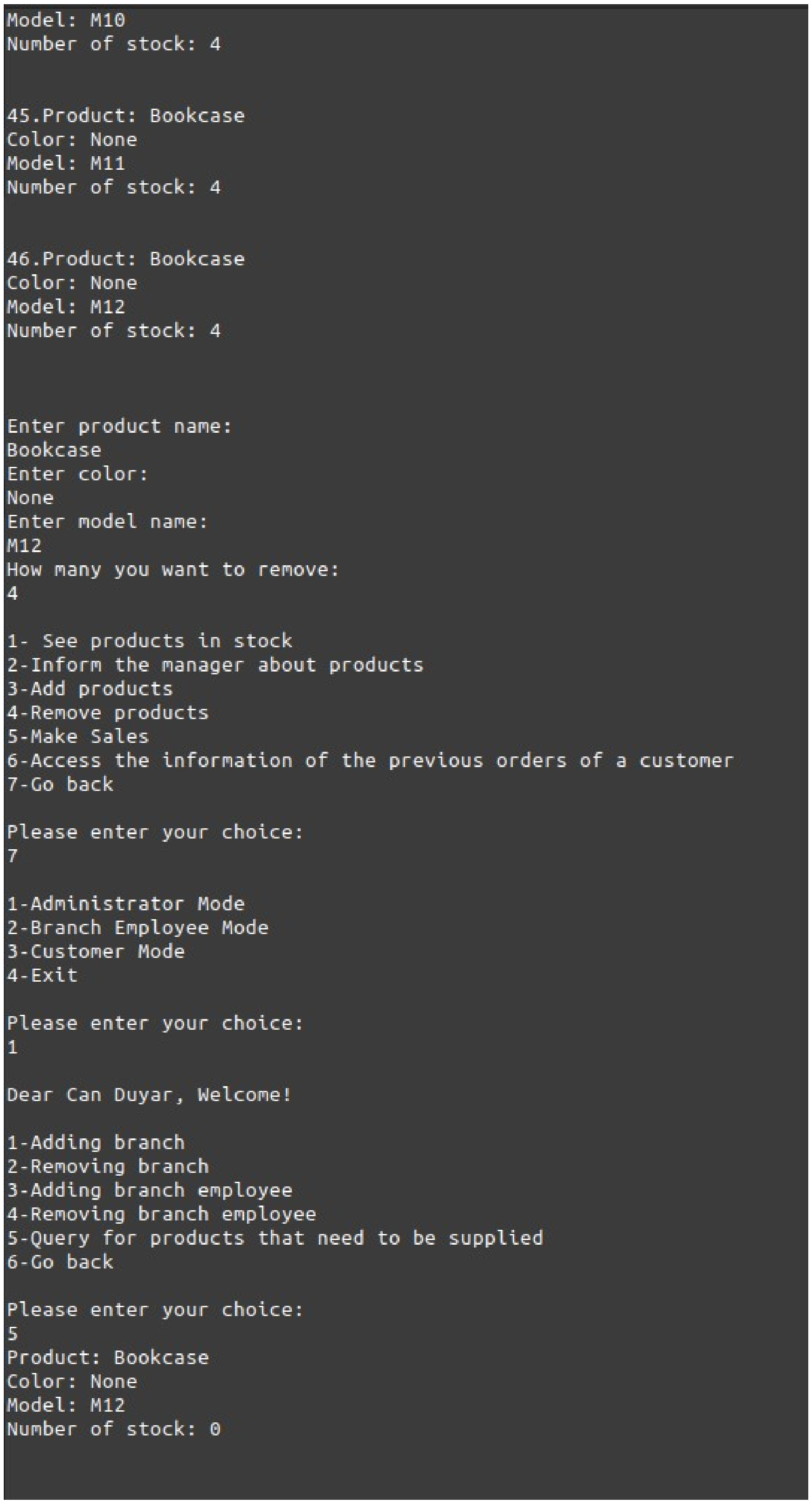
Test Case-10: Query for products that need to be supplied(Admin) : passed

*If all stock is full,for example; first time of the program…



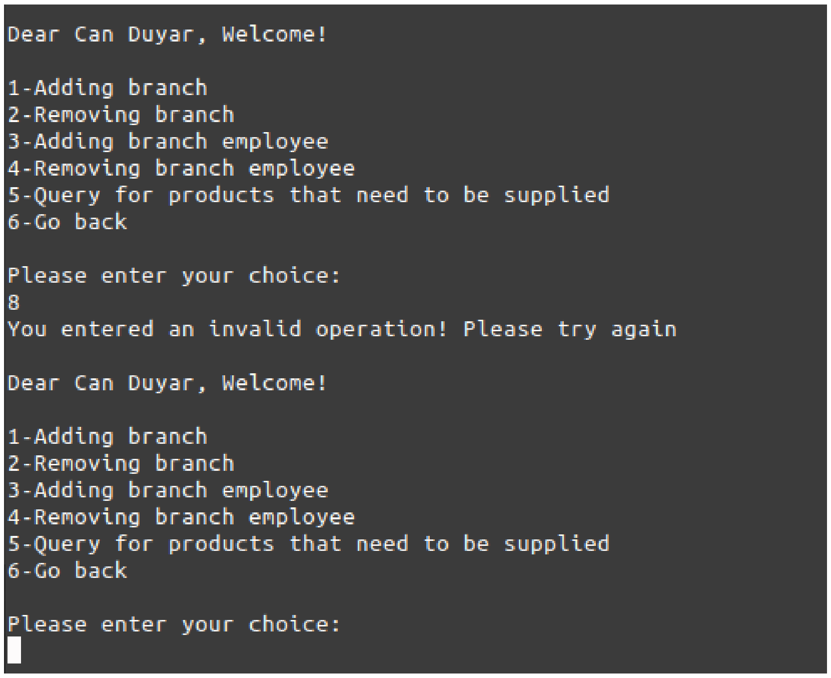
```
********* FURNITURE - AUTOMATION SYSTEM *********


1-Administrator Mode
2-Branch Employee Mode
3-Customer Mode
4-Exit

Please enter your choice:
1

Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
5
There is no furniture that need to be supplied....

Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
```

*If at least one stock is empty,for example; after a branch employee removes all stock of a product…



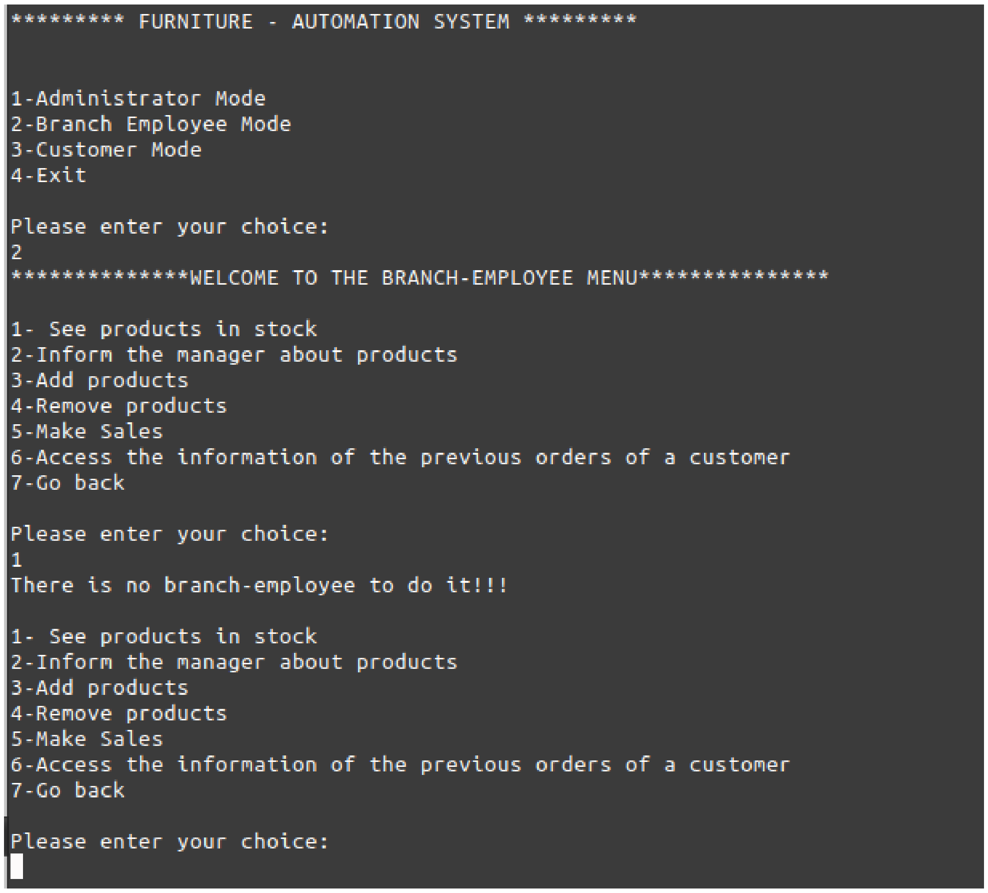
```
Model: M10
Number of stock: 4


45.Product: Bookcase
Color: None
Model: M11
Number of stock: 4


46.Product: Bookcase
Color: None
Model: M12
Number of stock: 4



Enter product name:
Bookcase
Enter color:
None
Enter model name:
M12
How many you want to remove:
4

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
7

1-Administrator Mode
2-Branch Employee Mode
3-Customer Mode
4-Exit

Please enter your choice:
1

Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
5
Product: Bookcase
Color: None
Model: M12
Number of stock: 0
```

Test Case-11: If administrator enters a number out of options(Admin)-it gives error : passed



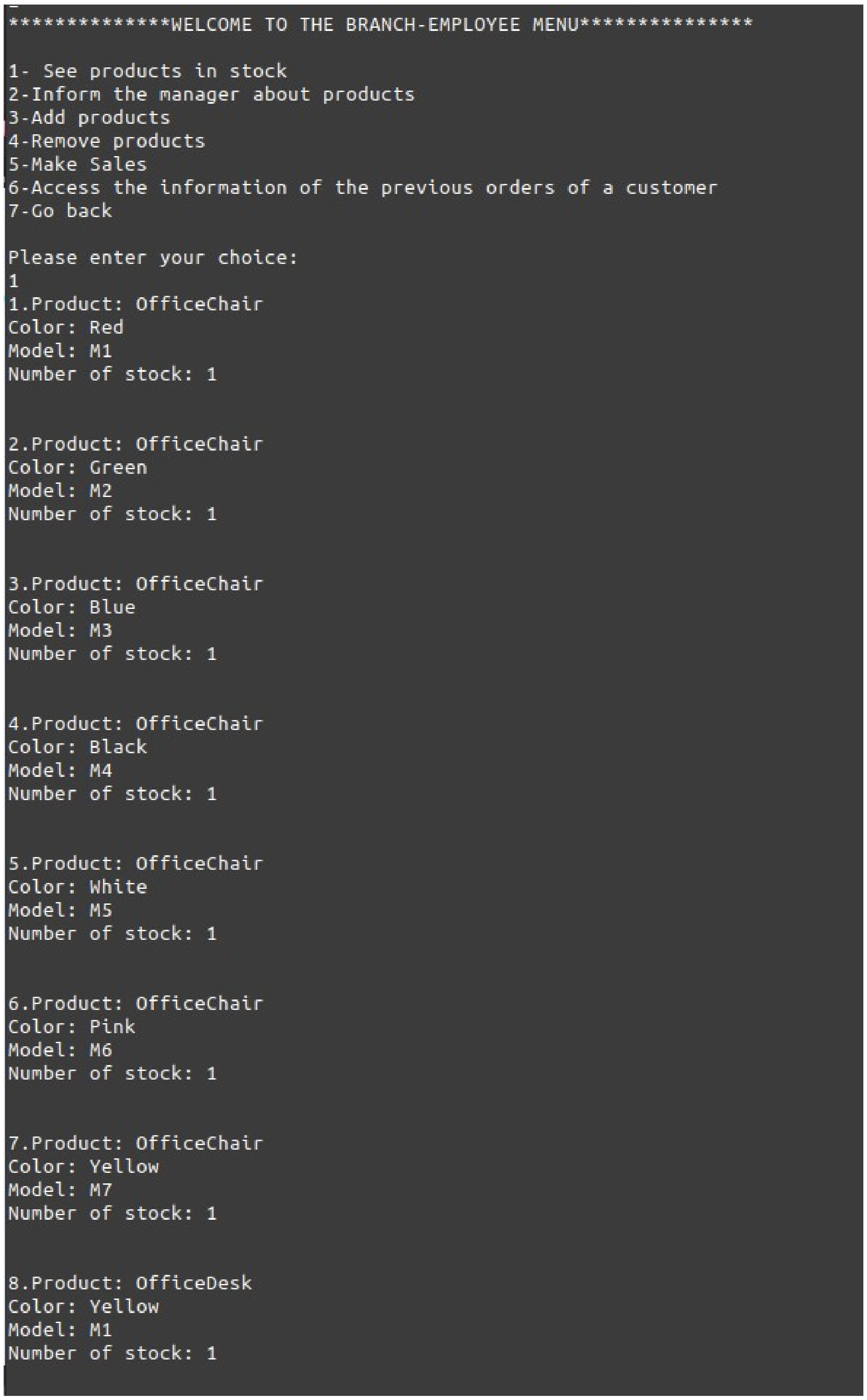
```
Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
8
You entered an invalid operation! Please try again

Dear Can Duyar, Welcome!

1-Adding branch
2-Removing branch
3-Adding branch employee
4-Removing branch employee
5-Query for products that need to be supplied
6-Go back

Please enter your choice:
```

Test Case-12: If we try do a branch-employee operation before adding a branch-employee – it gives error because there is no branch employee at the beginning of the program. First you should add at least one branch employee with the help of administrator then you can do branch employee's operation properly : passed

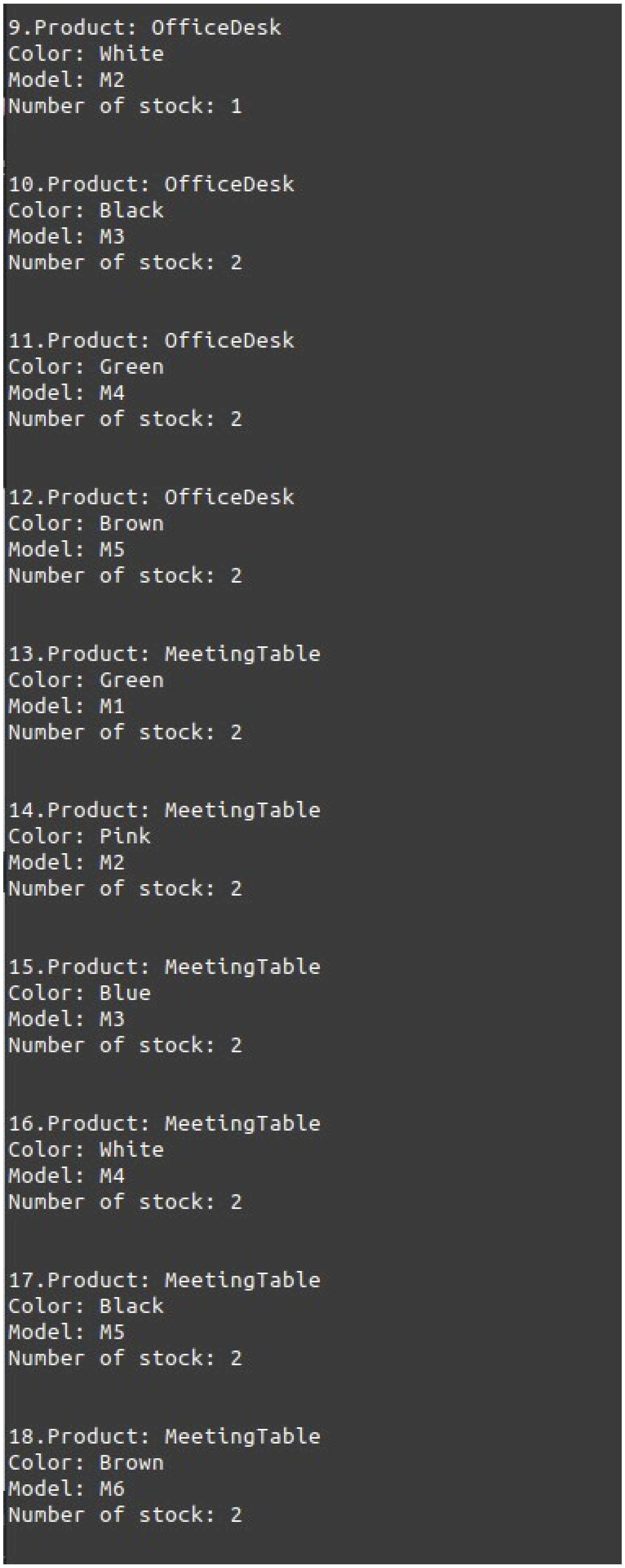


```
********* FURNITURE - AUTOMATION SYSTEM *********


1-Administrator Mode
2-Branch Employee Mode
3-Customer Mode
4-Exit

Please enter your choice:
2
**************WELCOME TO THE BRANCH-EMPLOYEE MENU***************

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
1
There is no branch-employee to do it!!!

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
```

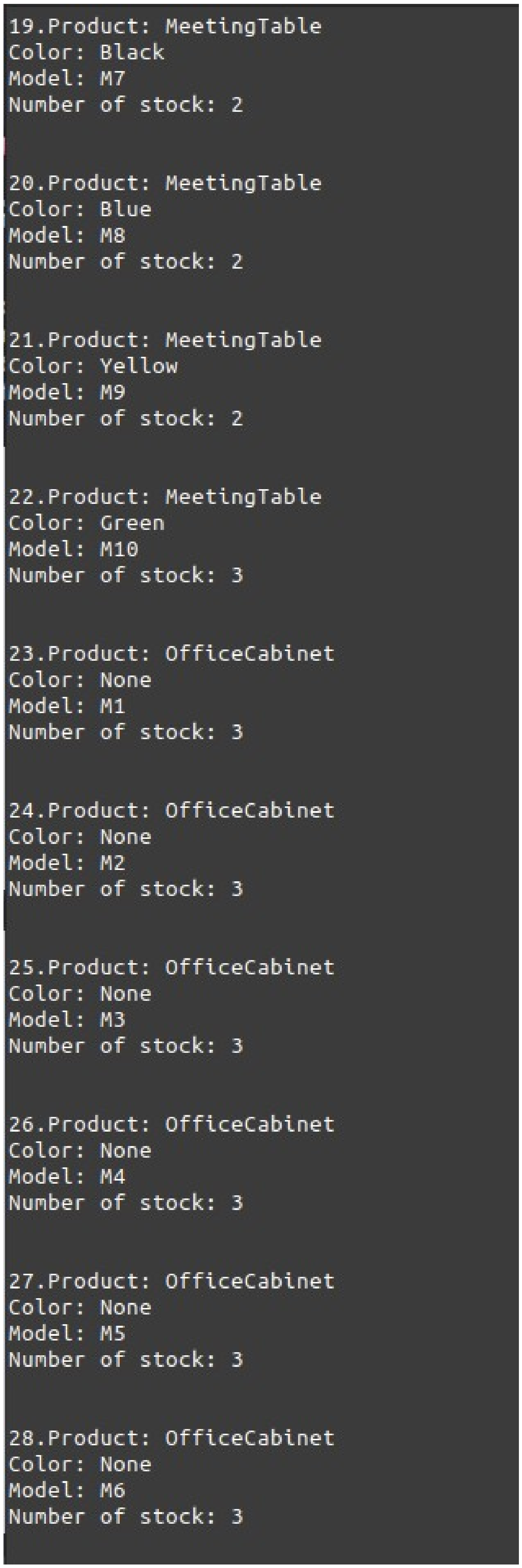
Test Case-13: See products in the store(BranchEmployee): passed
-The product list is too long due to the number of products. I added these products at the beginning of the program according to pdf instructions.



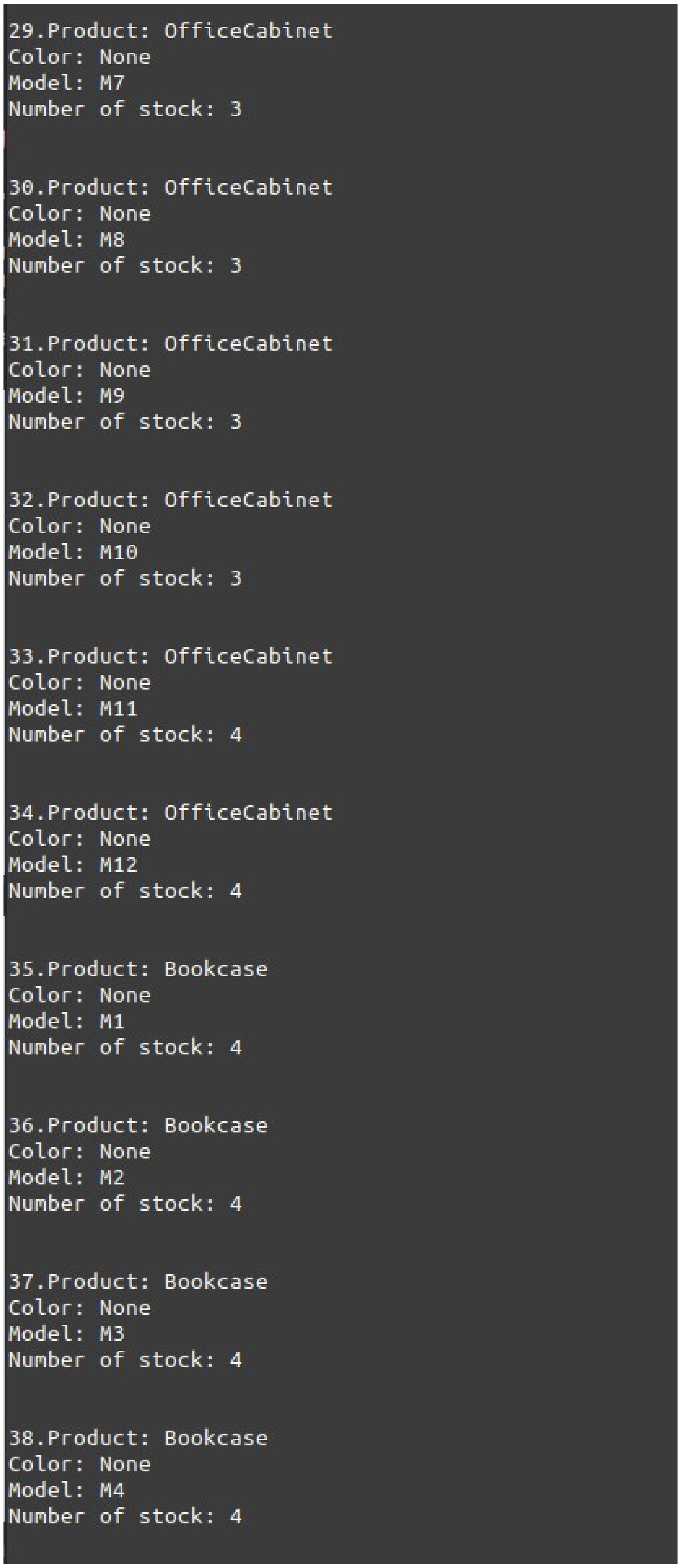
```
**************WELCOME TO THE BRANCH-EMPLOYEE MENU***************

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
1
1.Product: OfficeChair
Color: Red
Model: M1
Number of stock: 1


2.Product: OfficeChair
Color: Green
Model: M2
Number of stock: 1


3.Product: OfficeChair
Color: Blue
Model: M3
Number of stock: 1


4.Product: OfficeChair
Color: Black
Model: M4
Number of stock: 1


5.Product: OfficeChair
Color: White
Model: M5
Number of stock: 1


6.Product: OfficeChair
Color: Pink
Model: M6
Number of stock: 1


7.Product: OfficeChair
Color: Yellow
Model: M7
Number of stock: 1


8.Product: OfficeDesk
Color: Yellow
Model: M1
Number of stock: 1
```



