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

Ahmet Tuğkan Ayhan
1901042692

#########
# PART 1 #
#########

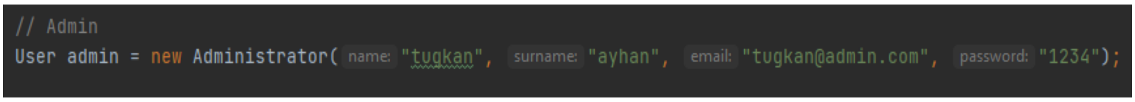
1. System Requirements

1.1. Functional Requirements

This is a furniture company automation system software, and to be able to use it’s functions, an admin and a company for this admin must be created and initialized.

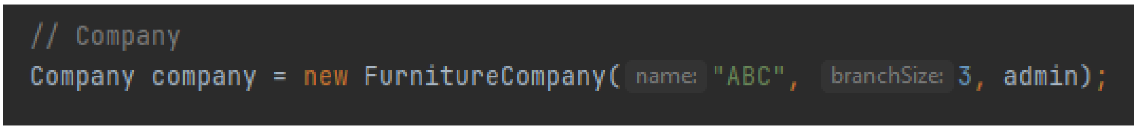
An admin can be initialized by giving it’s properties(name, surname, e-mail and password). E-mail format must contain “@admin.com” otherwise user(which is admin for this part) can’t login to the system.

A furniture company can be initialized by giving it’s properties(company name, initial branch size, admin). Company name can be any string, but initial branch size can not exceed 5 since max branch number is fixed to 5. A company can only have 1 admin at any moment.



```
// Admin
User admin = new Administrator( name: "tugkan",  surname: "ayhan",  email: "tugkan@admin.com",  password: "1234");
```

(a proper initialization for admin)



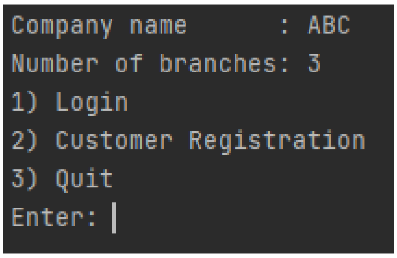
```
// Company
Company company = new FurnitureCompany( name: "ABC",  branchSize: 3, admin);
```

(a proper initialization for furniture company)

After initializing part is done, software is ready to use. In order to access any user’s commands, the current user must login to the system.

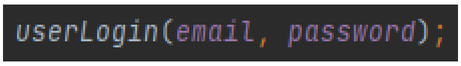
Initially 1 customer, 1 employee and (after you create) 1 admin is registered to the system. After entering correct email and password for the user, system will show the proper command menu according to the user type.

(menu)



```
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 
```

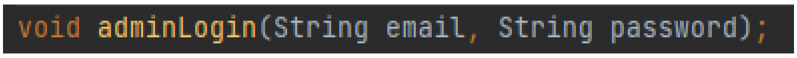
User must call userLogin method in order to get login screen.



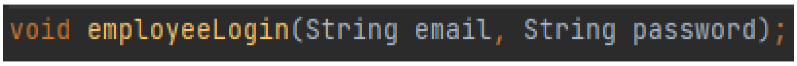
```
userLogin(email, password);
```

System decides which user is entering after getting both email and password.

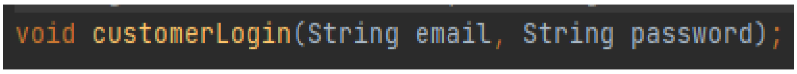
One of the methods below will be executed according to the user type:



```
void adminLogin(String email, String password);
```



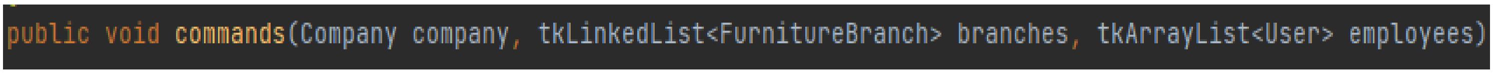
```
void employeeLogin(String email, String password);
```



```
void customerLogin(String email, String password);
```

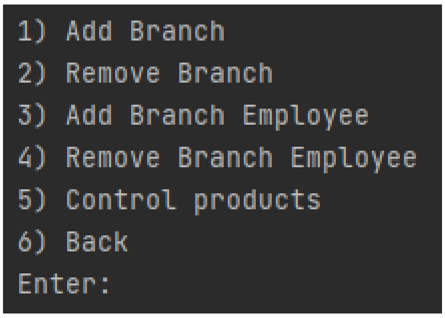
Now we can talk about user’s commands.