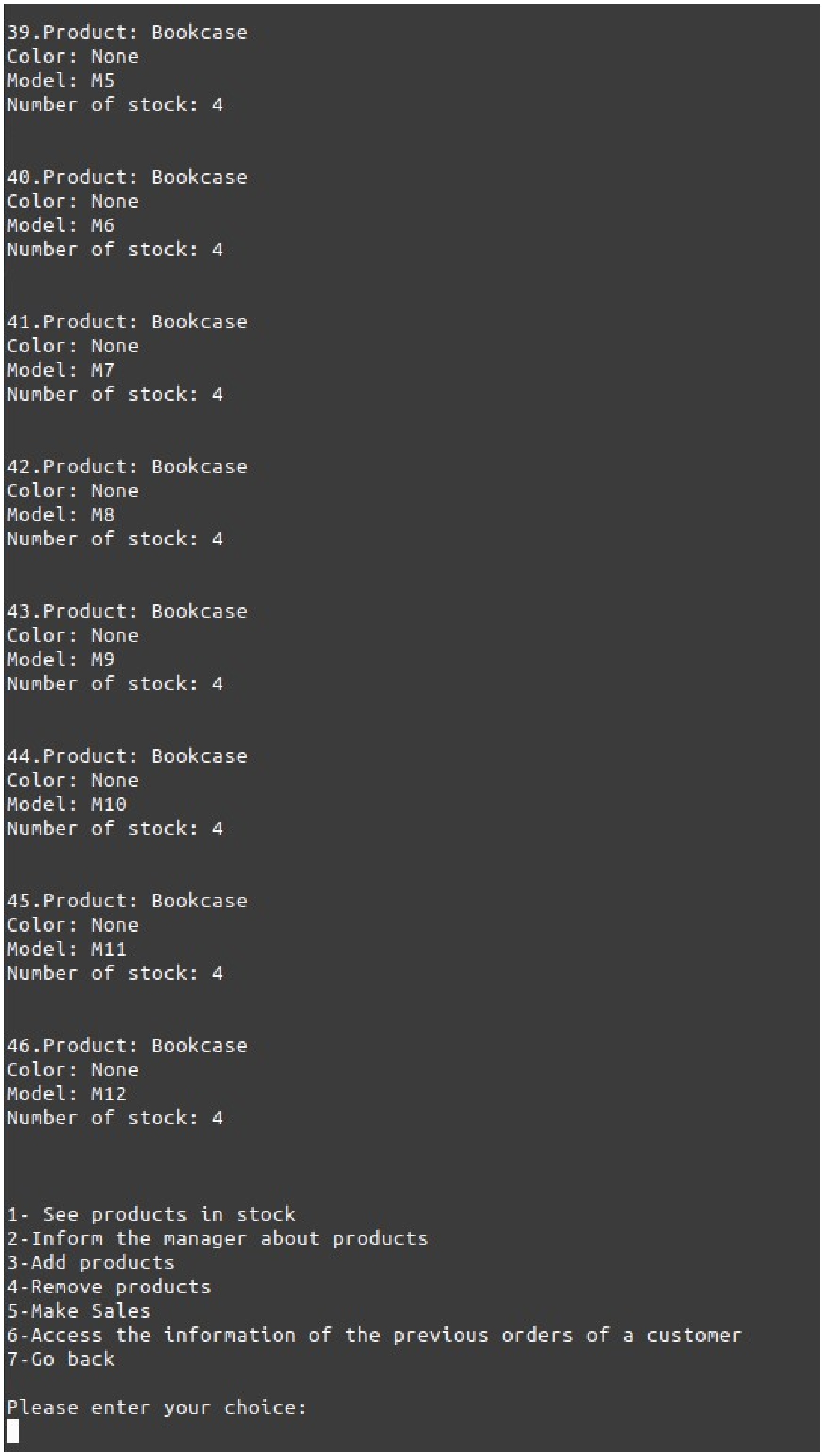
```
9.Product: OfficeDesk
Color: White
Model: M2
Number of stock: 1


10.Product: OfficeDesk
Color: Black
Model: M3
Number of stock: 2


11.Product: OfficeDesk
Color: Green
Model: M4
Number of stock: 2


12.Product: OfficeDesk
Color: Brown
Model: M5
Number of stock: 2


13.Product: MeetingTable
Color: Green
Model: M1
Number of stock: 2


14.Product: MeetingTable
Color: Pink
Model: M2
Number of stock: 2


15.Product: MeetingTable
Color: Blue
Model: M3
Number of stock: 2


16.Product: MeetingTable
Color: White
Model: M4
Number of stock: 2


17.Product: MeetingTable
Color: Black
Model: M5
Number of stock: 2


18.Product: MeetingTable
Color: Brown
Model: M6
Number of stock: 2
```



```
19.Product: MeetingTable
Color: Black
Model: M7
Number of stock: 2


20.Product: MeetingTable
Color: Blue
Model: M8
Number of stock: 2


21.Product: MeetingTable
Color: Yellow
Model: M9
Number of stock: 2


22.Product: MeetingTable
Color: Green
Model: M10
Number of stock: 3


23.Product: OfficeCabinet
Color: None
Model: M1
Number of stock: 3


24.Product: OfficeCabinet
Color: None
Model: M2
Number of stock: 3


25.Product: OfficeCabinet
Color: None
Model: M3
Number of stock: 3


26.Product: OfficeCabinet
Color: None
Model: M4
Number of stock: 3


27.Product: OfficeCabinet
Color: None
Model: M5
Number of stock: 3


28.Product: OfficeCabinet
Color: None
Model: M6
Number of stock: 3
```



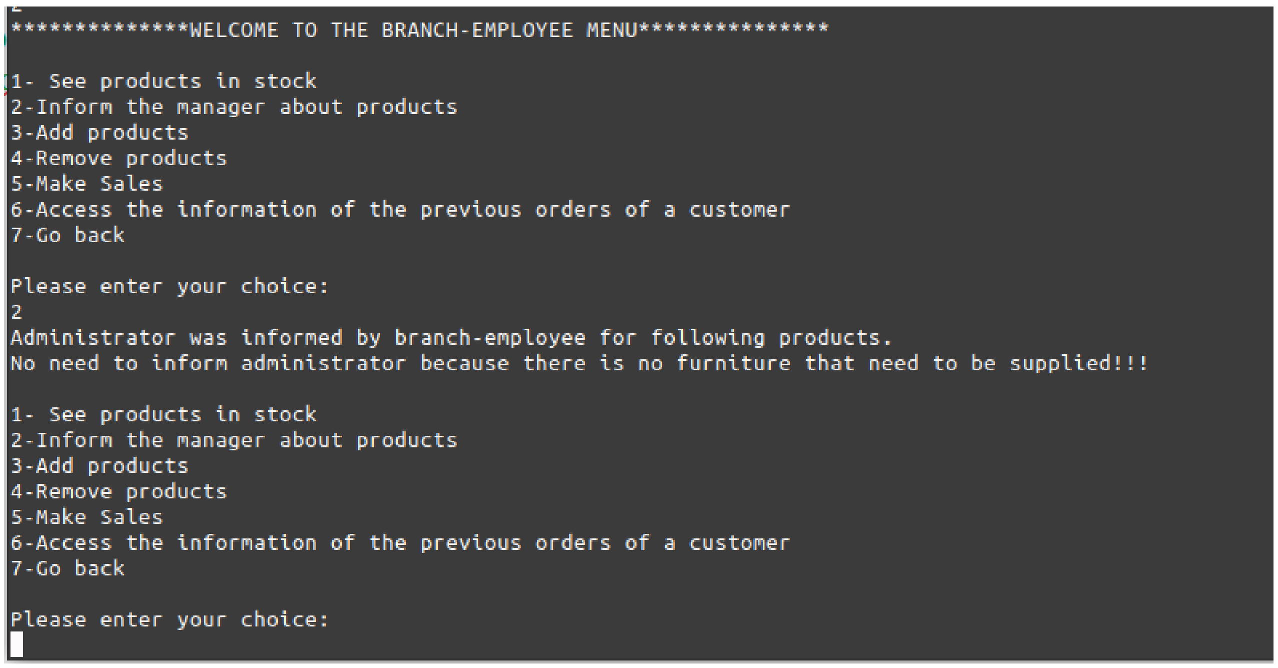
```
29.Product: OfficeCabinet
Color: None
Model: M7
Number of stock: 3


30.Product: OfficeCabinet
Color: None
Model: M8
Number of stock: 3


31.Product: OfficeCabinet
Color: None
Model: M9
Number of stock: 3


32.Product: OfficeCabinet
Color: None
Model: M10
Number of stock: 3


33.Product: OfficeCabinet
Color: None
Model: M11
Number of stock: 4


34.Product: OfficeCabinet
Color: None
Model: M12
Number of stock: 4


35.Product: Bookcase
Color: None
Model: M1
Number of stock: 4


36.Product: Bookcase
Color: None
Model: M2
Number of stock: 4


37.Product: Bookcase
Color: None
Model: M3
Number of stock: 4


38.Product: Bookcase
Color: None
Model: M4
Number of stock: 4
```



```
39.Product: Bookcase
Color: None
Model: M5
Number of stock: 4


40.Product: Bookcase
Color: None
Model: M6
Number of stock: 4


41.Product: Bookcase
Color: None
Model: M7
Number of stock: 4


42.Product: Bookcase
Color: None
Model: M8
Number of stock: 4


43.Product: Bookcase
Color: None
Model: M9
Number of stock: 4


44.Product: Bookcase
Color: None
Model: M10
Number of stock: 4


45.Product: Bookcase
Color: None
Model: M11
Number of stock: 4


46.Product: Bookcase
Color: None
Model: M12
Number of stock: 4



1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
```

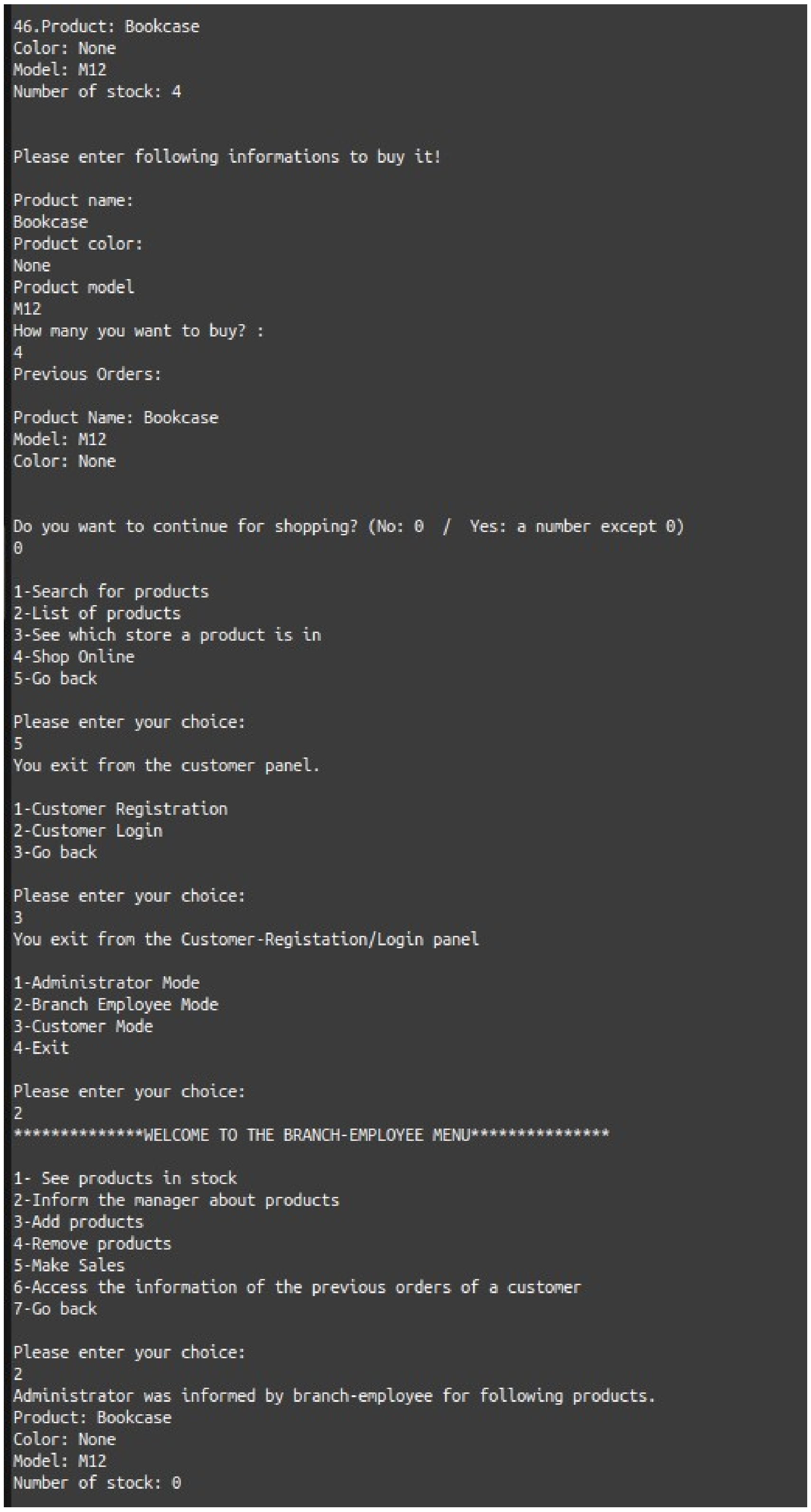
Test Case-14: Inform the manager(BranchEmployee): passed
- BranchEmployees inform the manager that the product should be purchased when any product is less than the requested amount.

*if one of the product stocks has not gone to zero after customer’s shopping then manager won’t be informed by branchemployee:



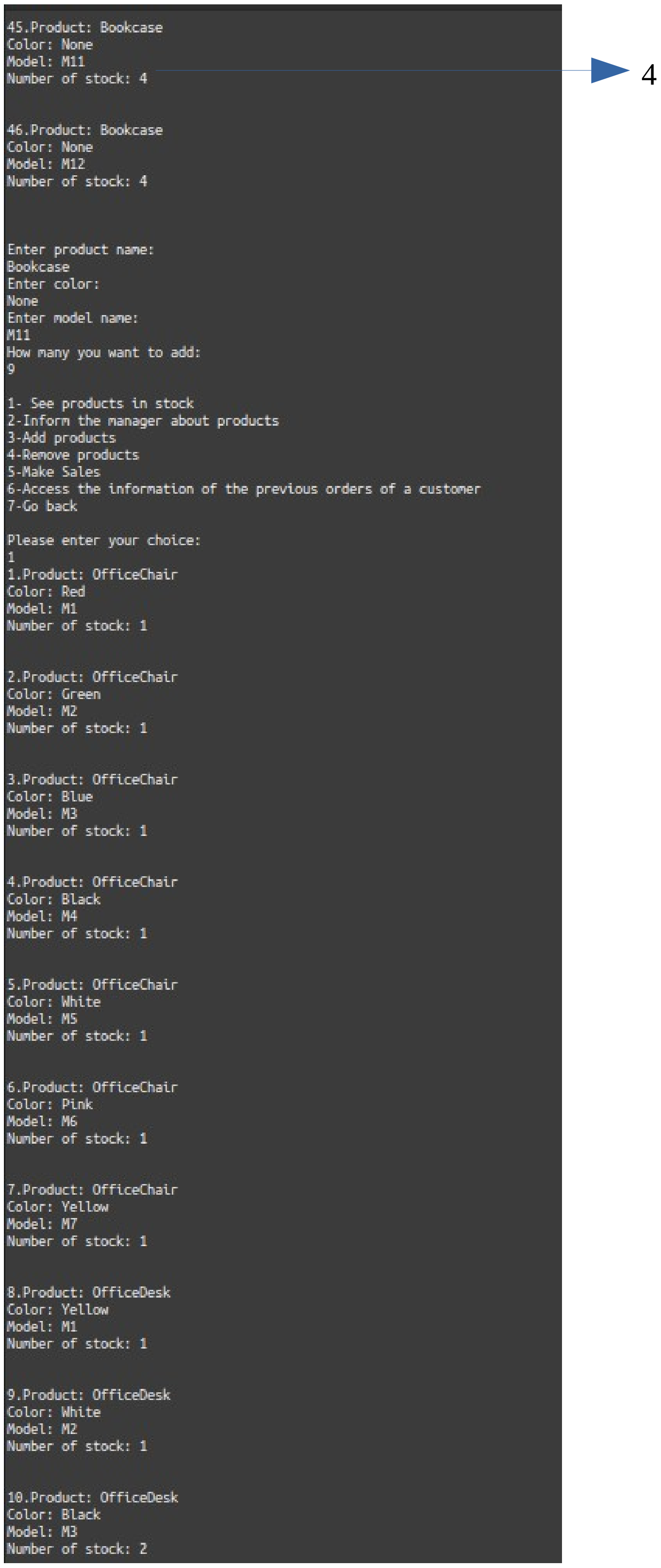
```
**************WELCOME TO THE BRANCH-EMPLOYEE MENU***************

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
2
Administrator was informed by branch-employee for following products.
No need to inform administrator because there is no furniture that need to be supplied!!!

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
```

*if one of the product stocks has gone to zero after customer's shopping then manager will be informed by branchemployee:

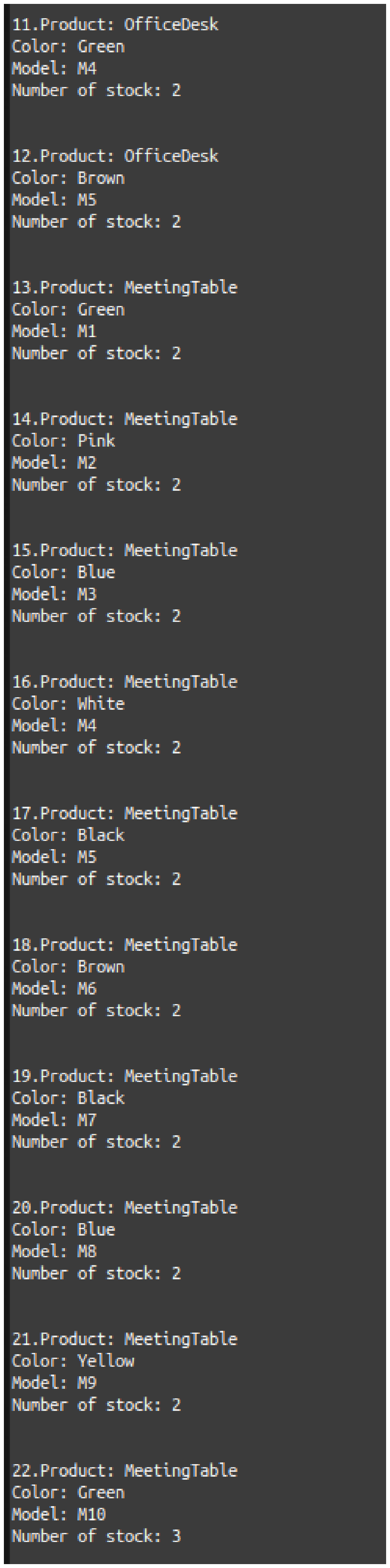


```
46.Product: Bookcase
Color: None
Model: M12
Number of stock: 4


Please enter following informations to buy it!

Product name:
Bookcase
Product color:
None
Product model
M12
How many you want to buy? :
4
Previous Orders:

Product Name: Bookcase
Model: M12
Color: None


Do you want to continue for shopping? (No: 0  /  Yes: a number except 0)
0

1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
5
You exit from the customer panel.

1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
3
You exit from the Customer-Registation/Login panel

1-Administrator Mode
2-Branch Employee Mode
3-Customer Mode
4-Exit

Please enter your choice:
2
**************WELCOME TO THE BRANCH-EMPLOYEE MENU***************

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
2
Administrator was informed by branch-employee for following products.
Product: Bookcase
Color: None
Model: M12
Number of stock: 0
```

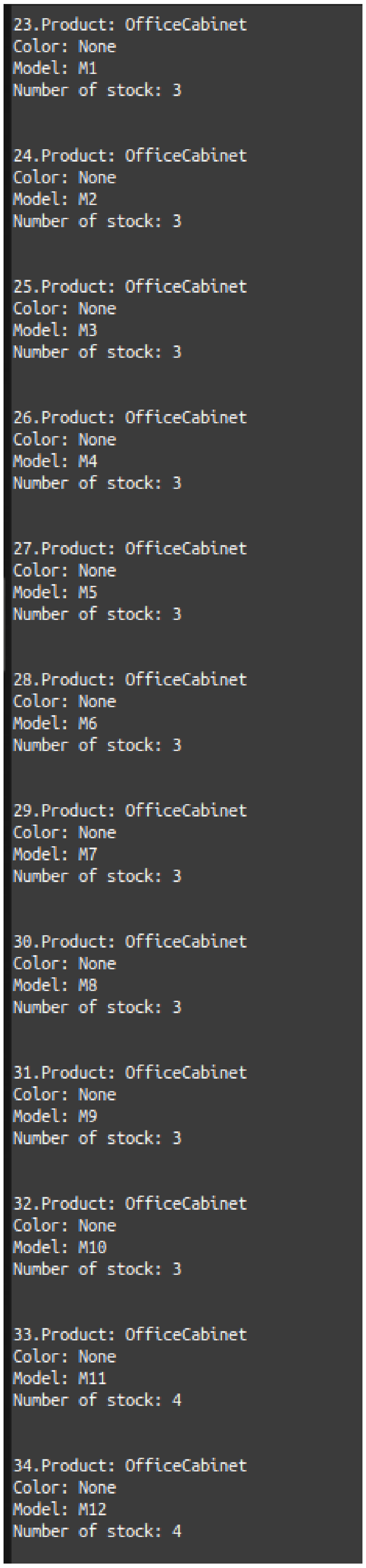
Test Case-15: Add products(BranchEmployee): passed
-I added Bookcase with none color,M11model and 4 pieces so there was 4 pieces at the beginning after my addition(I added 9 pieces in the example) it became 4+9 = 13.