They can be accessed by using “commands” method. Method takes company, it’s branches and it’s employees as parameter. Since it is a User interface method it must be called from classes that implement this method.



```
public void commands(Company company, tkLinkedList<FurnitureBranch> branches, tkArrayList<User> employees)
```

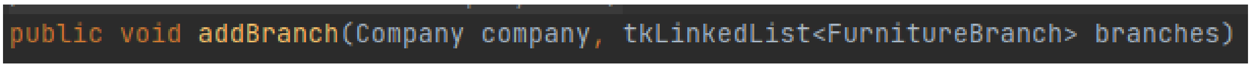
Administrator Commands



```
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter:
```

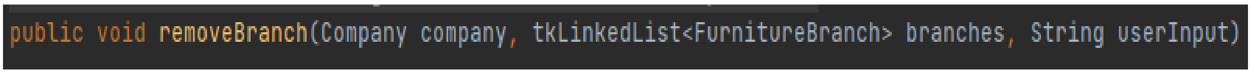
Administrator of the company can add/remove branches, add/remove branch employees and control products. Now let’s see which command uses which methods.

Add Branch must get company and it’s branches as a parameter. Method adds another branch for the company.



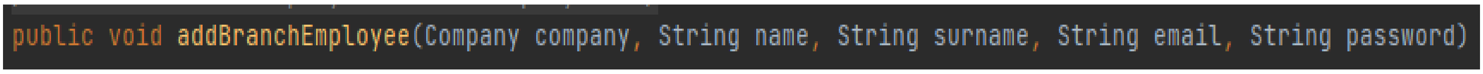
```
public void addBranch(Company company, tkLinkedList<FurnitureBranch> branches)
```

Remove Branch must get company and it’s branches as a parameter. Method removes the branch which selected by user’s itself.



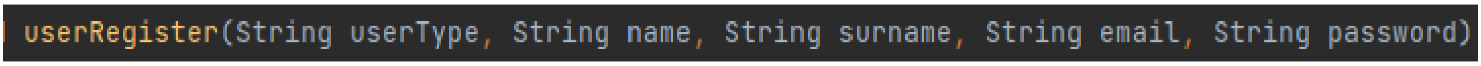
```
public void removeBranch(Company company, tkLinkedList<FurnitureBranch> branches, String userInput)
```

Add Branch Employee requires the company as a parameter. Method adds a branch employee to the company. Name, surname, mail and password must be declared for the branch employee in order to register success.



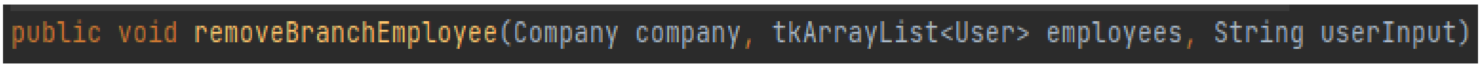
```
public void addBranchEmployee(Company company, String name, String surname, String email, String password)
```

Inside this method, userRegister method is called. It takes userType as parameter and must be given as “employee” since we are adding an employee.



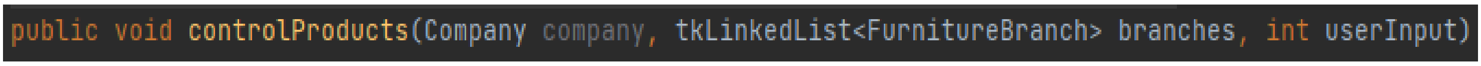
```
userRegister(String userType, String name, String surname, String email, String password)
```

Remove Branch Employee requires company and employee’s as parameters. Method removes the branch employee which user selected.



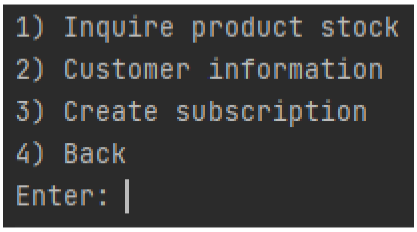
```
public void removeBranchEmployee(Company company, tkArrayList<User> employees, String userInput)
```

Control Products requires company and it’s branches as a parameter. Method calculates number of products that out of stock for the specified branch.



```
public void controlProducts(Company company, tkLinkedList<FurnitureBranch> branches, int userInput)
```