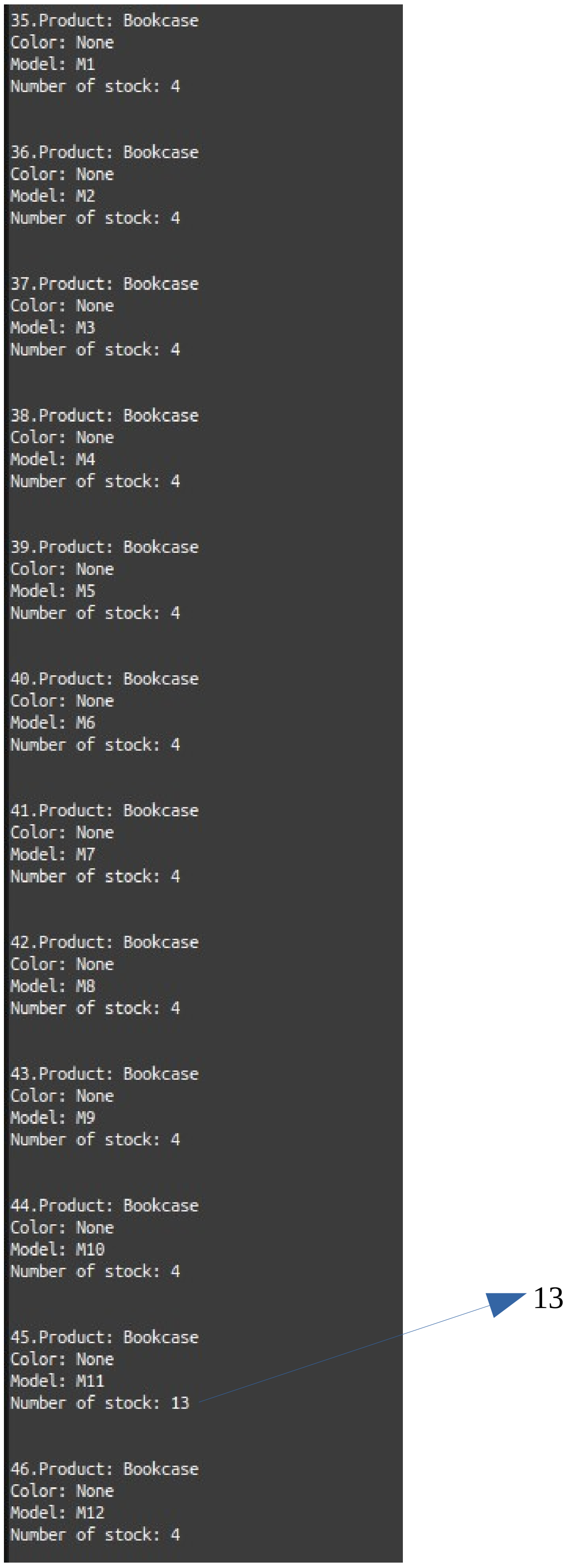
```
45.Product: Bookcase
Color: None
Model: M11
Number of stock: 4


46.Product: Bookcase
Color: None
Model: M12
Number of stock: 4



Enter product name:
Bookcase
Enter color:
None
Enter model name:
M11
How many you want to add:
9

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
1
1.Product: OfficeChair
Color: Red
Model: M1
Number of stock: 1


2.Product: OfficeChair
Color: Green
Model: M2
Number of stock: 1


3.Product: OfficeChair
Color: Blue
Model: M3
Number of stock: 1


4.Product: OfficeChair
Color: Black
Model: M4
Number of stock: 1


5.Product: OfficeChair
Color: White
Model: M5
Number of stock: 1


6.Product: OfficeChair
Color: Pink
Model: M6
Number of stock: 1


7.Product: OfficeChair
Color: Yellow
Model: M7
Number of stock: 1


8.Product: OfficeDesk
Color: Yellow
Model: M1
Number of stock: 1


9.Product: OfficeDesk
Color: White
Model: M2
Number of stock: 1


10.Product: OfficeDesk
Color: Black
Model: M3
Number of stock: 2
```



```
11.Product: OfficeDesk
Color: Green
Model: M4
Number of stock: 2


12.Product: OfficeDesk
Color: Brown
Model: M5
Number of stock: 2


13.Product: MeetingTable
Color: Green
Model: M1
Number of stock: 2


14.Product: MeetingTable
Color: Pink
Model: M2
Number of stock: 2


15.Product: MeetingTable
Color: Blue
Model: M3
Number of stock: 2


16.Product: MeetingTable
Color: White
Model: M4
Number of stock: 2


17.Product: MeetingTable
Color: Black
Model: M5
Number of stock: 2


18.Product: MeetingTable
Color: Brown
Model: M6
Number of stock: 2


19.Product: MeetingTable
Color: Black
Model: M7
Number of stock: 2


20.Product: MeetingTable
Color: Blue
Model: M8
Number of stock: 2


21.Product: MeetingTable
Color: Yellow
Model: M9
Number of stock: 2


22.Product: MeetingTable
Color: Green
Model: M10
Number of stock: 3
```

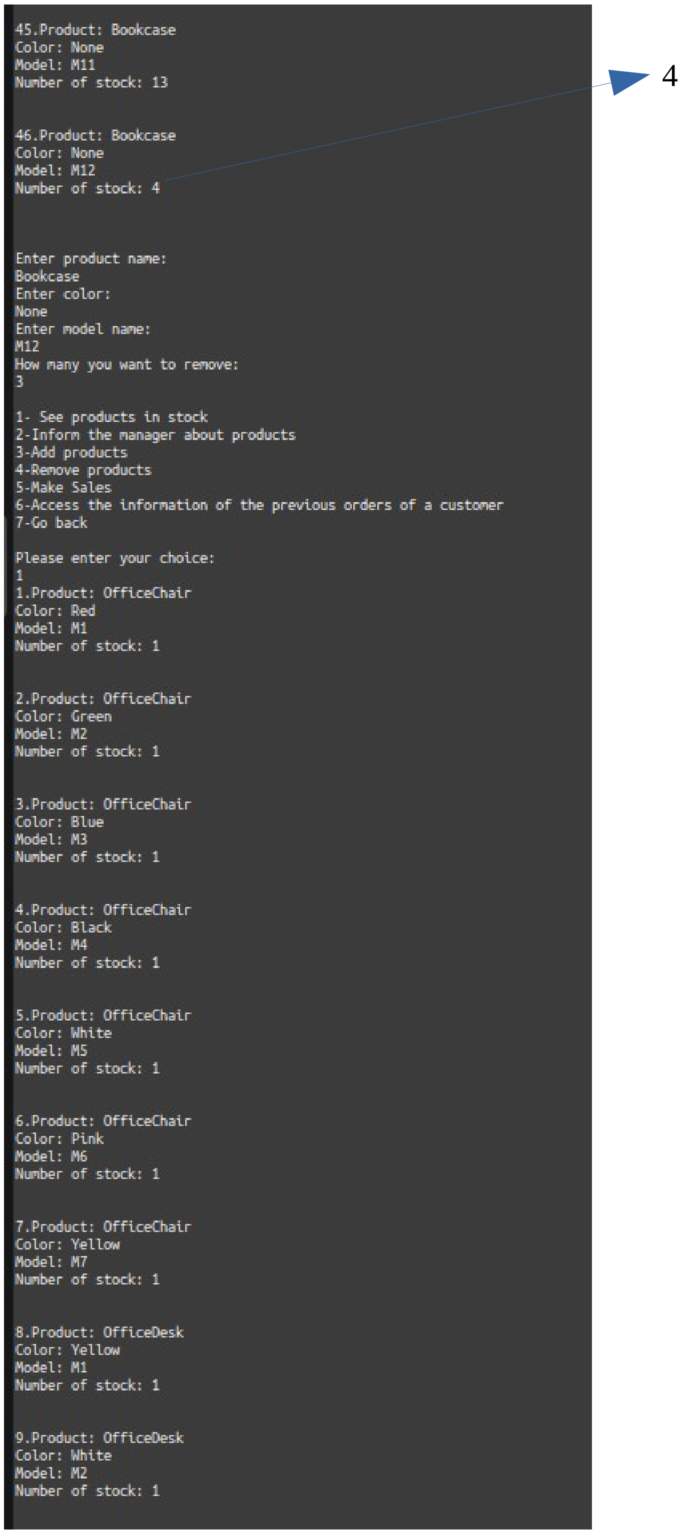


```
23.Product: OfficeCabinet
Color: None
Model: M1
Number of stock: 3


24.Product: OfficeCabinet
Color: None
Model: M2
Number of stock: 3


25.Product: OfficeCabinet
Color: None
Model: M3
Number of stock: 3


26.Product: OfficeCabinet
Color: None
Model: M4
Number of stock: 3


27.Product: OfficeCabinet
Color: None
Model: M5
Number of stock: 3


28.Product: OfficeCabinet
Color: None
Model: M6
Number of stock: 3


29.Product: OfficeCabinet
Color: None
Model: M7
Number of stock: 3


30.Product: OfficeCabinet
Color: None
Model: M8
Number of stock: 3


31.Product: OfficeCabinet
Color: None
Model: M9
Number of stock: 3


32.Product: OfficeCabinet
Color: None
Model: M10
Number of stock: 3


33.Product: OfficeCabinet
Color: None
Model: M11
Number of stock: 4


34.Product: OfficeCabinet
Color: None
Model: M12
Number of stock: 4
```

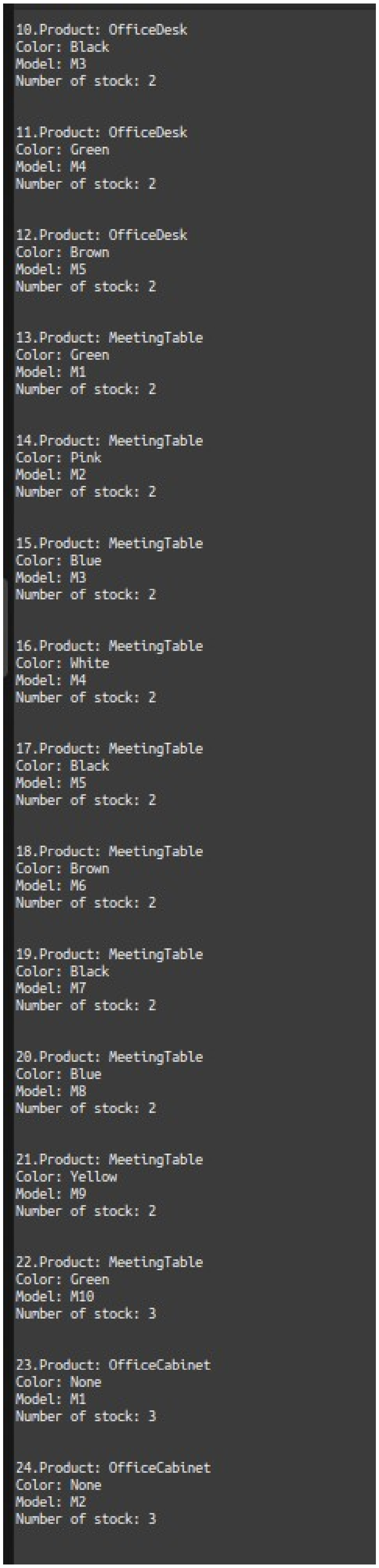


```
35.Product: Bookcase
Color: None
Model: M1
Number of stock: 4


36.Product: Bookcase
Color: None
Model: M2
Number of stock: 4


37.Product: Bookcase
Color: None
Model: M3
Number of stock: 4


38.Product: Bookcase
Color: None
Model: M4
Number of stock: 4


39.Product: Bookcase
Color: None
Model: M5
Number of stock: 4


40.Product: Bookcase
Color: None
Model: M6
Number of stock: 4


41.Product: Bookcase
Color: None
Model: M7
Number of stock: 4


42.Product: Bookcase
Color: None
Model: M8
Number of stock: 4


43.Product: Bookcase
Color: None
Model: M9
Number of stock: 4


44.Product: Bookcase
Color: None
Model: M10
Number of stock: 4


45.Product: Bookcase
Color: None
Model: M11
Number of stock: 13


46.Product: Bookcase
Color: None
Model: M12
Number of stock: 4
```

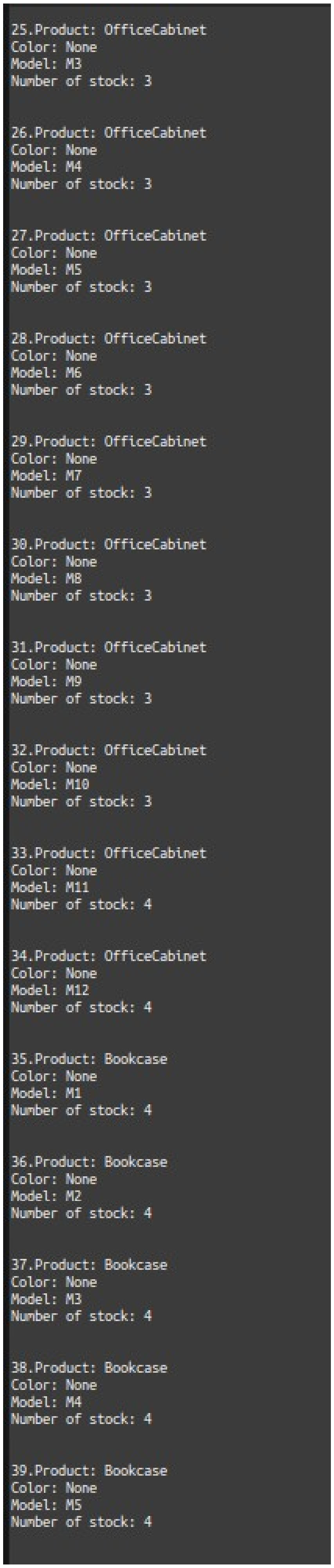
Test Case-15: Add products(BranchEmployee): passed
-I added Bookcase with none color,M12model and 4 pieces so there was 4 pieces at the beginning after my addition(I removed 3 pieces in the example) it became 4 - 3 = 1.



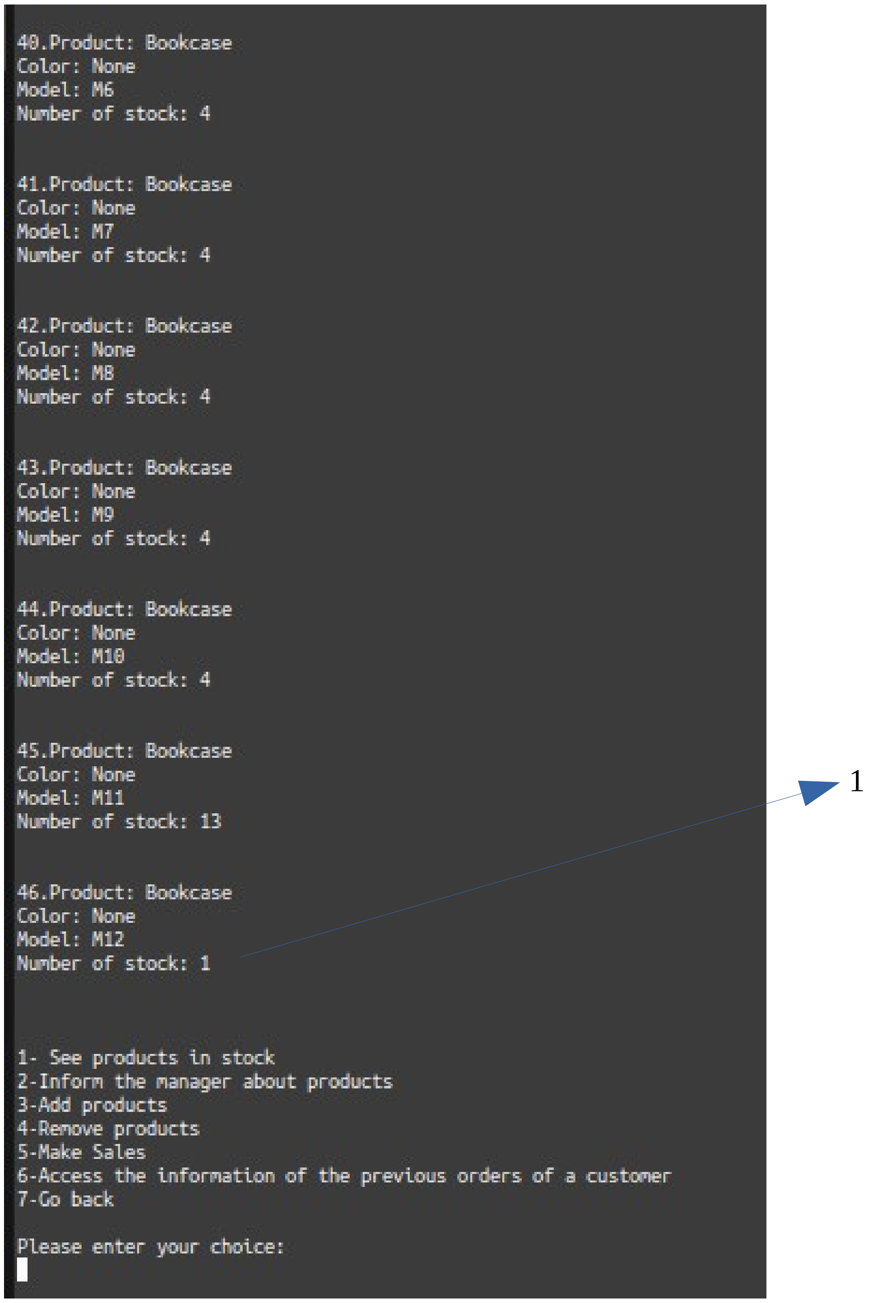
```
45.Product: Bookcase
Color: None
Model: M11
Number of stock: 13


46.Product: Bookcase
Color: None
Model: M12
Number of stock: 4



Enter product name:
Bookcase
Enter color:
None
Enter model name:
M12
How many you want to remove:
3

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
1
1.Product: OfficeChair
Color: Red
Model: M1
Number of stock: 1


2.Product: OfficeChair
Color: Green
Model: M2
Number of stock: 1


3.Product: OfficeChair
Color: Blue
Model: M3
Number of stock: 1


4.Product: OfficeChair
Color: Black
Model: M4
Number of stock: 1


5.Product: OfficeChair
Color: White
Model: M5
Number of stock: 1


6.Product: OfficeChair
Color: Pink
Model: M6
Number of stock: 1


7.Product: OfficeChair
Color: Yellow
Model: M7
Number of stock: 1


8.Product: OfficeDesk
Color: Yellow
Model: M1
Number of stock: 1


9.Product: OfficeDesk
Color: White
Model: M2
Number of stock: 1
```



```
10.Product: OfficeDesk
Color: Black
Model: M3
Number of stock: 2


11.Product: OfficeDesk
Color: Green
Model: M4
Number of stock: 2


12.Product: OfficeDesk
Color: Brown
Model: M5
Number of stock: 2


13.Product: MeetingTable
Color: Green
Model: M1
Number of stock: 2


14.Product: MeetingTable
Color: Pink
Model: M2
Number of stock: 2


15.Product: MeetingTable
Color: Blue
Model: M3
Number of stock: 2


16.Product: MeetingTable
Color: White
Model: M4
Number of stock: 2


17.Product: MeetingTable
Color: Black
Model: M5
Number of stock: 2


18.Product: MeetingTable
Color: Brown
Model: M6
Number of stock: 2


19.Product: MeetingTable
Color: Black
Model: M7
Number of stock: 2


20.Product: MeetingTable
Color: Blue
Model: M8
Number of stock: 2


21.Product: MeetingTable
Color: Yellow
Model: M9
Number of stock: 2


22.Product: MeetingTable
Color: Green
Model: M10
Number of stock: 3


23.Product: OfficeCabinet
Color: None
Model: M1
Number of stock: 3


24.Product: OfficeCabinet
Color: None
Model: M2
Number of stock: 3
```

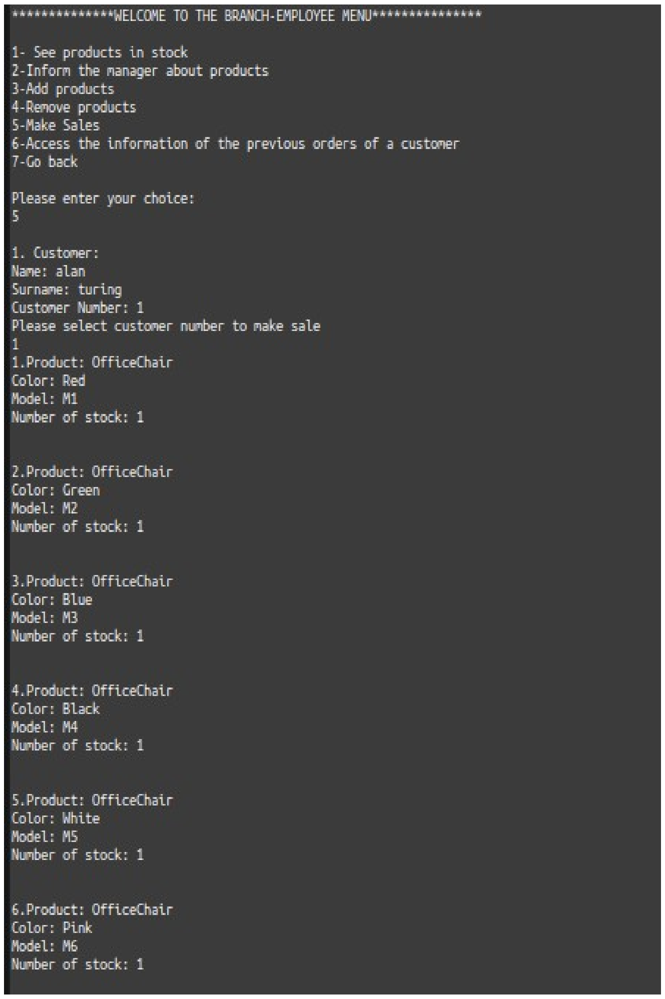


```
25.Product: OfficeCabinet
Color: None
Model: M3
Number of stock: 3


26.Product: OfficeCabinet
Color: None
Model: M4
Number of stock: 3


27.Product: OfficeCabinet
Color: None
Model: M5
Number of stock: 3


28.Product: OfficeCabinet
Color: None
Model: M6
Number of stock: 3


29.Product: OfficeCabinet
Color: None
Model: M7
Number of stock: 3


30.Product: OfficeCabinet
Color: None
Model: M8
Number of stock: 3


31.Product: OfficeCabinet
Color: None
Model: M9
Number of stock: 3


32.Product: OfficeCabinet
Color: None
Model: M10
Number of stock: 3


33.Product: OfficeCabinet
Color: None
Model: M11
Number of stock: 4


34.Product: OfficeCabinet
Color: None
Model: M12
Number of stock: 4


35.Product: Bookcase
Color: None
Model: M1
Number of stock: 4


36.Product: Bookcase
Color: None
Model: M2
Number of stock: 4


37.Product: Bookcase
Color: None
Model: M3
Number of stock: 4


38.Product: Bookcase
Color: None
Model: M4
Number of stock: 4


39.Product: Bookcase
Color: None
Model: M5
Number of stock: 4
```



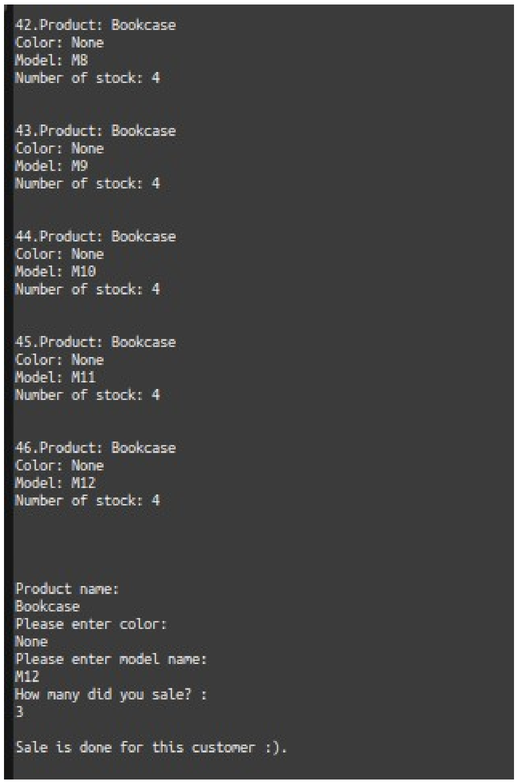
```
40.Product: Bookcase
Color: None
Model: M6
Number of stock: 4


41.Product: Bookcase
Color: None
Model: M7
Number of stock: 4


42.Product: Bookcase
Color: None
Model: M8
Number of stock: 4


43.Product: Bookcase
Color: None
Model: M9
Number of stock: 4


44.Product: Bookcase
Color: None
Model: M10
Number of stock: 4


45.Product: Bookcase
Color: None
Model: M11
Number of stock: 13


46.Product: Bookcase
Color: None
Model: M12
Number of stock: 1



1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
```

Test Case-16: Make Sales (BranchEmployee): passed



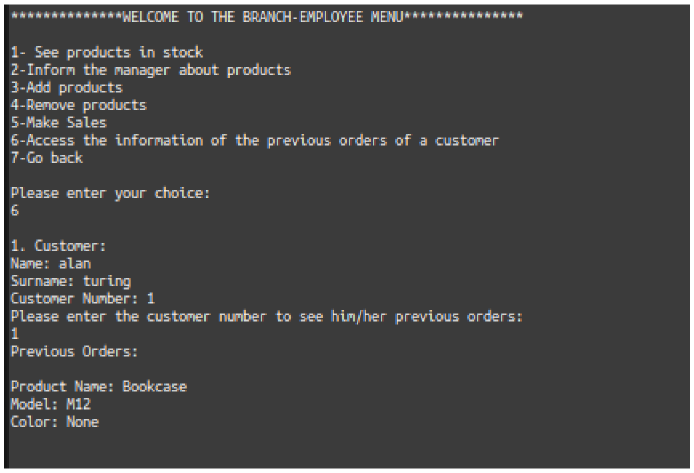
```
**************WELCOME TO THE BRANCH-EMPLOYEE MENU***************

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
5

1. Customer:
Name: alan
Surname: turing
Customer Number: 1
Please select customer number to make sale
1
1.Product: OfficeChair
Color: Red
Model: M1
Number of stock: 1


2.Product: OfficeChair
Color: Green
Model: M2
Number of stock: 1


3.Product: OfficeChair
Color: Blue
Model: M3
Number of stock: 1


4.Product: OfficeChair
Color: Black
Model: M4
Number of stock: 1


5.Product: OfficeChair
Color: White
Model: M5
Number of stock: 1


6.Product: OfficeChair
Color: Pink
Model: M6
Number of stock: 1
```

….



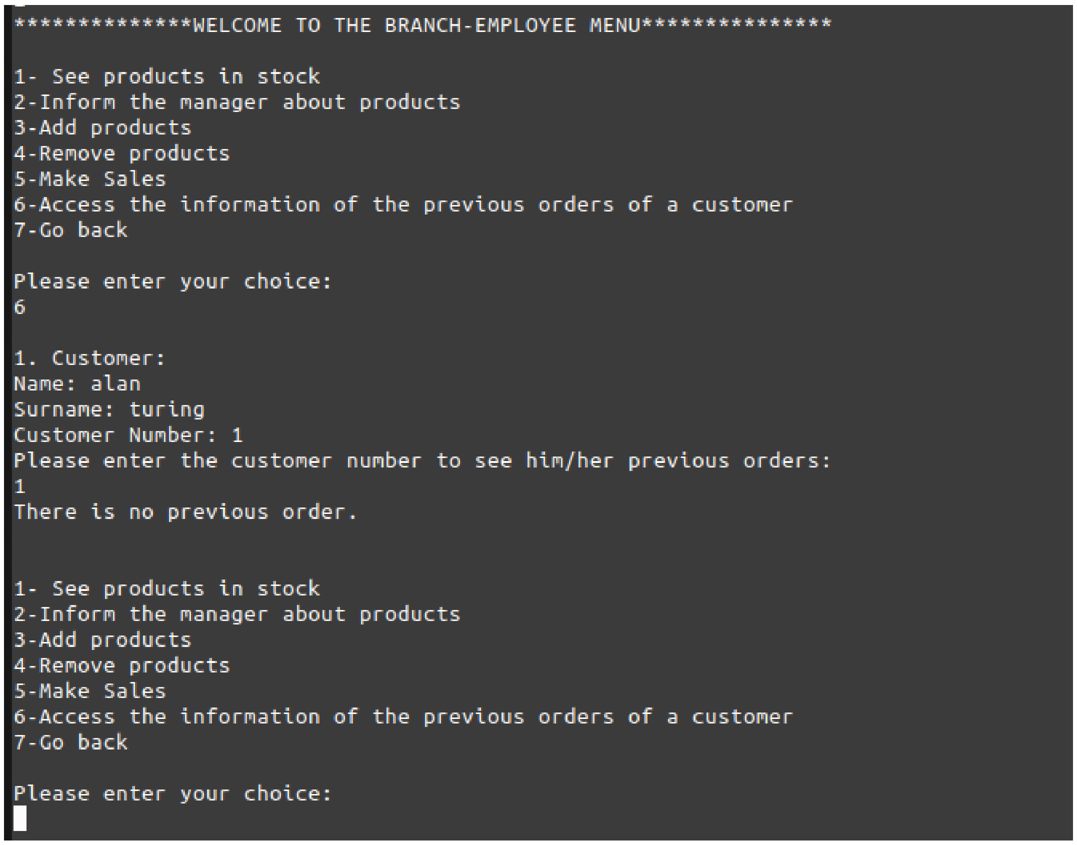
```
42.Product: Bookcase
Color: None
Model: M8
Number of stock: 4


43.Product: Bookcase
Color: None
Model: M9
Number of stock: 4


44.Product: Bookcase
Color: None
Model: M10
Number of stock: 4


45.Product: Bookcase
Color: None
Model: M11
Number of stock: 4


46.Product: Bookcase
Color: None
Model: M12
Number of stock: 4



Product name:
Bookcase
Please enter color:
None
Please enter model name:
M12
How many did you sale? :
3

Sale is done for this customer :).
```

Test Case-16: Access the information of the previous orders of a customer(if he/she bought at least one product) (BranchEmployee): passed



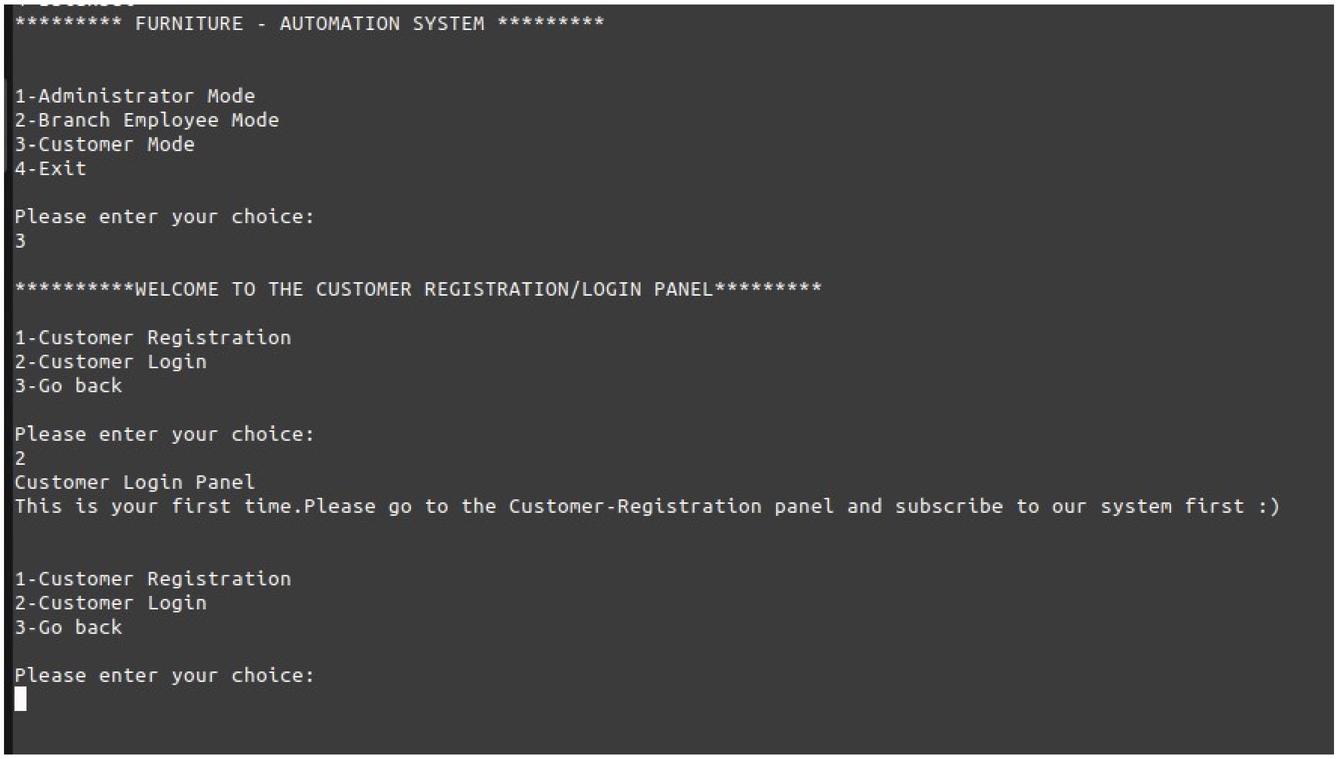
```
**************WELCOME TO THE BRANCH-EMPLOYEE MENU***************

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
6

1. Customer:
Name: alan
Surname: turing
Customer Number: 1
Please enter the customer number to see him/her previous orders:
1
Previous Orders:

Product Name: Bookcase
Model: M12
Color: None
```

Test Case-17: Access the information of the previous orders of a customer(he/she didn't buy product yet) (BranchEmployee): passed



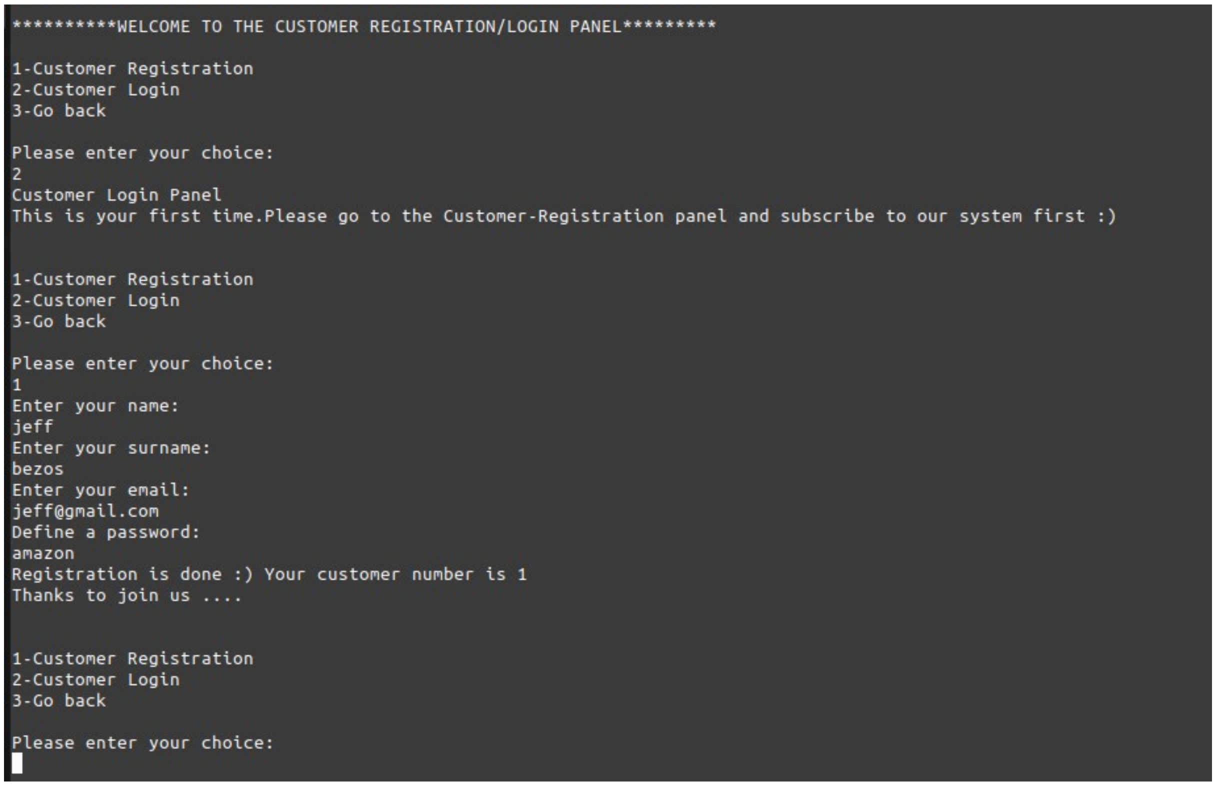
```
**************WELCOME TO THE BRANCH-EMPLOYEE MENU***************

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
6

1. Customer:
Name: alan
Surname: turing
Customer Number: 1
Please enter the customer number to see him/her previous orders:
1
There is no previous order.


1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
```

Test Case-18: If customer enters the login before registration,system will inform the customer about subscription: passed



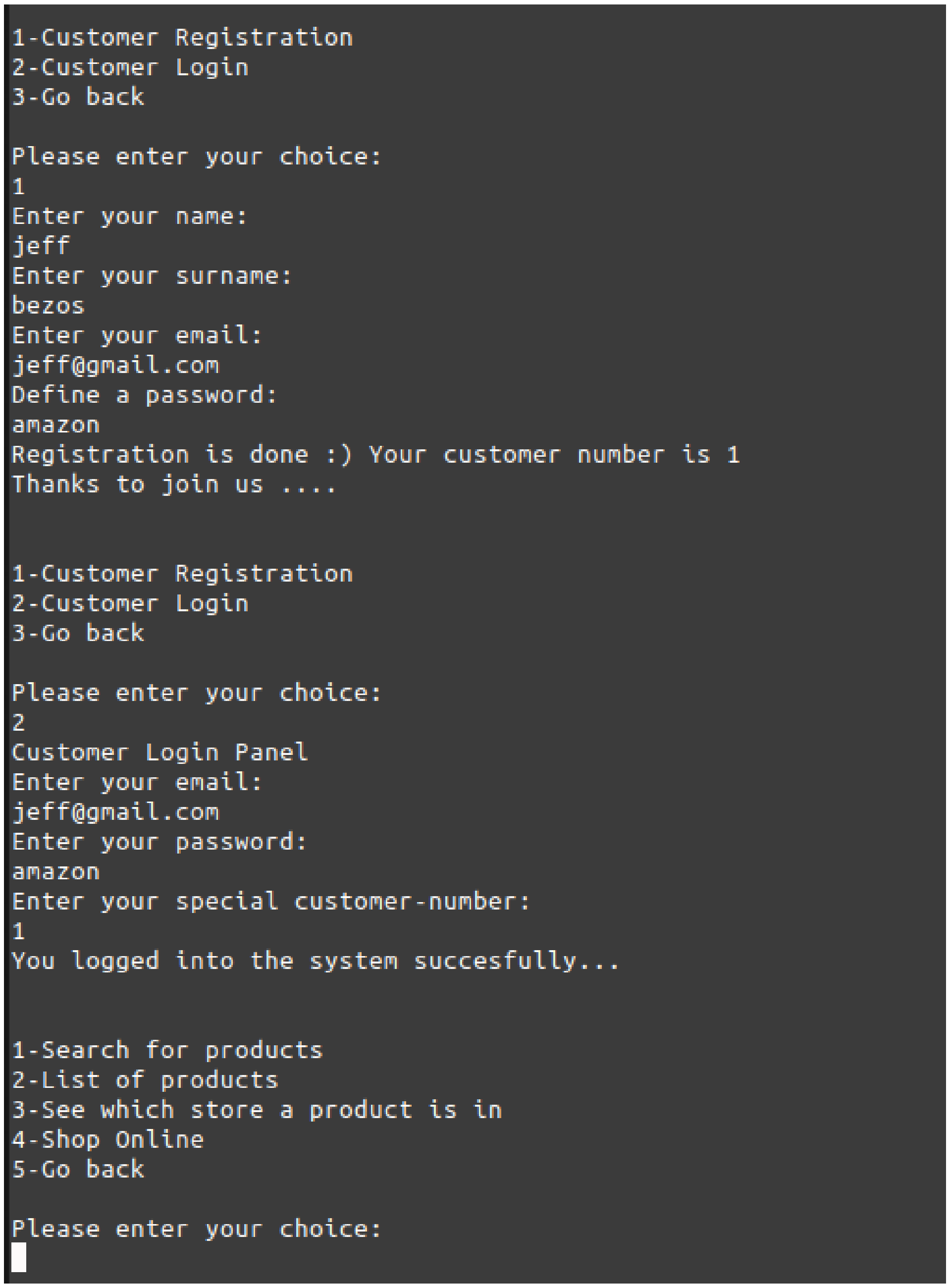
```
********* FURNITURE - AUTOMATION SYSTEM *********


1-Administrator Mode
2-Branch Employee Mode
3-Customer Mode
4-Exit

Please enter your choice:
3

**********WELCOME TO THE CUSTOMER REGISTRATION/LOGIN PANEL*********

1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
2
Customer Login Panel
This is your first time.Please go to the Customer-Registration panel and subscribe to our system first :)


1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
```

Test Case-19: If customer enters the registration part before login page,system will want information of customer to subscribe him/her and gives a special number to him/her. This number is special for each of customers: passed



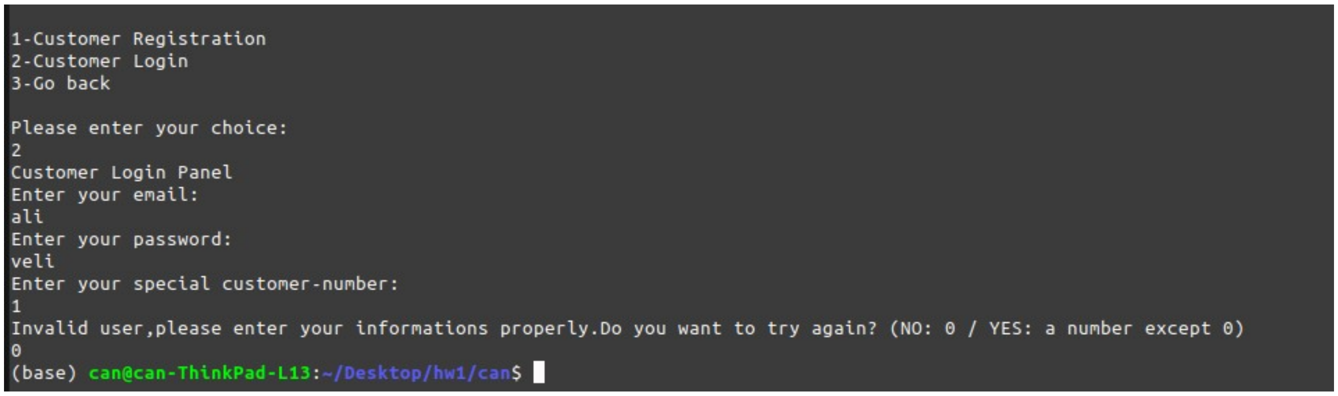
```
**********WELCOME TO THE CUSTOMER REGISTRATION/LOGIN PANEL*********

1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
2
Customer Login Panel
This is your first time.Please go to the Customer-Registration panel and subscribe to our system first :)


1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
1
Enter your name:
jeff
Enter your surname:
bezos
Enter your email:
jeff@gmail.com
Define a password:
amazon
Registration is done :) Your customer number is 1
Thanks to join us ....


1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
```

Test Case-20: If customer enters the email,password and customer-number information correctly then system allows him/her to see customer panel: passed



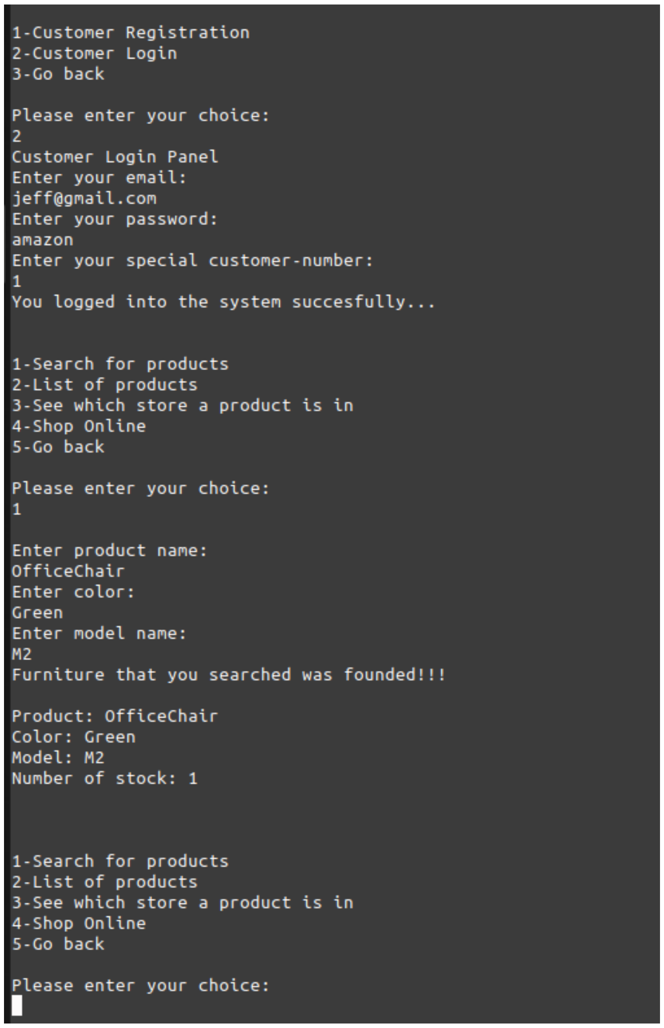
```
1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
1
Enter your name:
jeff
Enter your surname:
bezos
Enter your email:
jeff@gmail.com
Define a password:
amazon
Registration is done :) Your customer number is 1
Thanks to join us ....


1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
2
Customer Login Panel
Enter your email:
jeff@gmail.com
Enter your password:
amazon
Enter your special customer-number:
1
You logged into the system succesfully...


1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
```

Test Case-21: If customer enters the email,password and customer-number information as wrong then system doesn't allow him/her to see customer panel: → it gives error message: passed



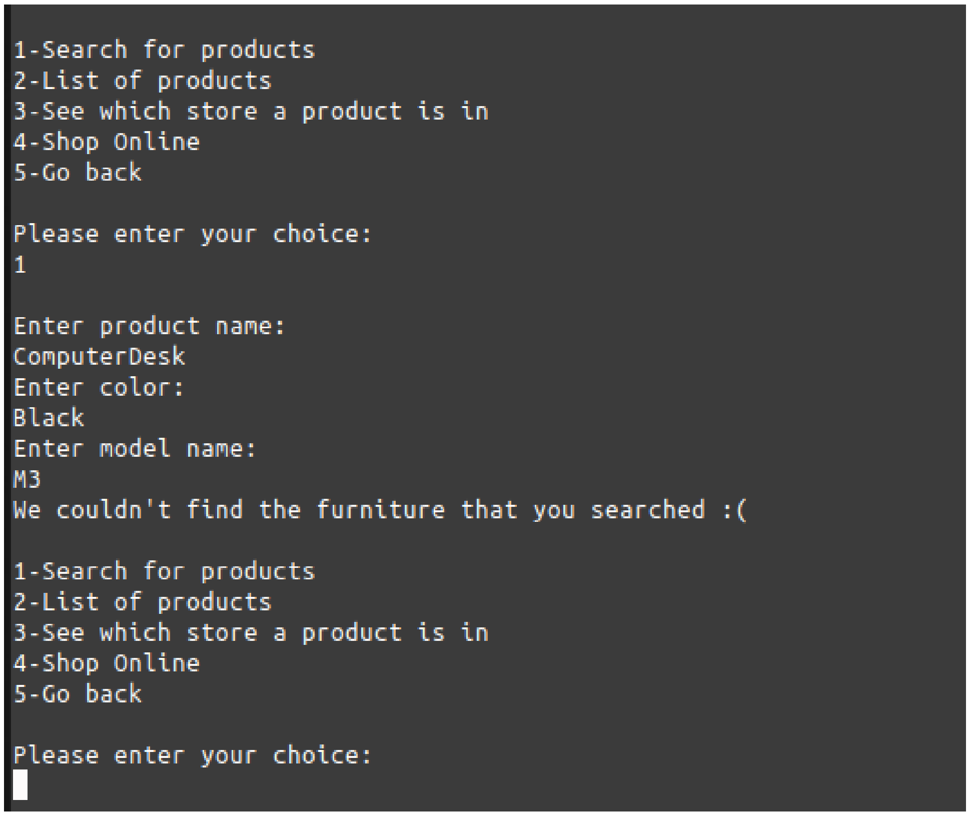
```
1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
2
Customer Login Panel
Enter your email:
ali
Enter your password:
veli
Enter your special customer-number:
1
Invalid user,please enter your informations properly.Do you want to try again? (NO: 0 / YES: a number except 0)
0
(base) can@can-ThinkPad-L13:~/Desktop/hw1/can$ 
```

Test Case-22: Search for products(customer): passed
*if it was founded,prints the information of product



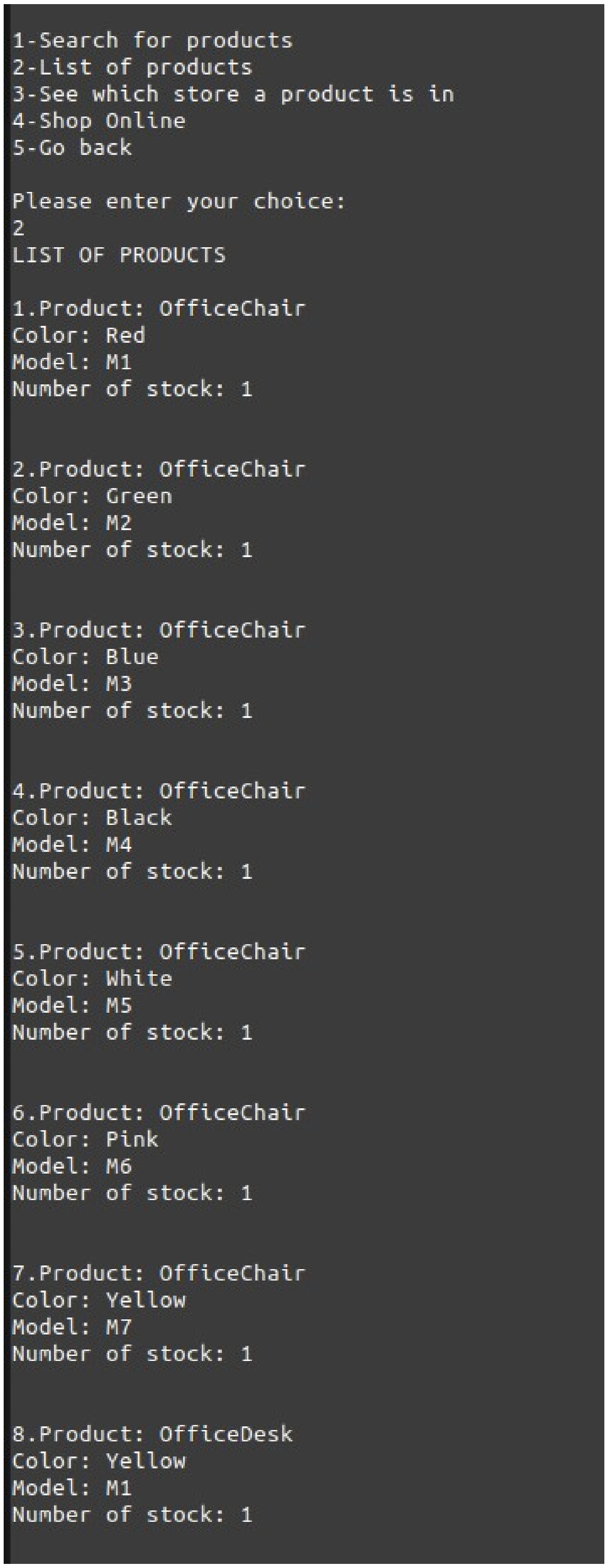
```
1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
2
Customer Login Panel
Enter your email:
jeff@gmail.com
Enter your password:
amazon
Enter your special customer-number:
1
You logged into the system succesfully...


1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
1

Enter product name:
OfficeChair
Enter color:
Green
Enter model name:
M2
Furniture that you searched was founded!!!

Product: OfficeChair
Color: Green
Model: M2
Number of stock: 1



1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
```