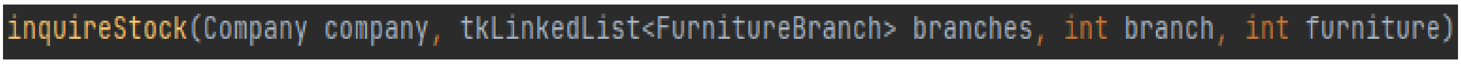
Branch Employee Commands



```
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: |
```

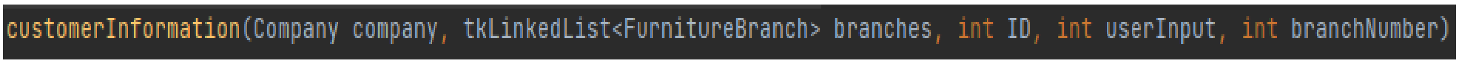
Branch Employee of the company can inquires product stock, can remove or add product from/to stock. Also sells product to customers. At last, he/she can investigate customer’s previous orders.

Inquire Product Stock gets company and it’s branches as a parameter. Method prints list of products from selected branch. Informs admin about products that out of stock. Employee can choose to refill these stocks.



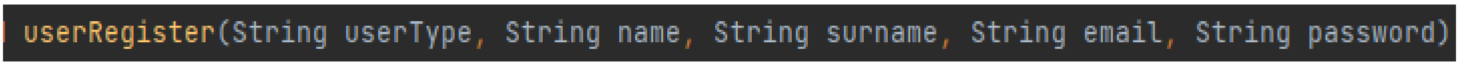
```
inquireStock(Company company, tkLinkedList<FurnitureBranch> branches, int branch, int furniture)
```

Customer Information requires company and it’s branches as a parameter. Method itself asks for customer ID. After finding customer, user of the method can access previous orders of the customer, also user can sell product to this customer by entering proper name, color and model for a product.



```
customerInformation(Company company, tkLinkedList<FurnitureBranch> branches, int ID, int userInput, int branchNumber)
```

Create Subscription calls userRegister method for customer registration.

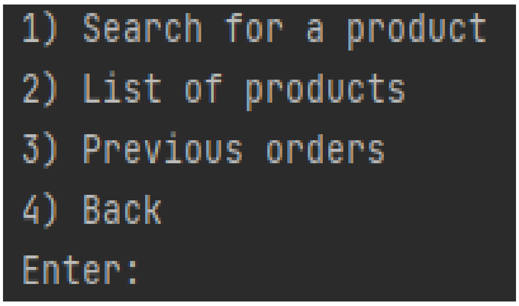


```
userRegister(String userType, String name, String surname, String email, String password)
```

userType is must be given as “customer” since we are adding new customer

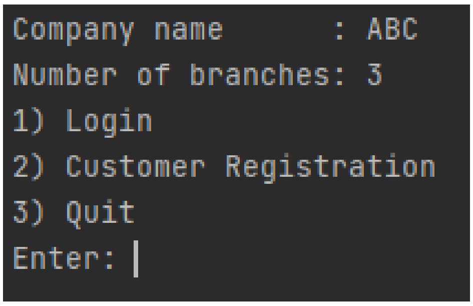
Customer Commands

Below it shows commands menu in Customer class and registration part in the main menu for the customer.



```
1) Search for a product
2) List of products
3) Previous orders
4) Back
Enter:
```

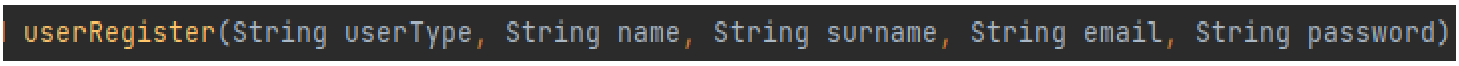
(Customer’s Menu)



```
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 
```

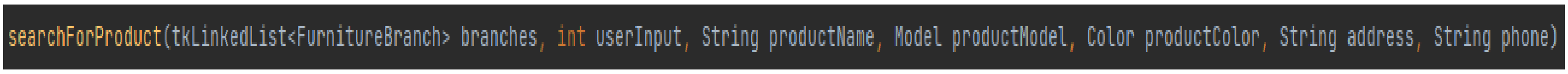
(Main Menu)

Customer Registration acts as same as create subscription method. Calls userRegister method. Parameter must be given as “customer” since user is a customer.