*if it was not founded: prints the error message

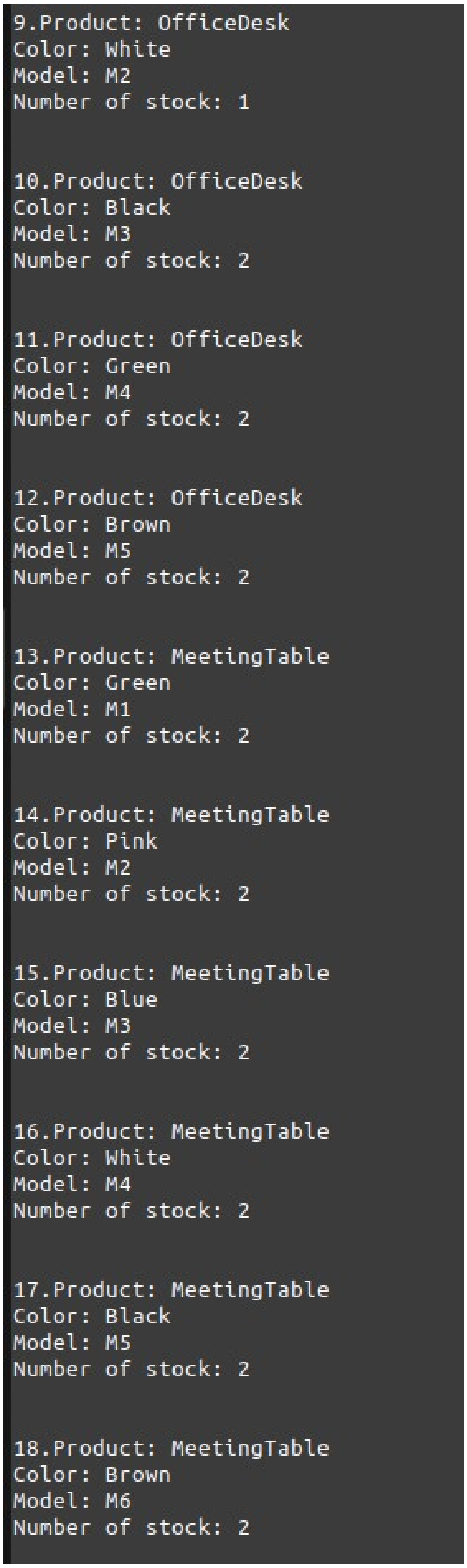


```
1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
1

Enter product name:
ComputerDesk
Enter color:
Black
Enter model name:
M3
We couldn't find the furniture that you searched :(

1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
```

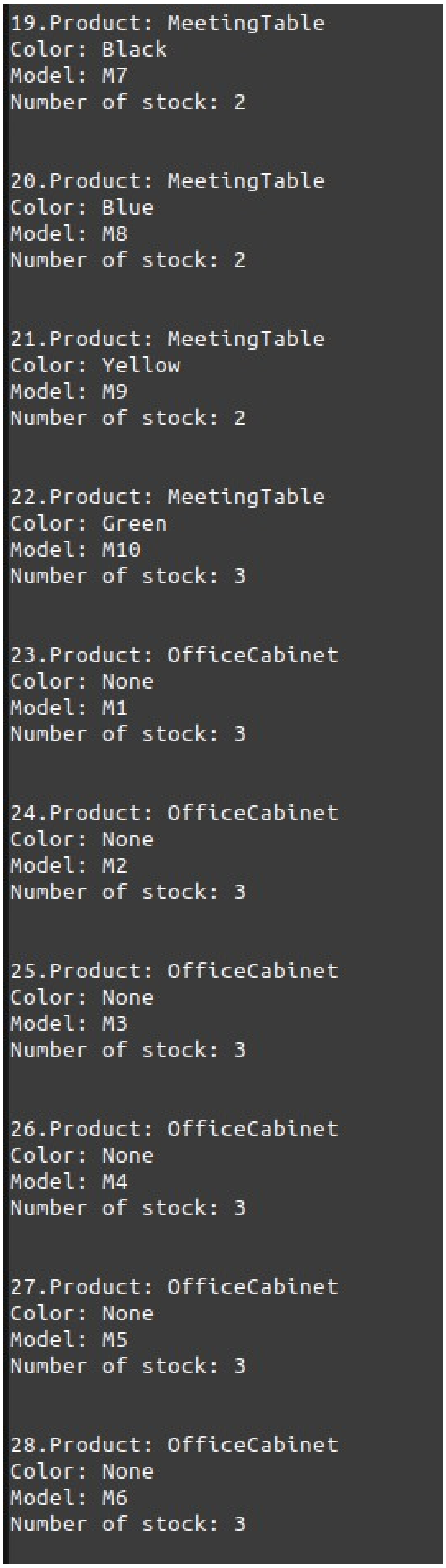
Test Case-23: List of products(customer): passed
*It shows the all products in the system.



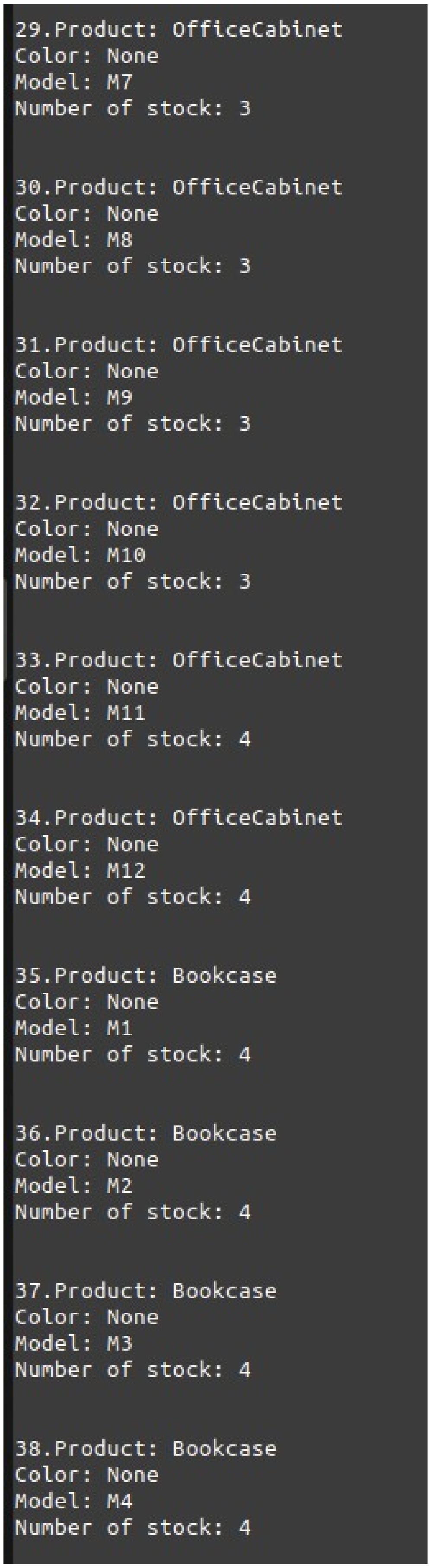
```
1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
2
LIST OF PRODUCTS

1.Product: OfficeChair
Color: Red
Model: M1
Number of stock: 1


2.Product: OfficeChair
Color: Green
Model: M2
Number of stock: 1


3.Product: OfficeChair
Color: Blue
Model: M3
Number of stock: 1


4.Product: OfficeChair
Color: Black
Model: M4
Number of stock: 1


5.Product: OfficeChair
Color: White
Model: M5
Number of stock: 1


6.Product: OfficeChair
Color: Pink
Model: M6
Number of stock: 1


7.Product: OfficeChair
Color: Yellow
Model: M7
Number of stock: 1


8.Product: OfficeDesk
Color: Yellow
Model: M1
Number of stock: 1
```



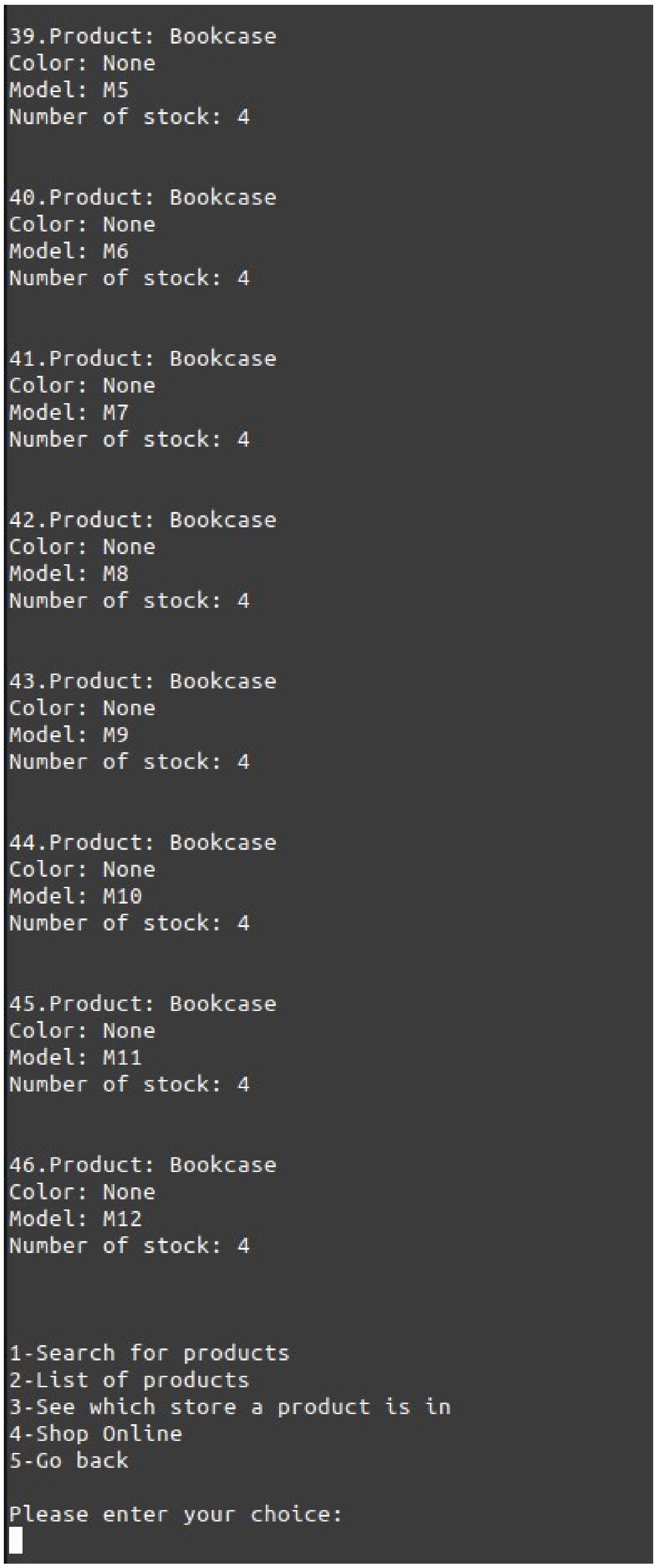
```
9.Product: OfficeDesk
Color: White
Model: M2
Number of stock: 1


10.Product: OfficeDesk
Color: Black
Model: M3
Number of stock: 2


11.Product: OfficeDesk
Color: Green
Model: M4
Number of stock: 2


12.Product: OfficeDesk
Color: Brown
Model: M5
Number of stock: 2


13.Product: MeetingTable
Color: Green
Model: M1
Number of stock: 2


14.Product: MeetingTable
Color: Pink
Model: M2
Number of stock: 2


15.Product: MeetingTable
Color: Blue
Model: M3
Number of stock: 2


16.Product: MeetingTable
Color: White
Model: M4
Number of stock: 2


17.Product: MeetingTable
Color: Black
Model: M5
Number of stock: 2


18.Product: MeetingTable
Color: Brown
Model: M6
Number of stock: 2
```



```
19.Product: MeetingTable
Color: Black
Model: M7
Number of stock: 2


20.Product: MeetingTable
Color: Blue
Model: M8
Number of stock: 2


21.Product: MeetingTable
Color: Yellow
Model: M9
Number of stock: 2


22.Product: MeetingTable
Color: Green
Model: M10
Number of stock: 3


23.Product: OfficeCabinet
Color: None
Model: M1
Number of stock: 3


24.Product: OfficeCabinet
Color: None
Model: M2
Number of stock: 3


25.Product: OfficeCabinet
Color: None
Model: M3
Number of stock: 3


26.Product: OfficeCabinet
Color: None
Model: M4
Number of stock: 3


27.Product: OfficeCabinet
Color: None
Model: M5
Number of stock: 3


28.Product: OfficeCabinet
Color: None
Model: M6
Number of stock: 3
```

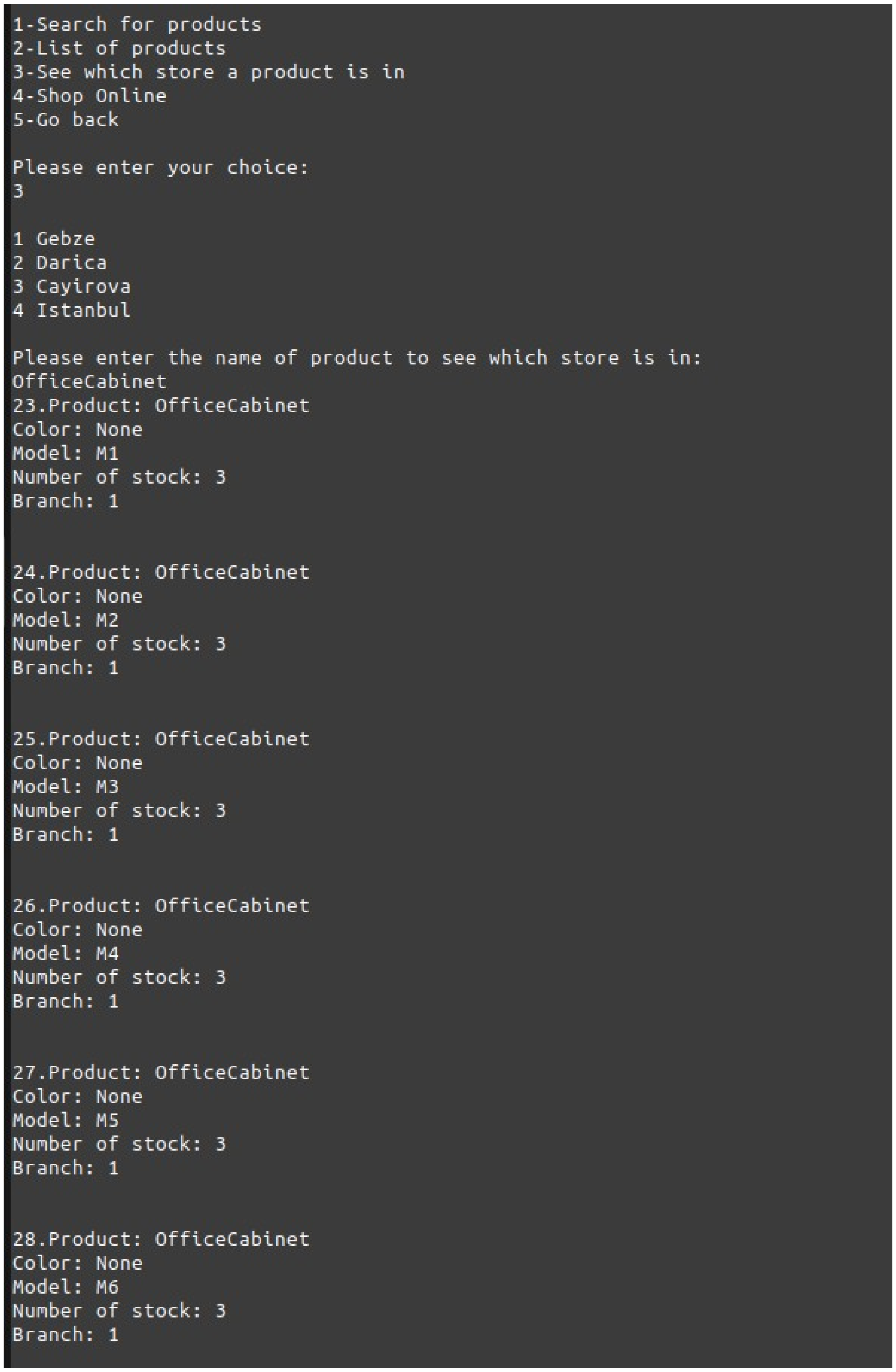


```
29.Product: OfficeCabinet
Color: None
Model: M7
Number of stock: 3


30.Product: OfficeCabinet
Color: None
Model: M8
Number of stock: 3


31.Product: OfficeCabinet
Color: None
Model: M9
Number of stock: 3


32.Product: OfficeCabinet
Color: None
Model: M10
Number of stock: 3


33.Product: OfficeCabinet
Color: None
Model: M11
Number of stock: 4


34.Product: OfficeCabinet
Color: None
Model: M12
Number of stock: 4


35.Product: Bookcase
Color: None
Model: M1
Number of stock: 4


36.Product: Bookcase
Color: None
Model: M2
Number of stock: 4


37.Product: Bookcase
Color: None
Model: M3
Number of stock: 4


38.Product: Bookcase
Color: None
Model: M4
Number of stock: 4
```

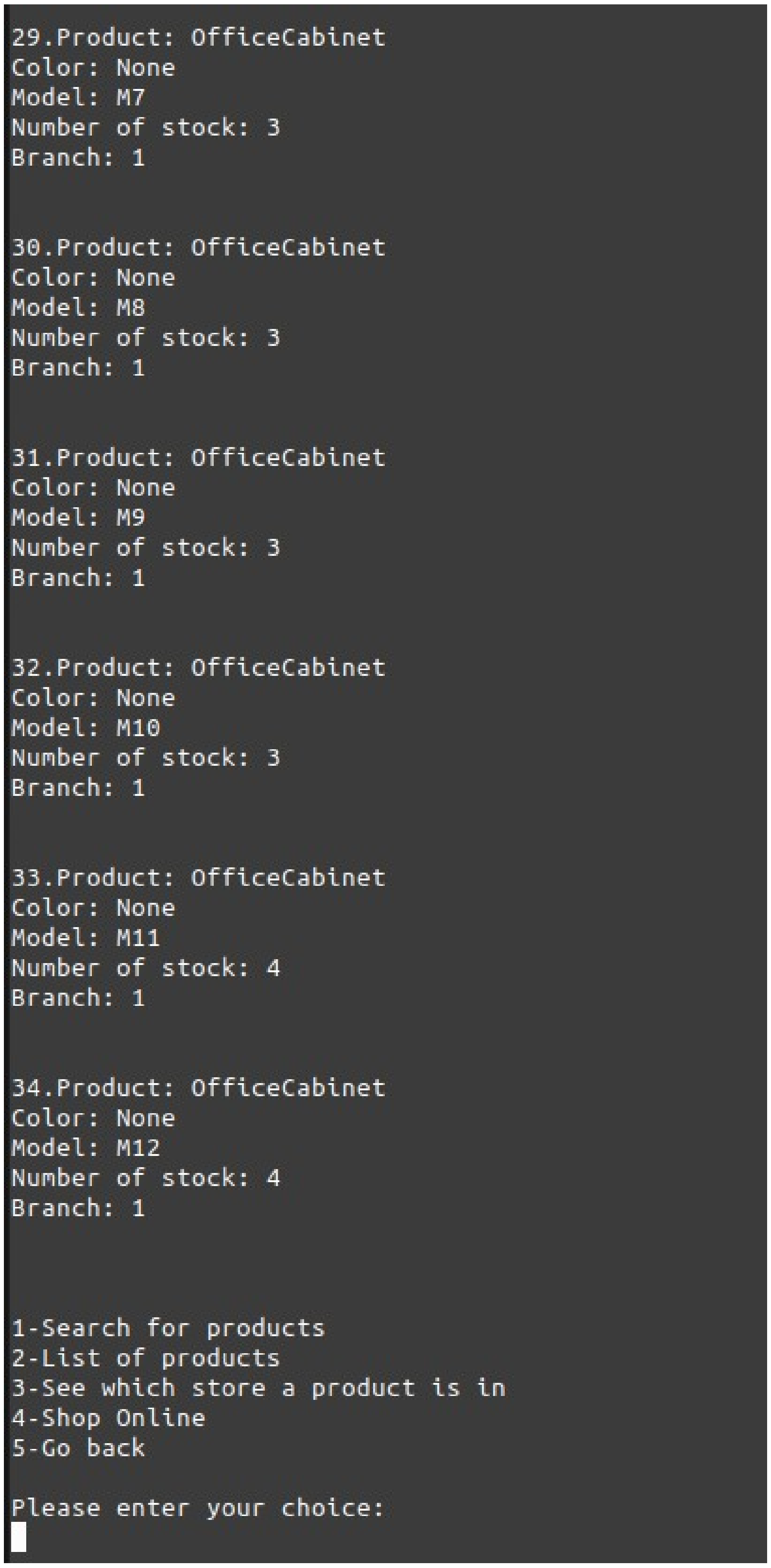


```
39.Product: Bookcase
Color: None
Model: M5
Number of stock: 4


40.Product: Bookcase
Color: None
Model: M6
Number of stock: 4


41.Product: Bookcase
Color: None
Model: M7
Number of stock: 4


42.Product: Bookcase
Color: None
Model: M8
Number of stock: 4


43.Product: Bookcase
Color: None
Model: M9
Number of stock: 4


44.Product: Bookcase
Color: None
Model: M10
Number of stock: 4


45.Product: Bookcase
Color: None
Model: M11
Number of stock: 4


46.Product: Bookcase
Color: None
Model: M12
Number of stock: 4



1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
```

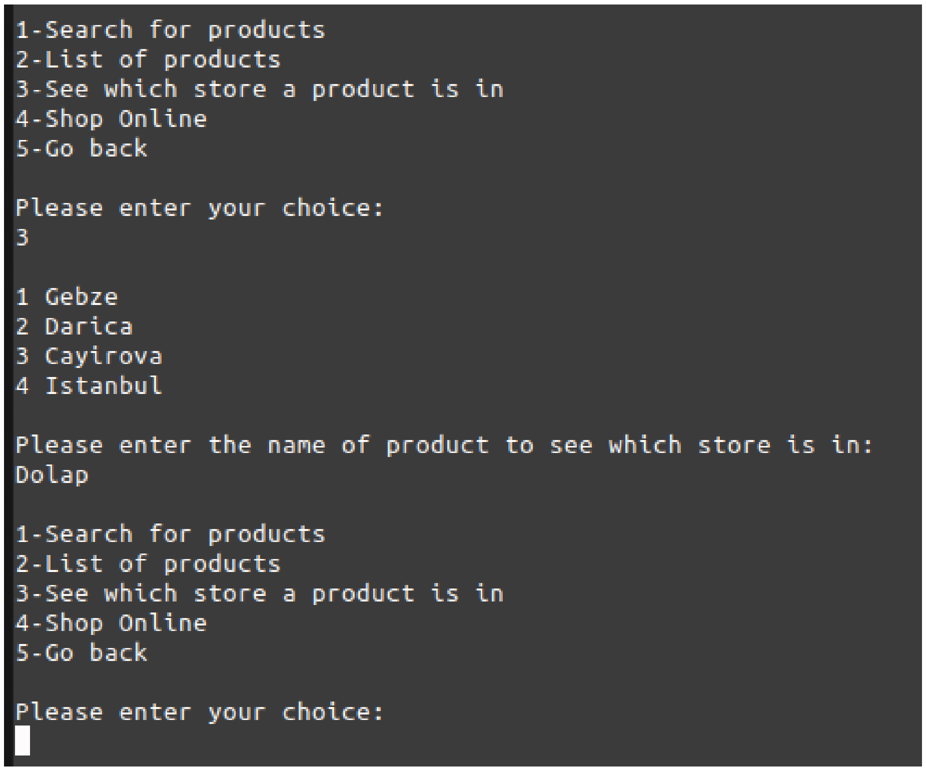
Test Case-24: See which store a product is in.(customer): passed
*if it's founded in the store.



```
1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
3

1 Gebze
2 Darica
3 Cayirova
4 Istanbul

Please enter the name of product to see which store is in:
OfficeCabinet
23.Product: OfficeCabinet
Color: None
Model: M1
Number of stock: 3
Branch: 1


24.Product: OfficeCabinet
Color: None
Model: M2
Number of stock: 3
Branch: 1


25.Product: OfficeCabinet
Color: None
Model: M3
Number of stock: 3
Branch: 1


26.Product: OfficeCabinet
Color: None
Model: M4
Number of stock: 3
Branch: 1


27.Product: OfficeCabinet
Color: None
Model: M5
Number of stock: 3
Branch: 1


28.Product: OfficeCabinet
Color: None
Model: M6
Number of stock: 3
Branch: 1
```

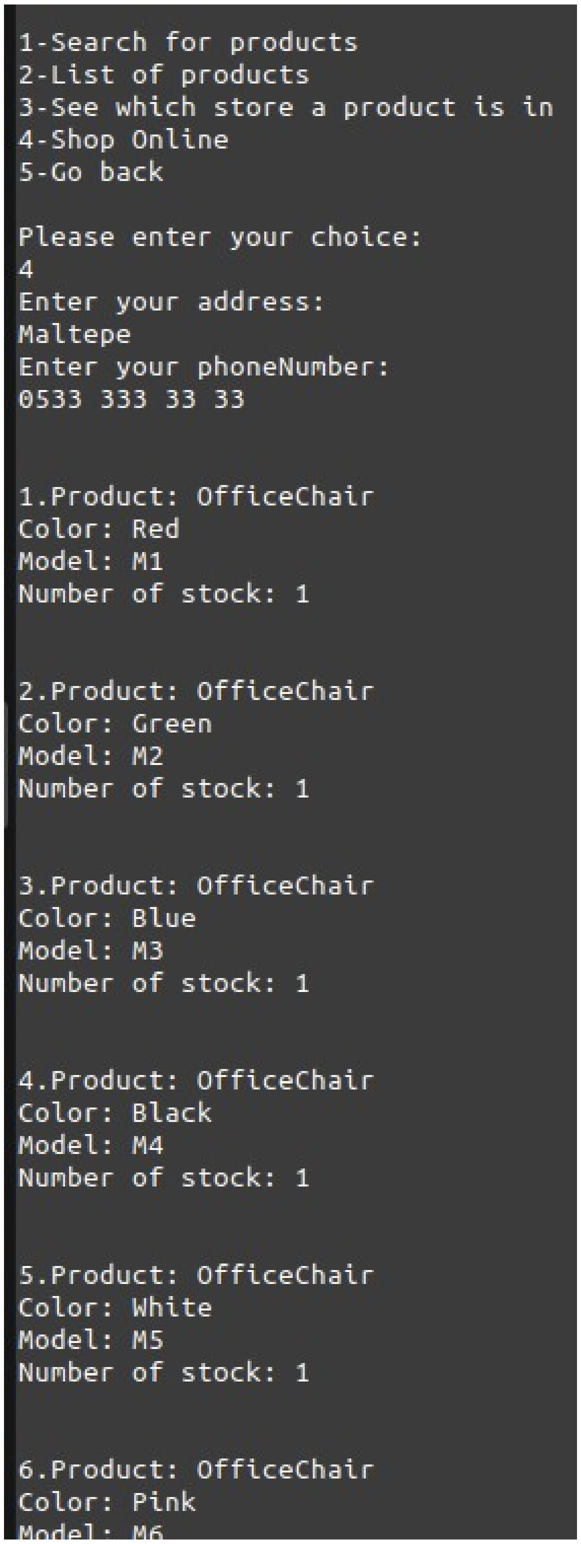

```
29.Product: OfficeCabinet
Color: None
Model: M7
Number of stock: 3
Branch: 1


30.Product: OfficeCabinet
Color: None
Model: M8
Number of stock: 3
Branch: 1


31.Product: OfficeCabinet
Color: None
Model: M9
Number of stock: 3
Branch: 1


32.Product: OfficeCabinet
Color: None
Model: M10
Number of stock: 3
Branch: 1


33.Product: OfficeCabinet
Color: None
Model: M11
Number of stock: 4
Branch: 1


34.Product: OfficeCabinet
Color: None
Model: M12
Number of stock: 4
Branch: 1



1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
```

*if it’s not founded in the store. Then it returns customer menu again!!!



```
1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
3

1 Gebze
2 Darica
3 Cayirova
4 Istanbul

Please enter the name of product to see which store is in:
Dolap

1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
```

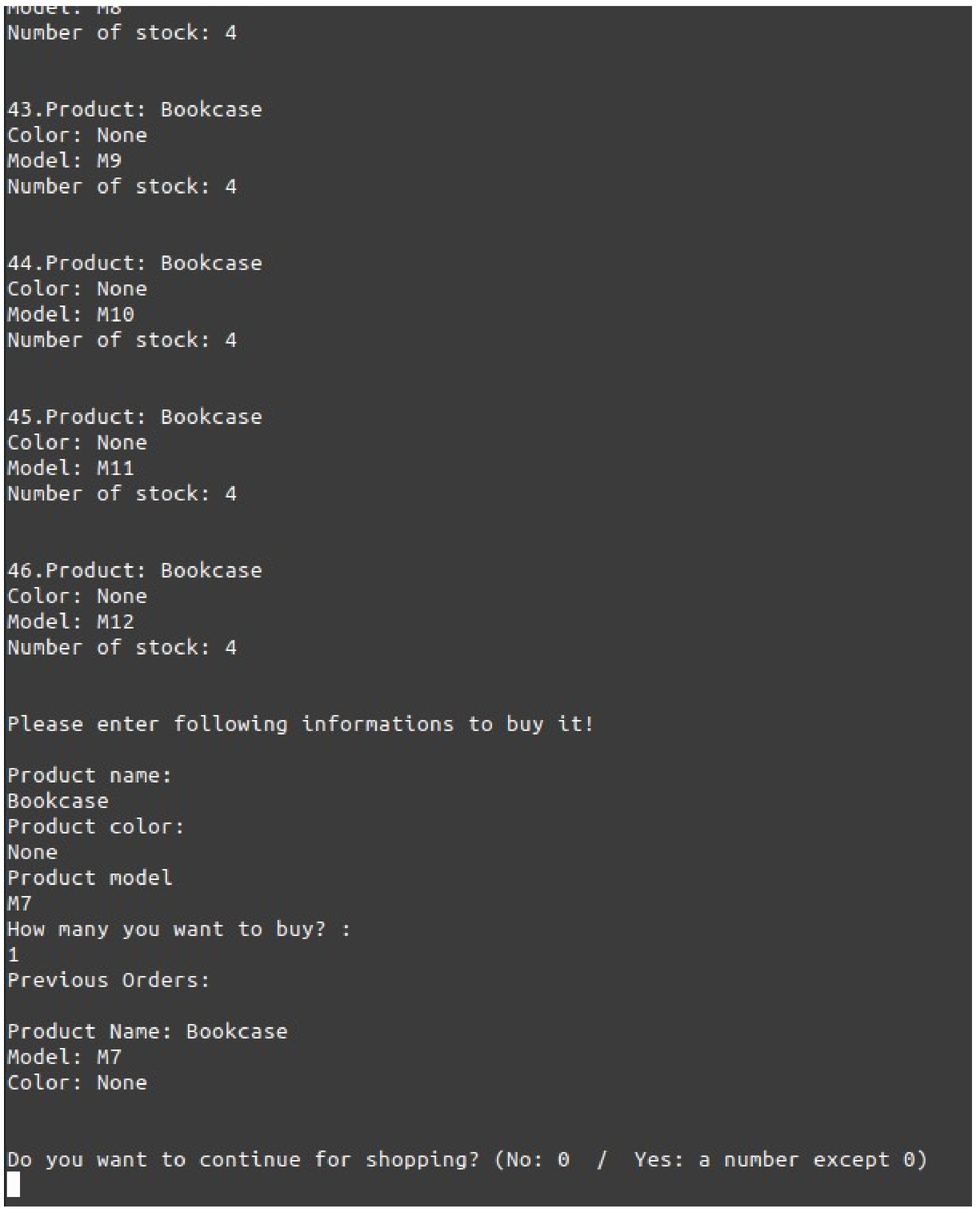
Test Case-25: Shopping online.(customer): passed
*system asks the address and phone information and customer starts the shopping, he/she can show his/her previous orders during the shopping.



```
1-Search for products
2-List of products
3-See which store a product is in
4-Shop Online
5-Go back

Please enter your choice:
4
Enter your address:
Maltepe
Enter your phoneNumber:
0533 333 33 33


1.Product: OfficeChair
Color: Red
Model: M1
Number of stock: 1


2.Product: OfficeChair
Color: Green
Model: M2
Number of stock: 1


3.Product: OfficeChair
Color: Blue
Model: M3
Number of stock: 1


4.Product: OfficeChair
Color: Black
Model: M4
Number of stock: 1


5.Product: OfficeChair
Color: White
Model: M5
Number of stock: 1


6.Product: OfficeChair
Color: Pink
Model: M6
```

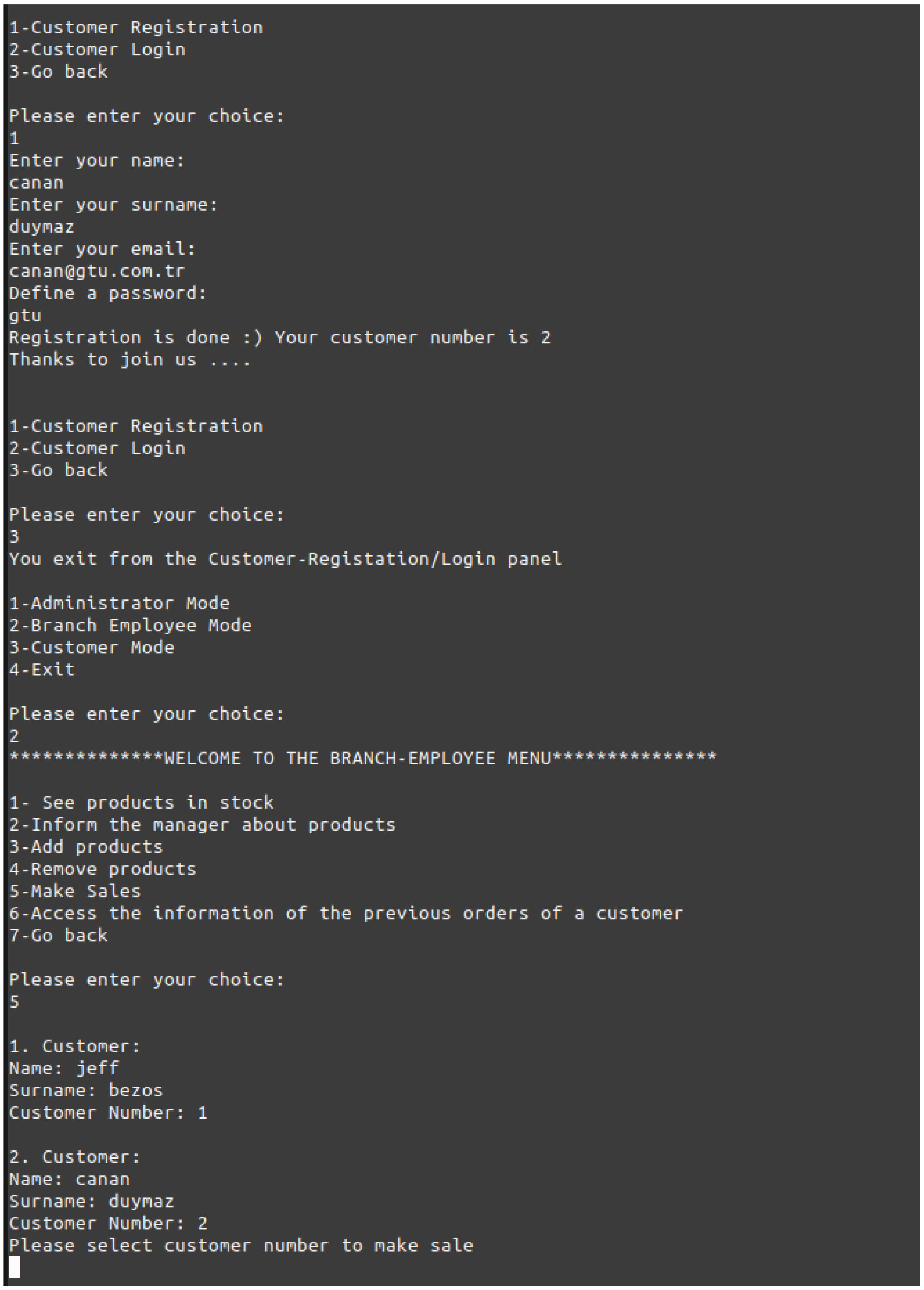
.

.



```
Number of stock: 4


43.Product: Bookcase
Color: None
Model: M9
Number of stock: 4


44.Product: Bookcase
Color: None
Model: M10
Number of stock: 4


45.Product: Bookcase
Color: None
Model: M11
Number of stock: 4


46.Product: Bookcase
Color: None
Model: M12
Number of stock: 4


Please enter following informations to buy it!

Product name:
Bookcase
Product color:
None
Product model
M7
How many you want to buy? :
1
Previous Orders:

Product Name: Bookcase
Model: M7
Color: None


Do you want to continue for shopping? (No: 0  /  Yes: a number except 0)
```

Test Case-26: Adding more than one customer to the system with special customer numbers. (customer): passed

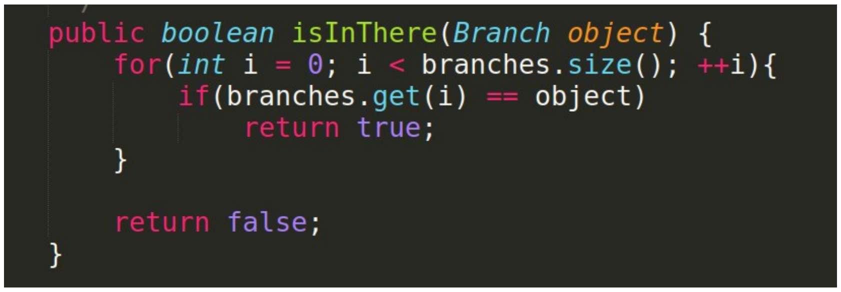


```
1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
1
Enter your name:
canan
Enter your surname:
duymaz
Enter your email:
canan@gtu.com.tr
Define a password:
gtu
Registration is done :) Your customer number is 2
Thanks to join us ....


1-Customer Registration
2-Customer Login
3-Go back

Please enter your choice:
3
You exit from the Customer-Registation/Login panel

1-Administrator Mode
2-Branch Employee Mode
3-Customer Mode
4-Exit

Please enter your choice:
2
**************WELCOME TO THE BRANCH-EMPLOYEE MENU***************

1- See products in stock
2-Inform the manager about products
3-Add products
4-Remove products
5-Make Sales
6-Access the information of the previous orders of a customer
7-Go back

Please enter your choice:
5

1. Customer:
Name: jeff
Surname: bezos
Customer Number: 1

2. Customer:
Name: canan
Surname: duymaz
Customer Number: 2
Please select customer number to make sale
```

*****MY AUTOMATION SYSTEM SOFTWARE(WHICH IS CHANGED BY KWLINKEDLIST,KWARRAYLIST AND HYBRIDLIST IMPLEMENTATION) PASSED FROM ALL TESTS****

PART-2: Analyze the time complexity (except getters,setters and specific constructors)

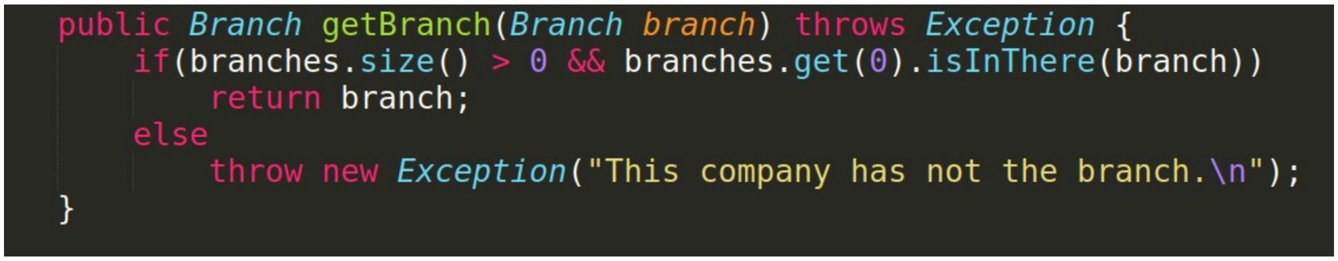
Branch class:



```java
public boolean isInThere(Branch object) {
    for(int i = 0; i < branches.size(); ++i){
        if(branches.get(i) == object)
            return true;
    }

    return false;
}
```

*there is just one for loop and if condition. If we analyze the time complexity of the for loop then we can say that worst case is O(n) because turns until the value of branches.size() → n (there will be best and worst cases because of "return", best case is TETA(1)) and we can also say that if condition is also O(n) because get method of linkedlist is O(n). That's why time complexity is equal to
T(n) = O(N)*O(n) = O(n^2)

Company Class:

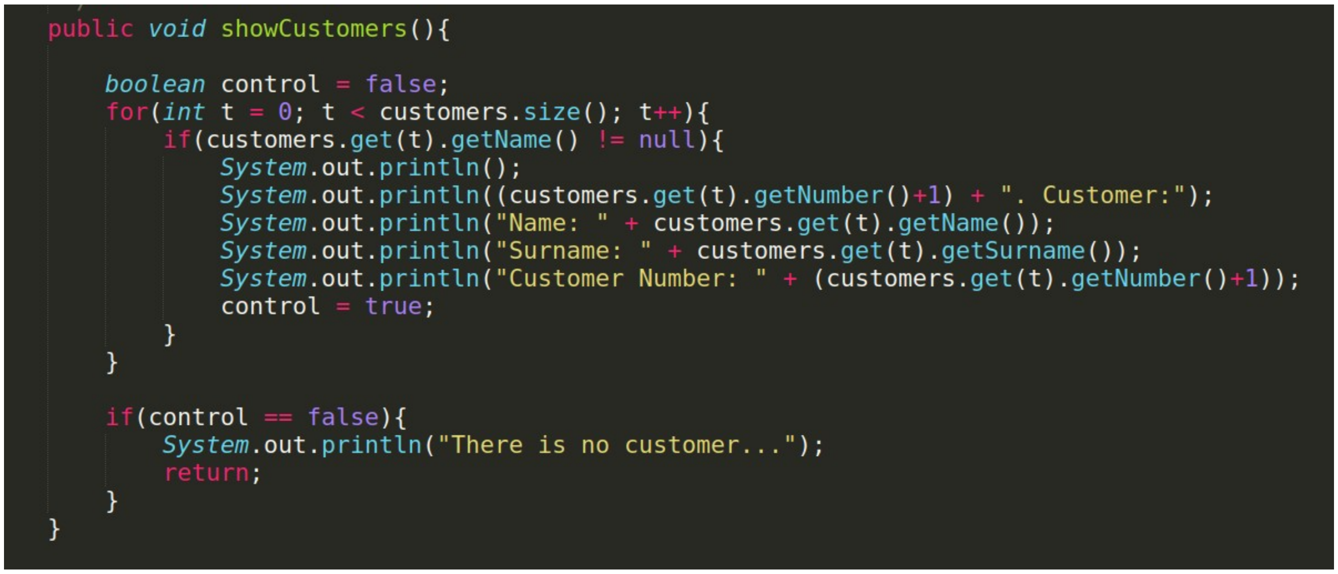


```java
public Branch getBranch(Branch branch) throws Exception {
    if(branches.size() > 0 && branches.get(0).isInThere(branch))
        return branch;
    else
        throw new Exception("This company has not the branch.\n");
}
```

*we know that from our previous complexity calculation, isInThere(branch) is O(n^2). and in this case, we have best case and worst case because of if statement.

T(n)best = TETA(1) → else statement
T(n)worst = O(n)+O(n^2) = O(n^2).

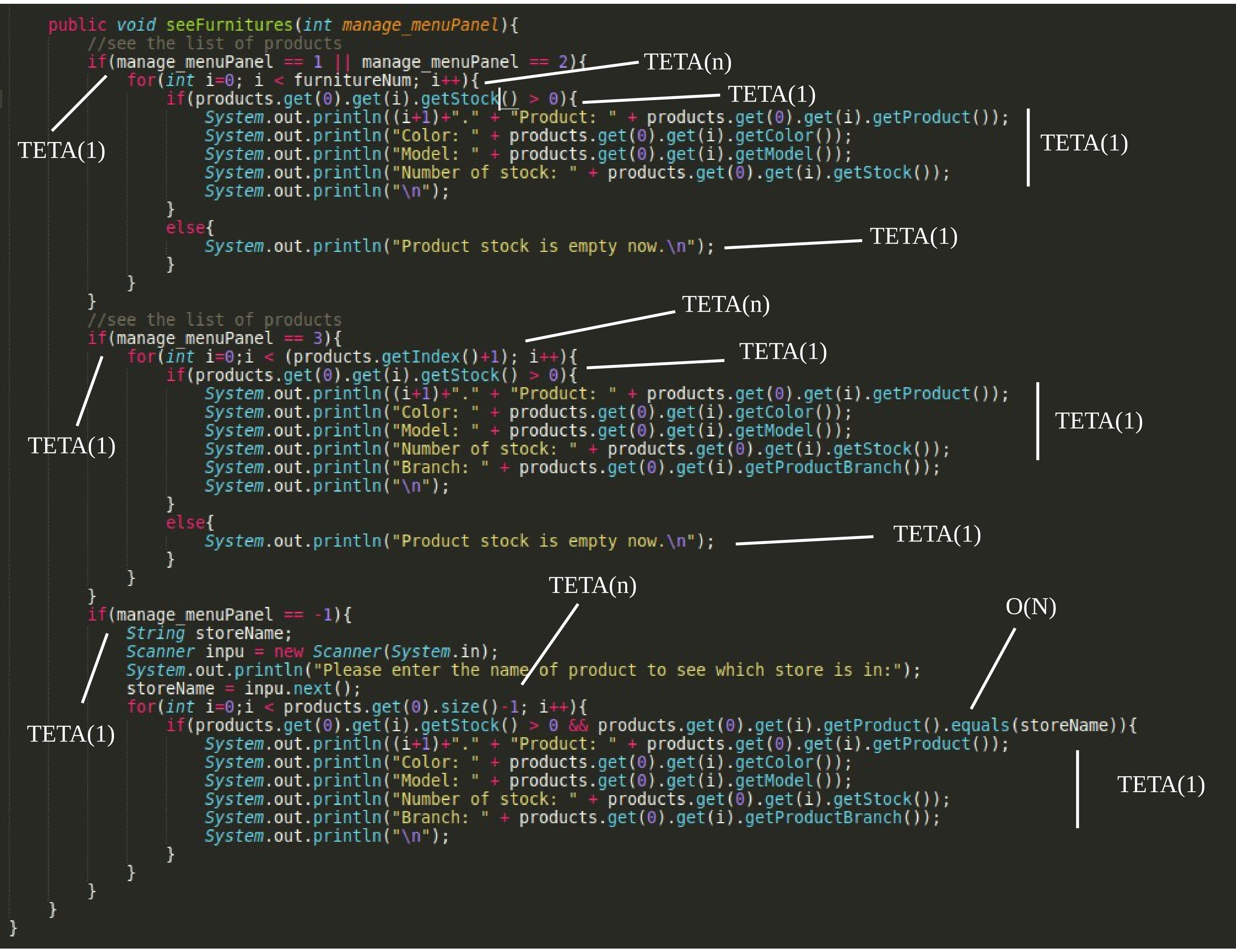
T(n) = O(N^2)



```java
public void showCustomers(){

    boolean control = false;
    for(int t = 0; t < customers.size(); t++){
        if(customers.get(t).getName() != null){
            System.out.println();
            System.out.println((customers.get(t).getNumber()+1) + ". Customer:");
            System.out.println("Name: " + customers.get(t).getName());
            System.out.println("Surname: " + customers.get(t).getSurname());
            System.out.println("Customer Number: " + (customers.get(t).getNumber()+1));
            control = true;
        }
    }

    if(control == false){
        System.out.println("There is no customer...");
        return;
    }
}
```

*boolean control will be TETA(1) constant time. For loop turns until value of customers.size() so we can say that it's TETA(n). customers is an arraylist and we know that, get(index) method is constant time in arraylist so if statement should be teta(1). Println() parts are also TETA(1) because of the same reason("get" method of arraylist). After that there is a "if" condition and it's also TETA(1) because there is no loop etc. Therefore,

T(n) = TETA(1) + TETA(n) + TETA(1) = TETA(n)
T(n) = TETA(n).