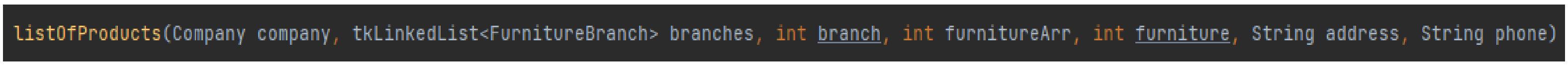
```
userRegister(String userType, String name, String surname, String email, String password)
```

Search For a Product must take company and it’s branches as parameters. After method call, branches will be listed and after branches, furniture types will be listed. Customer can choose a product to buy. It requires address and phone number since it is an online shopping.



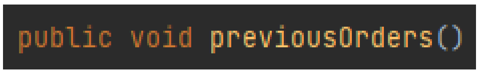
```
searchForProduct(tkLinkedList<FurnitureBranch> branches, int userInput, String productName, Model productModel, Color productColor, String address, String phone)
```

List Of Products takes company and it's branches as parameters as well. Different from **search for a product** it gives list of products to let customer choose product that he/she wants. Given input must be integer. Customer can't choose any product that out of stock.



```
listOfProducts(Company company, tkLinkedList<FurnitureBranch> branches, int branch, int furnitureArr, int furniture, String address, String phone)
```

Previous Orders doesn't take any parameter. Method only shows previous orders of the customer. If there is no order yet, then it will be empty.



```
public void previousOrders()
```

1.2. Non-Functional Requirements

- Initially, there will be 3 branch.
- Each branch will have 1 product for every furniture type, model and color. (It is up to you to increase it)
- addBranch method shouldn't add more than 5 branch.
- removeBranch method shouldn't remove if branch count is 0
- addBranchEmployee method shouldn't add more than 3 employee
- removeBranchEmployee method shouldn't remove if employee count is 0
- controlProducts inputs shouldn't exceed branch number
- Inside all commands method, inputs shouldn't throw exception which unhandled.
- Customer should be able to see their products details after they do any purchase. (online or in-store)
- Customer's shouldn't be able to buy any product that is out of stock.

MAX VALUES

- MAX_BRANCH_NUMBER = 3
- MAX_EMPLOYEE_NUMBER = 3
- MAX_CUSTOMER_NUMBER = 3

These can be changed.

2. Diagrams

2.1. Class Diagram

Class diagram was so large it didn’t fit to this document. You can find it in the homework file as “Class_Diagram”

2.2. Use Case Diagram

Use case diagram was too large for this document as well. You can find it in the homework file as “UseCase_Diagram”

3. Problem Solution Approach

In this homework, problem was communication between user and the company.

Each user, which in this homework they were administrator, branch employee and customer, should’ve their commands to let them communicate with company’s system. So they could add/remove products & employees & branches, see current stock, fill them if they are out of stock, see their previous purchases, etc.

So, Since all commands belong to an individual, I created an interface called User. Then, implemented it for Admin, Branch Employee and Customer. By using polymorphism I used same method(which is commands()) but it’s different implementations.

Secondly, since I am dealing with a furniture company I created a class for it. But it looked so specific to me then I decided to implement it from Company class that I’ve created. Main body of the system was going to be here.

After I implement it, I started to write my Branch class because almost every company have branches and homework was asking for it also. In the branch class I decided to put my Furniture products. There were 5 different product types and I thought writing an interface for Furniture would be better. After writing it I implement Furniture class for these 5 different products.

I also used enumerators since it looked more appropriate to me.

4. Test Cases

I will number the cases so we can easily follow them in the next part

<u>Menu & Admin</u>

1. Compile / **give invalid input #**
2. Compile / Login / **give invalid input #**
3. Compile / Login / Admin / **give invalid input**
4. Compile / Login / Admin / Add Branch / **try to exceed 5**
5. Compile / Login / Admin / Remove Branch / **try to remove while 0 branch**
6. Compile / Login / Admin / Add Branch Employee / **try to exceed 3**
7. Compile / Login / Admin / Remove Branch / **try to remove while 0 employee**
8. Compile / Login / Admin / Control Products / **first control stock in branch1 after that switch to employee and sell something in branch1. At last, login with admin and control stock in branch1 again**
9. Compile / Login / Admin / **Back**

<u>Branch Employee</u>

10. Compile / Login / Employee / **give invalid input**
11. Compile / Login / Employee / Inquire Product / Branch1 / Office Desks / **don't inquire**
12. Compile / Login / Employee / Inquire Product / Branch1 / Office Desks / **inquire** / **try to fill while stocks are full**
13. Compile / Login / Employee / Inquire Product / Branch3 / Bookcase / **First Sell "Bookcase v9 default" to a customer then refill it**
14. Compile / Login / Employee / Customer Information / **give invalid ID**
15. Compile / Login / Employee / **create a customer subscription and sell something to it. Then look his/her previous orders**
16. Compile / Login / Employee / **Back**