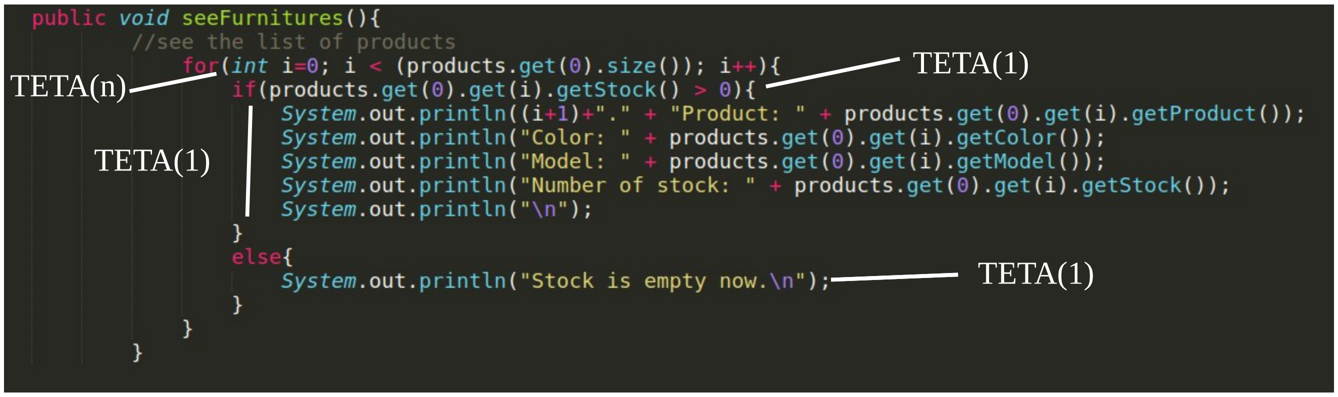


```
    public void seeFurnitures(int manage_menuPanel){
        //see the list of products
        if(manage_menuPanel == 1 || manage_menuPanel == 2){
            for(int i=0; i < furnitureNum; i++){
                if(products.get(0).get(i).getStock() > 0){
                    System.out.println((i+1)+"." + "Product: " + products.get(0).get(i).getProduct());
                    System.out.println("Color: " + products.get(0).get(i).getColor());
                    System.out.println("Model: " + products.get(0).get(i).getModel());
                    System.out.println("Number of stock: " + products.get(0).get(i).getStock());
                    System.out.println("\n");
                }
                else{
                    System.out.println("Product stock is empty now.\n");
                }
            }
        }
        //see the list of products
        if(manage_menuPanel == 3){
            for(int i=0;i < (products.getIndex()+1); i++){
                if(products.get(0).get(i).getStock() > 0){
                    System.out.println((i+1)+"." + "Product: " + products.get(0).get(i).getProduct());
                    System.out.println("Color: " + products.get(0).get(i).getColor());
                    System.out.println("Model: " + products.get(0).get(i).getModel());
                    System.out.println("Number of stock: " + products.get(0).get(i).getStock());
                    System.out.println("Branch: " + products.get(0).get(i).getProductBranch());
                    System.out.println("\n");
                }
                else{
                    System.out.println("Product stock is empty now.\n");
                }
            }
        }
        if(manage_menuPanel == -1){
            String storeName;
            Scanner inpu = new Scanner(System.in);
            System.out.println("Please enter the name of product to see which store is in:");
            storeName = inpu.next();
            for(int i=0;i < products.get(0).size()-1; i++){
                if(products.get(0).get(i).getStock() > 0 && products.get(0).get(i).getProduct().equals(storeName)){
                    System.out.println((i+1)+"." + "Product: " + products.get(0).get(i).getProduct());
                    System.out.println("Color: " + products.get(0).get(i).getColor());
                    System.out.println("Model: " + products.get(0).get(i).getModel());
                    System.out.println("Number of stock: " + products.get(0).get(i).getStock());
                    System.out.println("Branch: " + products.get(0).get(i).getProductBranch());
                    System.out.println("\n");
                }
            }
        }
    }
}
```

*Last if condition will be O(n^2) because of equals method, if condition will be O(n) istead of teta(1) so last part of the code will be O(n^2).

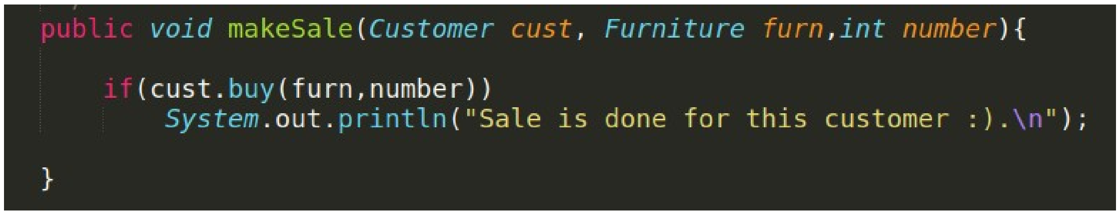
*Other if statements will be Teta(n) * Teta(1)
Therefore T(n) = O(n^2)

BranchEmployee class:



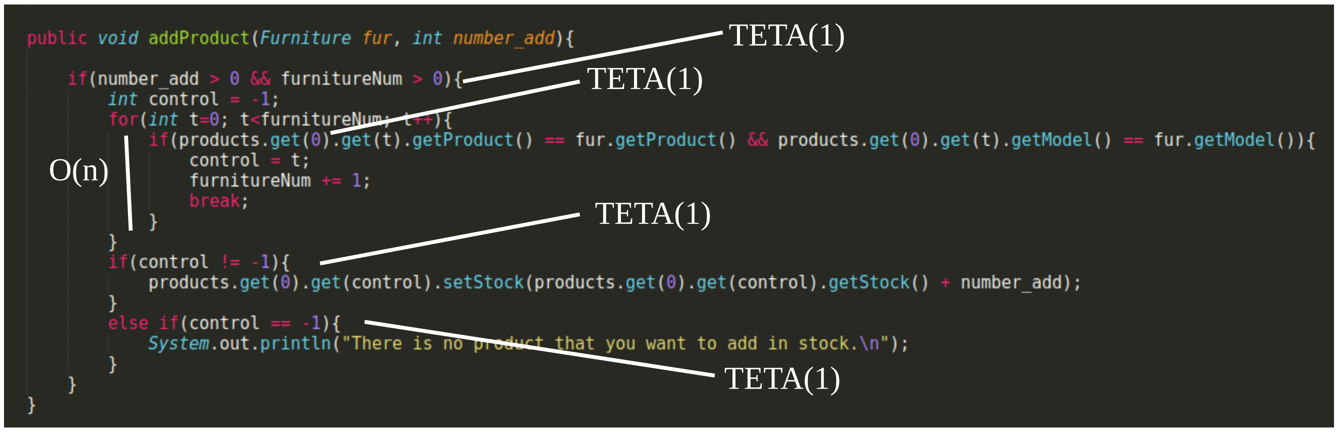
```java
public void seeFurnitures(){
        //see the list of products
        for(int i=0; i < (products.get(0).size()); i++){
            if(products.get(0).get(i).getStock() > 0){
                System.out.println((i+1)+"." + "Product: " + products.get(0).get(i).getProduct());
                System.out.println("Color: " + products.get(0).get(i).getColor());
                System.out.println("Model: " + products.get(0).get(i).getModel());
                System.out.println("Number of stock: " + products.get(0).get(i).getStock());
                System.out.println("\n");
            }
            else{
                System.out.println("Stock is empty now.\n");
            }
        }
    }
```

*T(n) = Teta(n) * T(1) = Teta(n)
T(n) = Teta(n).



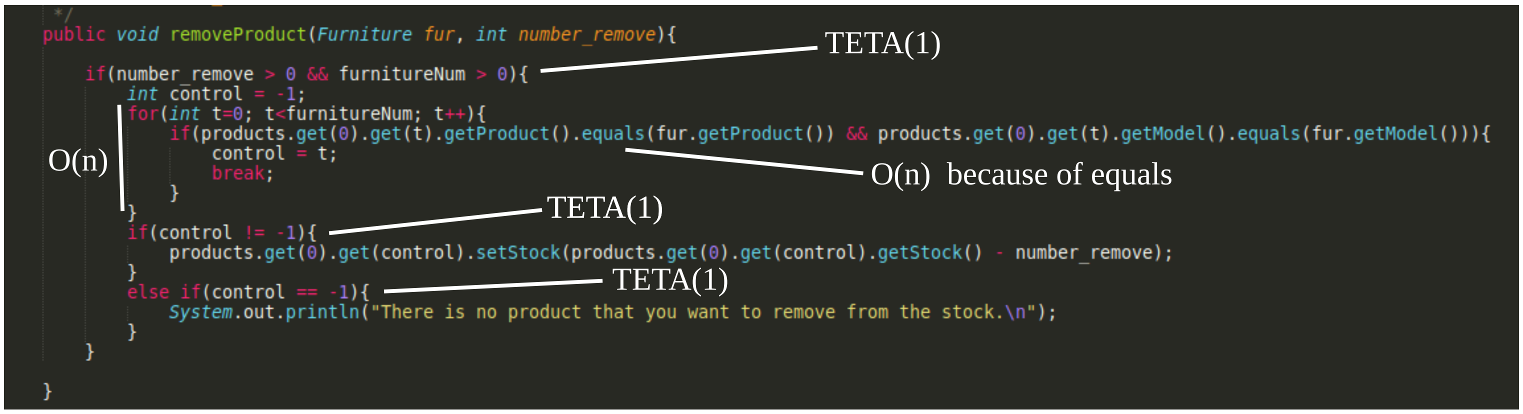
```java
public void makeSale(Customer cust, Furniture furn,int number){

    if(cust.buy(furn,number))
        System.out.println("Sale is done for this customer :).\n");

}
```

*”buy” method is O(n^2) Therefore,
T(n) = O(n^2) in this case



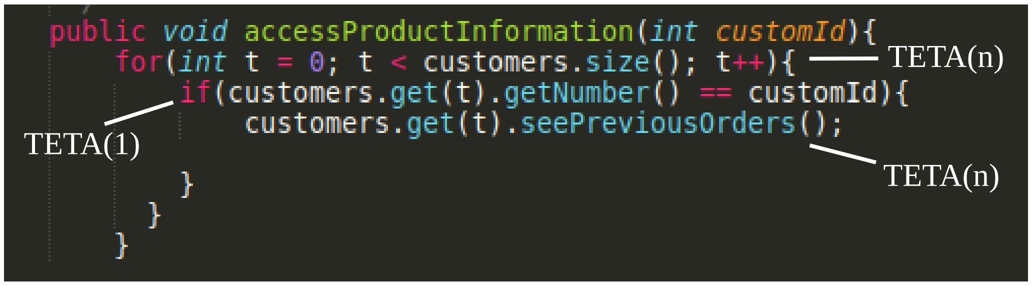
```java
public void addProduct(Furniture fur, int number_add){

    if(number_add > 0 && furnitureNum > 0){
        int control = -1;
        for(int t=0; t<furnitureNum; t++){
            if(products.get(0).get(t).getProduct() == fur.getProduct() && products.get(0).get(t).getModel() == fur.getModel()){
                control = t;
                furnitureNum += 1;
                break;
            }
        }
        if(control != -1){
            products.get(0).get(control).setStock(products.get(0).get(control).getStock() + number_add);
        }
        else if(control == -1){
            System.out.println("There is no product that you want to add in stock.\n");
        }
    }
}
```

Therefore, T(n) = O(n)



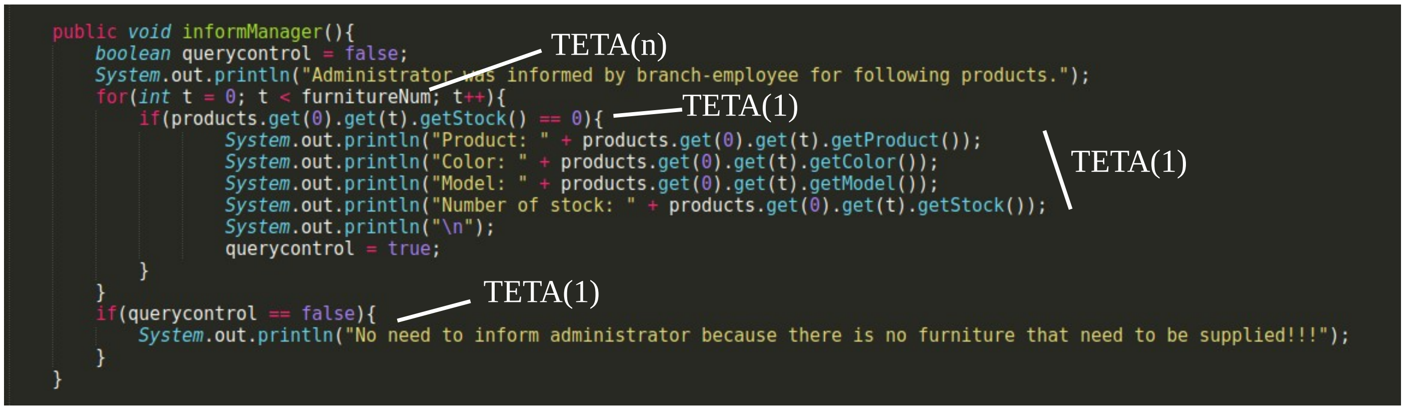
```
public void removeProduct(Furniture fur, int number_remove){

    if(number_remove > 0 && furnitureNum > 0){
        int control = -1;
        for(int t=0; t<furnitureNum; t++){
            if(products.get(0).get(t).getProduct().equals(fur.getProduct()) && products.get(0).get(t).getModel().equals(fur.getModel())){
                control = t;
                break;
            }
        }
        if(control != -1){
            products.get(0).get(control).setStock(products.get(0).get(control).getStock() - number_remove);
        }
        else if(control == -1){
            System.out.println("There is no product that you want to remove from the stock.\n");
        }
    }

}
```

Therefore, T(n) = O(n)*O(n) + ….. =O(n^2)
T(n) = O(n^2)



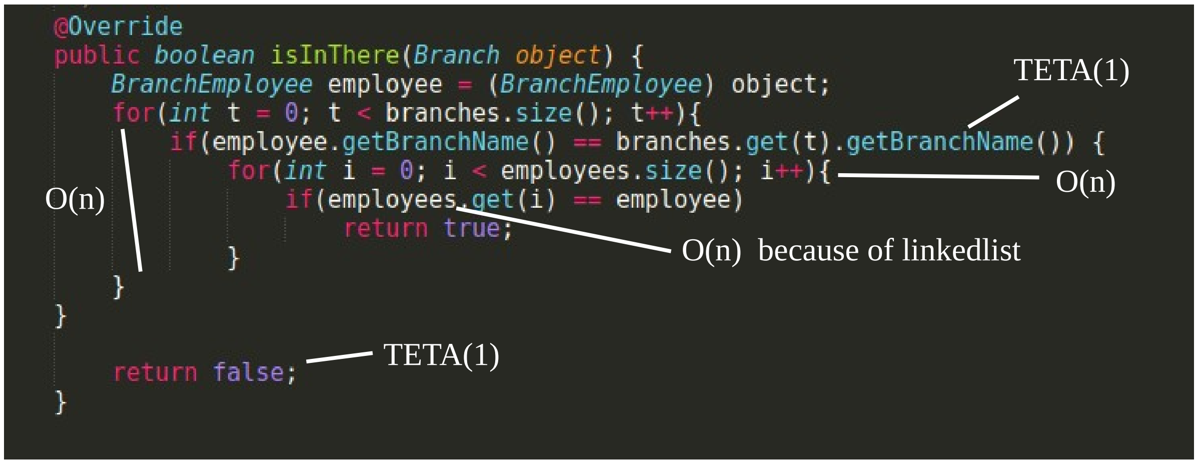
```
public void accessProductInformation(int customId){
    for(int t = 0; t < customers.size(); t++){
        if(customers.get(t).getNumber() == customId){
            customers.get(t).seePreviousOrders();

        }
    }
}
```

Therefore, T(n) = TETA(n) * TETA(1)
T(n) = TETA(n)



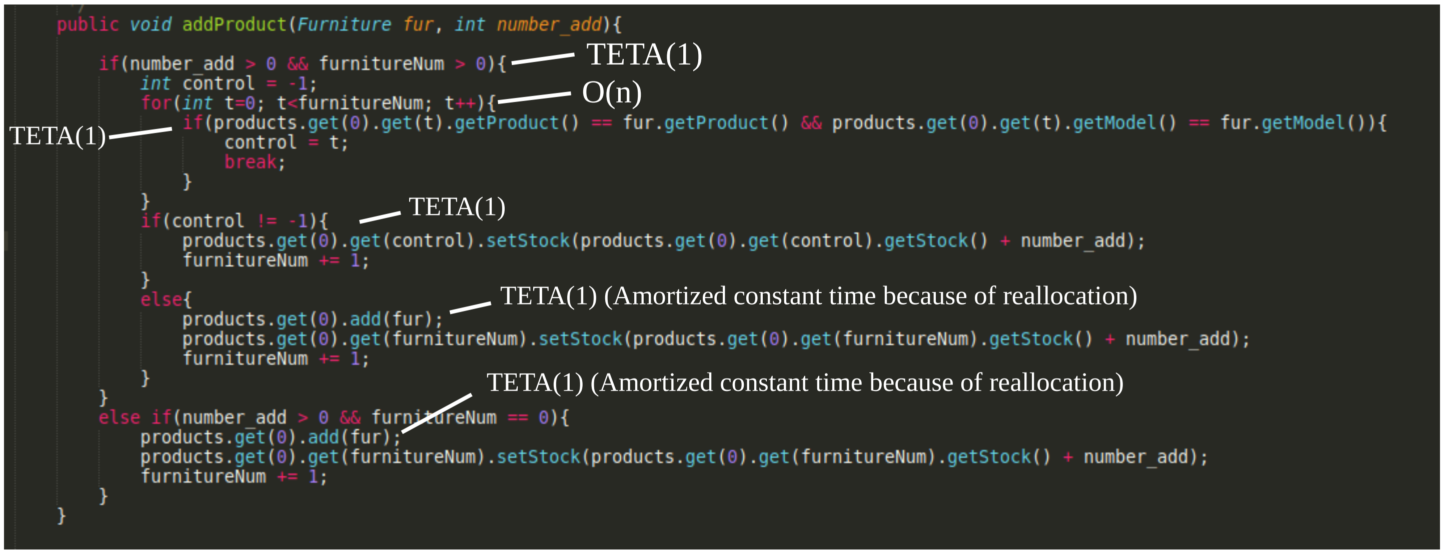
```
public void informManager(){
    boolean querycontrol = false;
    System.out.println("Administrator was informed by branch-employee for following products.");
    for(int t = 0; t < furnitureNum; t++){
        if(products.get(0).get(t).getStock() == 0){
                System.out.println("Product: " + products.get(0).get(t).getProduct());
                System.out.println("Color: " + products.get(0).get(t).getColor());
                System.out.println("Model: " + products.get(0).get(t).getModel());
                System.out.println("Number of stock: " + products.get(0).get(t).getStock());
                System.out.println("\n");
                querycontrol = true;
        }
    }
    if(querycontrol == false){
        System.out.println("No need to inform administrator because there is no furniture that need to be supplied!!!");
    }
}
```

* T(n) = TETA(n)*TETA(1) = TETA(n)
T(n) = TETA(n)

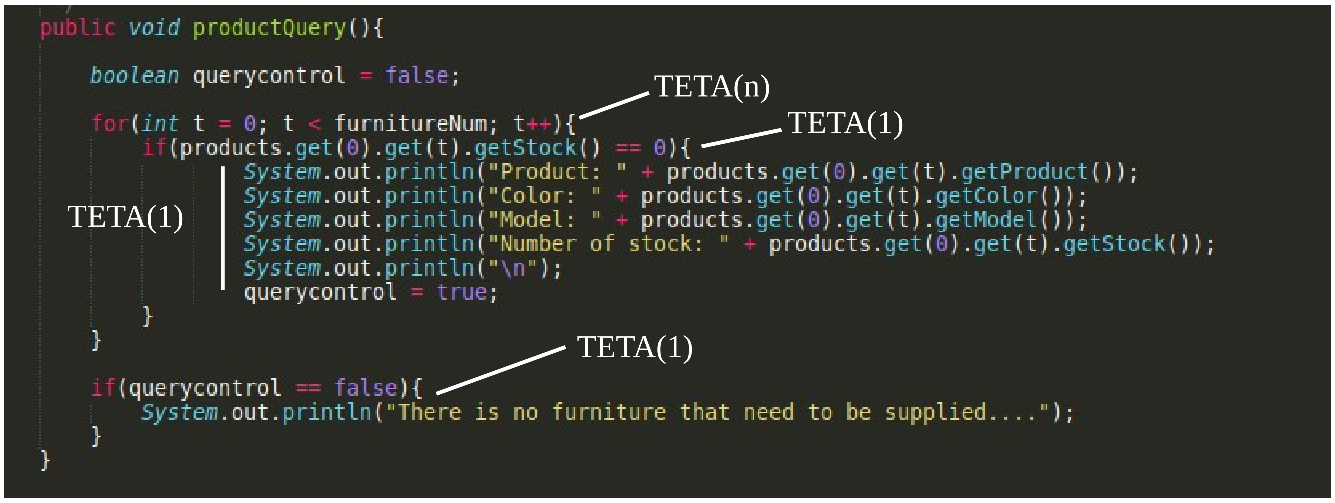

```java
@Override
public boolean isInThere(Branch object) {
    BranchEmployee employee = (BranchEmployee) object;
    for(int t = 0; t < branches.size(); t++){
        if(employee.getBranchName() == branches.get(t).getBranchName()) {
            for(int i = 0; i < employees.size(); i++){
                if(employees.get(i) == employee)
                    return true;
            }
        }
    }

    return false;
}
```

*T(n) = O(n) + O(n) * O(n) = O(n^2)
T(n) = O(n^2)

Administrator class:


```java
public void addProduct(Furniture fur, int number_add){

    if(number_add > 0 && furnitureNum > 0){
        int control = -1;
        for(int t=0; t<furnitureNum; t++){
            if(products.get(0).get(t).getProduct() == fur.getProduct() && products.get(0).get(t).getModel() == fur.getModel()){
                control = t;
                break;
            }
        }
        if(control != -1){
            products.get(0).get(control).setStock(products.get(0).get(control).getStock() + number_add);
            furnitureNum += 1;
        }
        else{
            products.get(0).add(fur);
            products.get(0).get(furnitureNum).setStock(products.get(0).get(furnitureNum).getStock() + number_add);
            furnitureNum += 1;
        }
    }
    else if(number_add > 0 && furnitureNum == 0){
        products.get(0).add(fur);
        products.get(0).get(furnitureNum).setStock(products.get(0).get(furnitureNum).getStock() + number_add);
        furnitureNum += 1;
    }
}
```
TETA(1)
O(n)
TETA(1)
TETA(1)
TETA(1) (Amortized constant time because of reallocation)
TETA(1) (Amortized constant time because of reallocation)

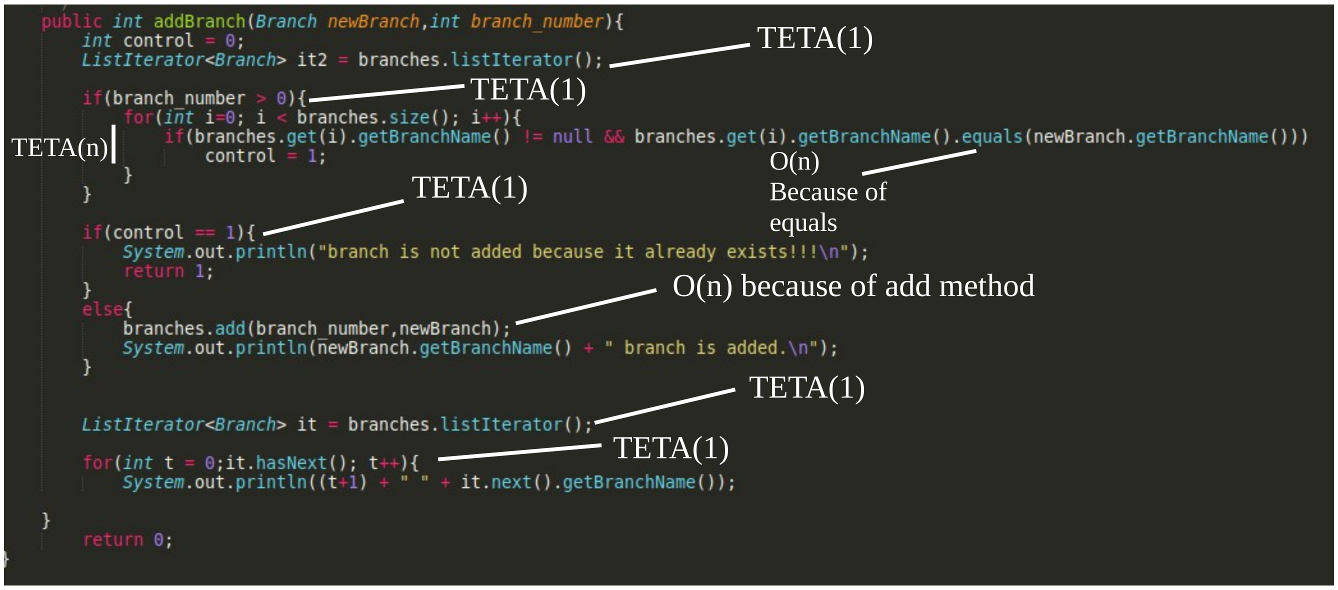
T(n) = TETA(1) + O(n) * TETA(1) = O(n)
T(n) = O(n)



```java
public void productQuery(){

    boolean querycontrol = false;

    for(int t = 0; t < furnitureNum; t++){
        if(products.get(0).get(t).getStock() == 0){
            System.out.println("Product: " + products.get(0).get(t).getProduct());
            System.out.println("Color: " + products.get(0).get(t).getColor());
            System.out.println("Model: " + products.get(0).get(t).getModel());
            System.out.println("Number of stock: " + products.get(0).get(t).getStock());
            System.out.println("\n");
            querycontrol = true;
        }
    }

    if(querycontrol == false){
        System.out.println("There is no furniture that need to be supplied....");
    }
}
```

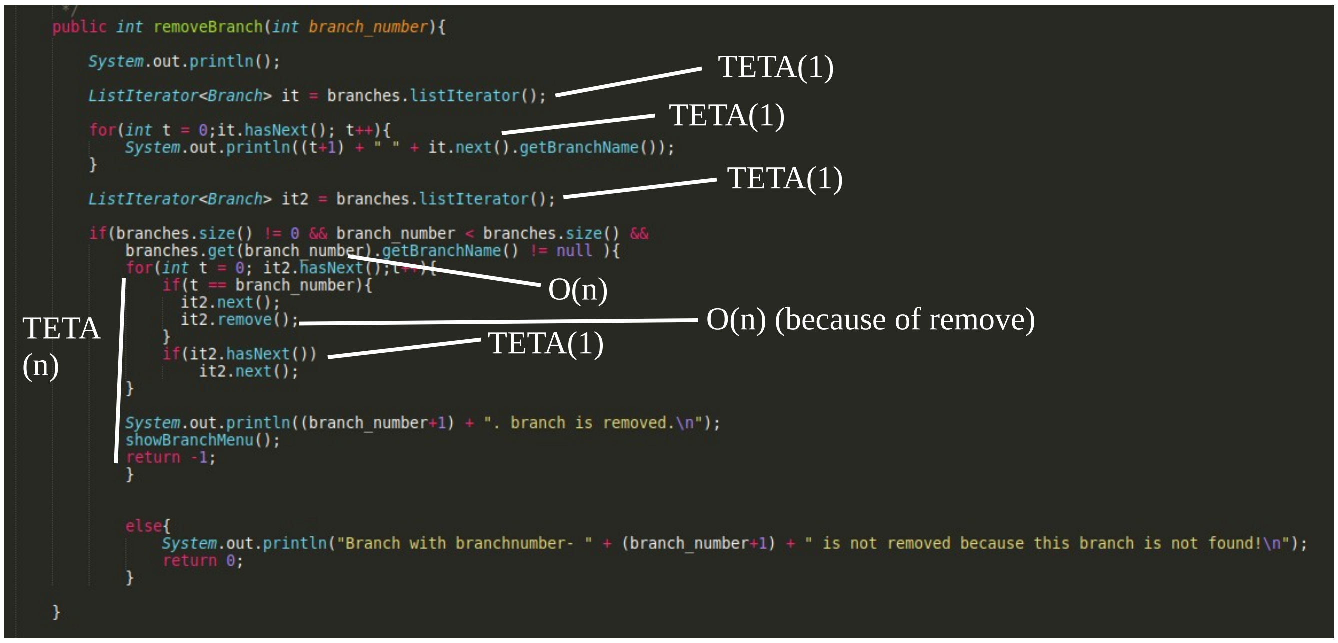
* T(n) = TETA(n)*TETA(1) + TETA(1) = TETA(n)

T(n) = TETA(n)



```java
public int addBranch(Branch newBranch,int branch_number){
    int control = 0;
    ListIterator<Branch> it2 = branches.listIterator();

    if(branch_number > 0){
        for(int i=0; i < branches.size(); i++){
            if(branches.get(i).getBranchName() != null && branches.get(i).getBranchName().equals(newBranch.getBranchName()))
                control = 1;
        }
    }

    if(control == 1){
        System.out.println("branch is not added because it already exists!!!\n");
        return 1;
    }
    else{
        branches.add(branch_number,newBranch);
        System.out.println(newBranch.getBranchName() + " branch is added.\n");
    }


    ListIterator<Branch> it = branches.listIterator();

    for(int t = 0;it.hasNext(); t++){
        System.out.println((t+1) + " " + it.next().getBranchName());

}
    return 0;
}
```

*T(n) = TETA(1) + TETA(n) * O(n)
T(n) = O(n^2)

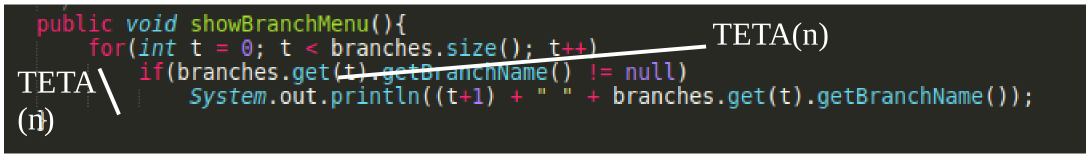


```java
public int removeBranch(int branch_number){

    System.out.println();

    ListIterator<Branch> it = branches.listIterator();

    for(int t = 0;it.hasNext(); t++){
        System.out.println((t+1) + " " + it.next().getBranchName());
    }

    ListIterator<Branch> it2 = branches.listIterator();

    if(branches.size() != 0 && branch_number < branches.size() &&
        branches.get(branch_number).getBranchName() != null ){
        for(int t = 0; it2.hasNext();t++){
            if(t == branch_number){
              it2.next();
              it2.remove();
            }
            if(it2.hasNext())
                it2.next();
        }

        System.out.println((branch_number+1) + ". branch is removed.\n");
        showBranchMenu();
        return -1;
        }


        else{
            System.out.println("Branch with branchnumber- " + (branch_number+1) + " is not removed because this branch is not found!\n");
            return 0;
        }

}
```

*it2.remove() is O(n) so worst case is first if,best case is second if in the for loop,
T(n)best = O(n) + TETA(n)*TETA(1) = O(n)
T(n)best = O(n)

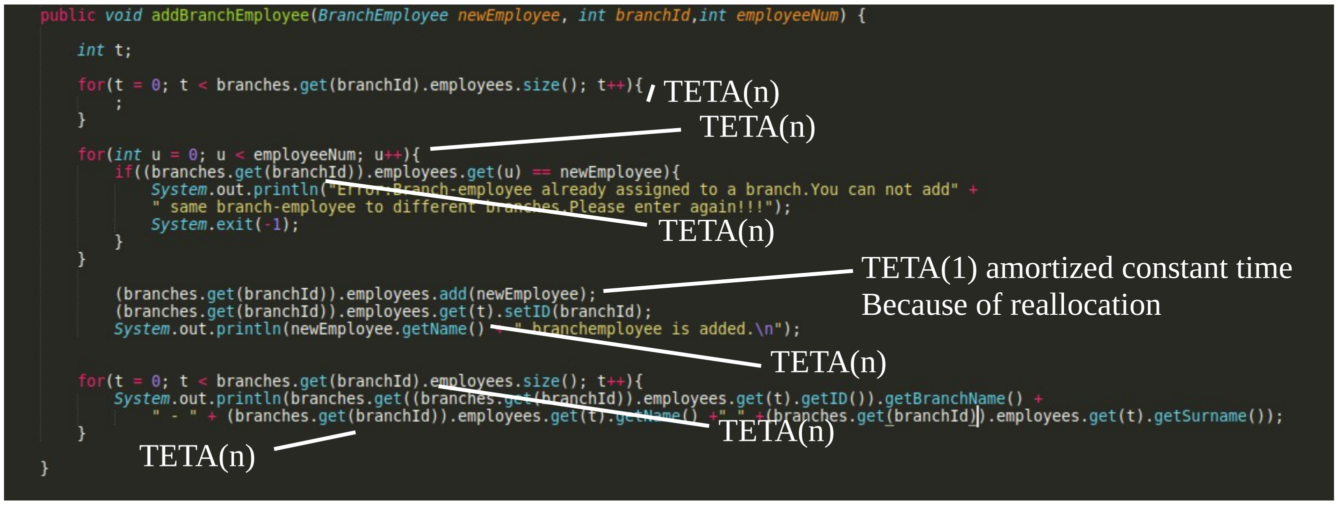
T(n)worst = O(n) + TETA(n)*O(n) = O(n^2)
T(n)worst = O(n^2)

T(n) = O(n^2)



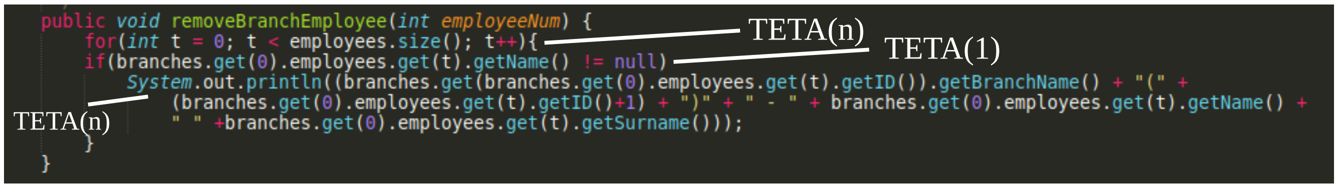
```java
public void showBranchMenu(){
    for(int t = 0; t < branches.size(); t++)
        if(branches.get(t).getBranchName() != null)
            System.out.println((t+1) + " " + branches.get(t).getBranchName());
```

*T(n) = TETA(n) * TETA(n) = TETA(n^2)
T(n) = TETA(n^2)



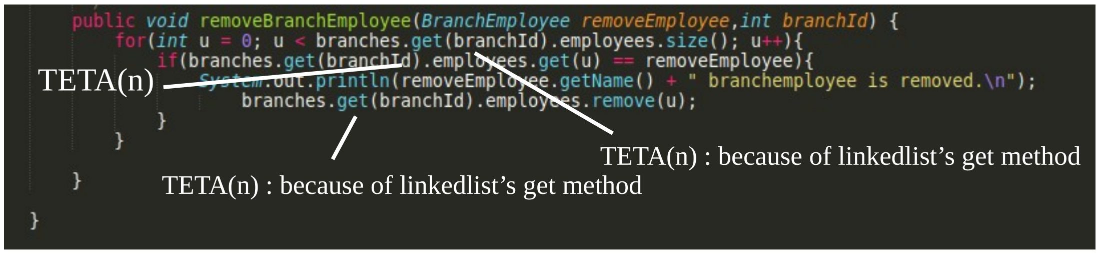
```java
public void addBranchEmployee(BranchEmployee newEmployee, int branchId,int employeeNum) {

    int t;

    for(t = 0; t < branches.get(branchId).employees.size(); t++){
        ;
    }

    for(int u = 0; u < employeeNum; u++){
        if((branches.get(branchId)).employees.get(u) == newEmployee){
            System.out.println("Error:Branch-employee already assigned to a branch.You can not add" +
            " same branch-employee to different branches.Please enter again!!!");
            System.exit(-1);
        }
    }

        (branches.get(branchId)).employees.add(newEmployee);
        (branches.get(branchId)).employees.get(t).setID(branchId);
        System.out.println(newEmployee.getName() + " branchemployee is added.\n");


    for(t = 0; t < branches.get(branchId).employees.size(); t++){
        System.out.println(branches.get((branches.get(branchId)).employees.get(t).getID()).getBranchName() +
            " - " + (branches.get(branchId)).employees.get(t).getName() +" " +(branches.get(branchId)).employees.get(t).getSurname());
    }

}
```

T(n) = TETA(n)*TETA(n) = TETA(n^2)
T(n) = TETA(n^2)



```java
public void removeBranchEmployee(int employeeNum) {
    for(int t = 0; t < employees.size(); t++){
    if(branches.get(0).employees.get(t).getName() != null)
        System.out.println((branches.get(branches.get(0).employees.get(t).getID()).getBranchName() + "(" +
            (branches.get(0).employees.get(t).getID()+1) + ")" + " - " + branches.get(0).employees.get(t).getName() +
            " " +branches.get(0).employees.get(t).getSurname()));
    }
}
```

T(n) = TETA(n)

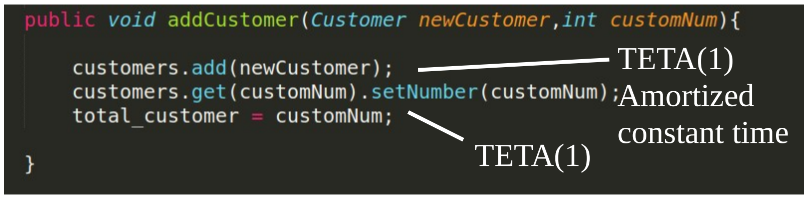


```java
    public void removeBranchEmployee(BranchEmployee removeEmployee,int branchId) {
        for(int u = 0; u < branches.get(branchId).employees.size(); u++){
            if(branches.get(branchId).employees.get(u) == removeEmployee){
                System.out.println(removeEmployee.getName() + " branchemployee is removed.\n");
                branches.get(branchId).employees.remove(u);
            }
        }

    }

}
```

TETA(n)
TETA(n) : because of linkedlist’s get method
TETA(n) : because of linkedlist’s get method

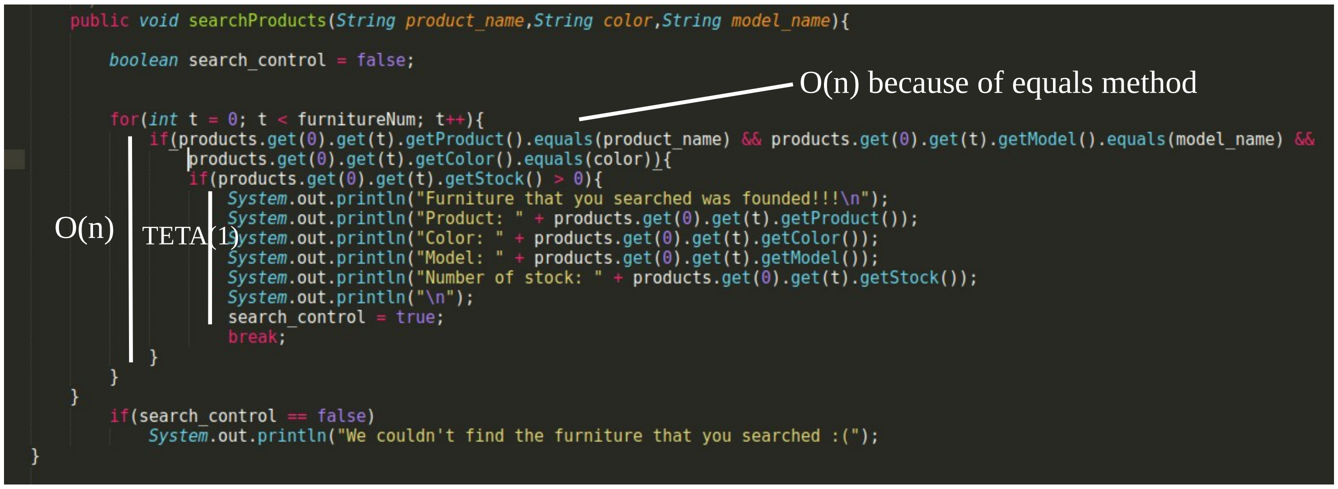
T(n) = TETA(n)*TETA(n) = TETA(n^2)
T(n) = TETA(n^2)

Customer Class:



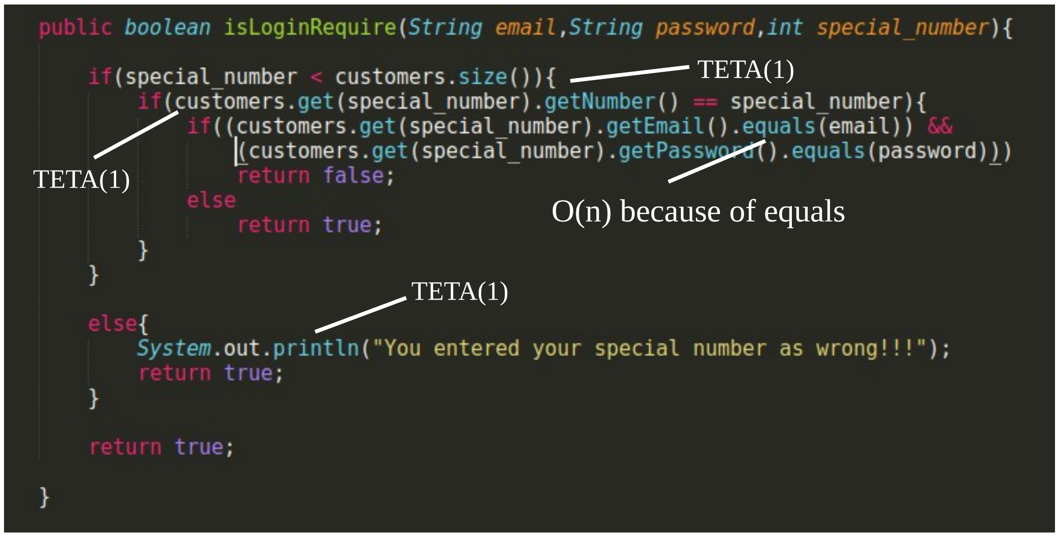
```
public void addCustomer(Customer newCustomer,int customNum){

    customers.add(newCustomer);
    customers.get(customNum).setNumber(customNum);
    total_customer = customNum;

}
```

T(n) = TETA(1)



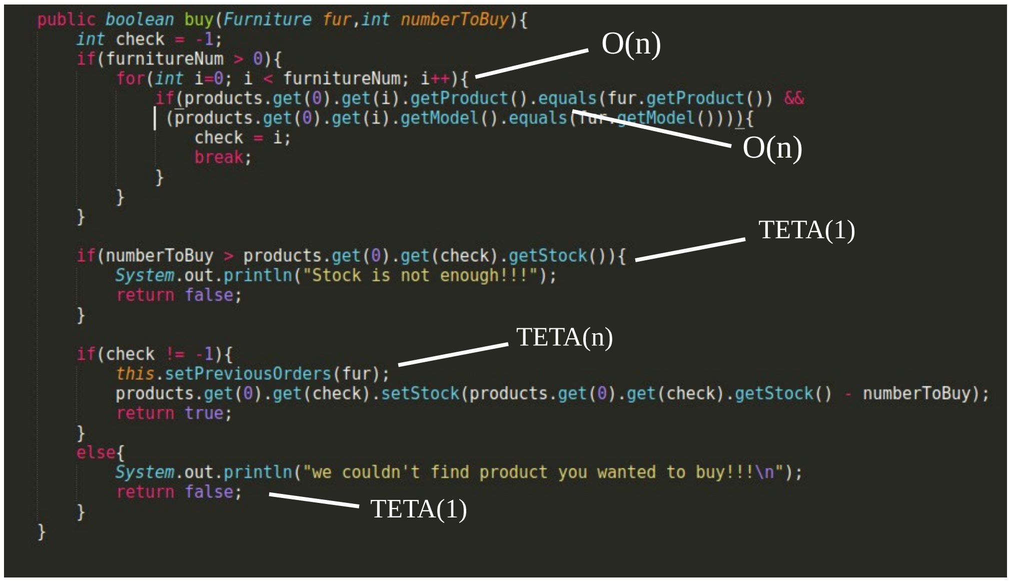
```
public void searchProducts(String product_name,String color,String model_name){

    boolean search_control = false;


    for(int t = 0; t < furnitureNum; t++){
        if(products.get(0).get(t).getProduct().equals(product_name) && products.get(0).get(t).getModel().equals(model_name) &&
            products.get(0).get(t).getColor().equals(color)){
            if(products.get(0).get(t).getStock() > 0){
                System.out.println("Furniture that you searched was founded!!!\n");
                System.out.println("Product: " + products.get(0).get(t).getProduct());
                System.out.println("Color: " + products.get(0).get(t).getColor());
                System.out.println("Model: " + products.get(0).get(t).getModel());
                System.out.println("Number of stock: " + products.get(0).get(t).getStock());
                System.out.println("\n");
                search_control = true;
                break;
            }
        }
    }
    if(search_control == false)
        System.out.println("We couldn't find the furniture that you searched :(");
}
```

* it turns just one for best case and turns n times as worst case,
T(n) = O(n)*O(n) = O(n^2)
T(n) = O(n^2)



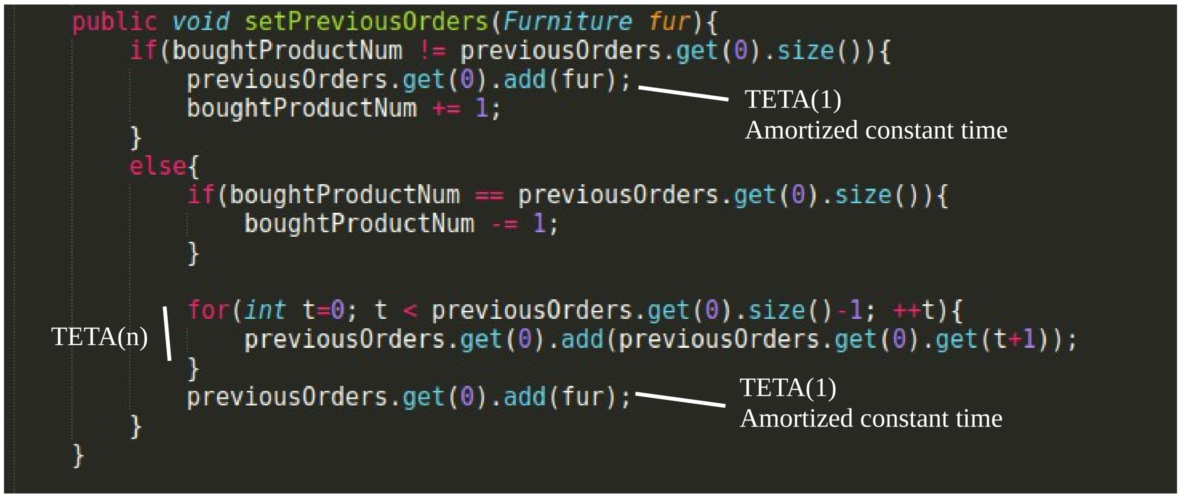
```
public boolean isLoginRequire(String email,String password,int special_number){

    if(special_number < customers.size()){
        if(customers.get(special_number).getNumber() == special_number){
            if((customers.get(special_number).getEmail().equals(email)) &&
                (customers.get(special_number).getPassword().equals(password)))
                return false;
            else
                return true;

        }
    }

    else{
        System.out.println("You entered your special number as wrong!!!");
        return true;
    }

    return true;

}
```

* T(n) = O(n)



```java
public boolean buy(Furniture fur,int numberToBuy){
    int check = -1;
    if(furnitureNum > 0){
        for(int i=0; i < furnitureNum; i++){
            if(products.get(0).get(i).getProduct().equals(fur.getProduct()) &&
            (products.get(0).get(i).getModel().equals(fur.getModel()))){
                check = i;
                break;
            }
        }
    }

    if(numberToBuy > products.get(0).get(check).getStock()){
        System.out.println("Stock is not enough!!!");
        return false;
    }

    if(check != -1){
        this.setPreviousOrders(fur);
        products.get(0).get(check).setStock(products.get(0).get(check).getStock() - numberToBuy);
        return true;
    }
    else{
        System.out.println("we couldn't find product you wanted to buy!!!\n");
        return false;
    }
}
```

*T(n) = O(n) * O(n) = O(n^2)
T(n) = O(n^2)



```java
public void setPreviousOrders(Furniture fur){
    if(boughtProductNum != previousOrders.get(0).size()){
        previousOrders.get(0).add(fur);
        boughtProductNum += 1;
    }
    else{
        if(boughtProductNum == previousOrders.get(0).size()){
            boughtProductNum -= 1;
        }

        for(int t=0; t < previousOrders.get(0).size()-1; ++t){
            previousOrders.get(0).add(previousOrders.get(0).get(t+1));
        }
        previousOrders.get(0).add(fur);
    }
}
```

T(n)best = TETA(1) if “if” condition is true
T(n) worst = TETA(n) if “if” condition is false