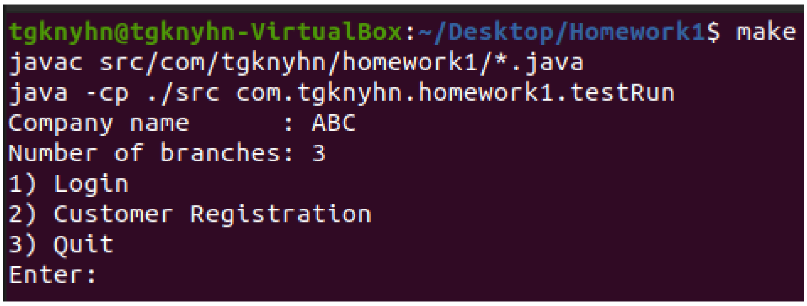
<u>Customer</u> **(all will be in one section which is 18.)**

17. Compile / Register / **Login with new registered customer**
18. Compile / Login / Customer / **give invalid input (18.1)**
19. Compile / Login / Customer / **Search for "Meeting Table v3 black" and buy it. (18.2)**
20. Compile / Login / Customer / **List "Branch 2- Office Desk" and buy v2 red (18.3)**
21. Compile / Login / Customer / **Look to your previous orders (18.4)**
22. Compile / Login / Customer / **Back (18.5)**

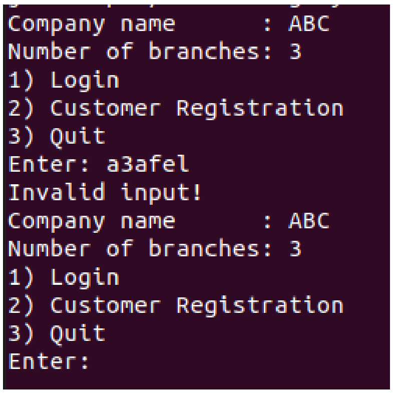
5. Running and Results

Let’s see their result with screenshots

1) Compile / **give invalid input #**

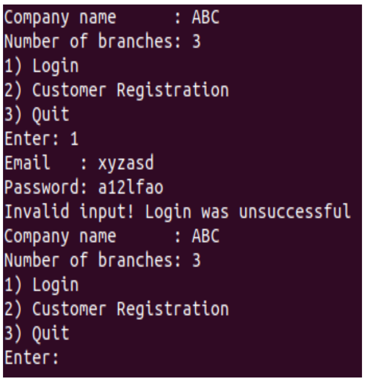


```
tgknyhn@tgknyhn-VirtualBox:~/Desktop/Homework1$ make
javac src/com/tgknyhn/homework1/*.java
java -cp ./src com.tgknyhn.homework1.testRun
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter:
```

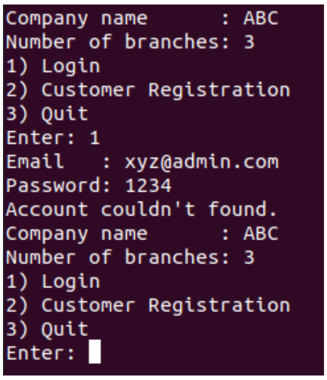


```
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: a3afel
Invalid input!
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter:
```

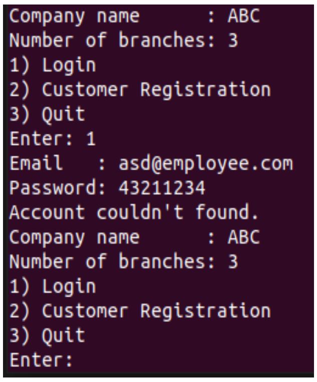
2) Compile / Login / **give invalid input #**



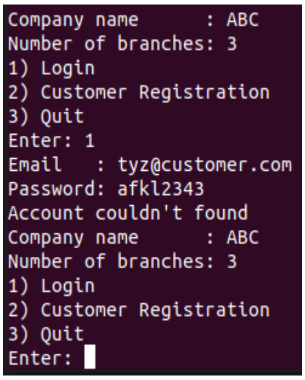
```
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email    : xyzasd
Password: a12lfao
Invalid input! Login was unsuccessful
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter:
```



```
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email    : xyz@admin.com
Password: 1234
Account couldn't found.
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter:
```

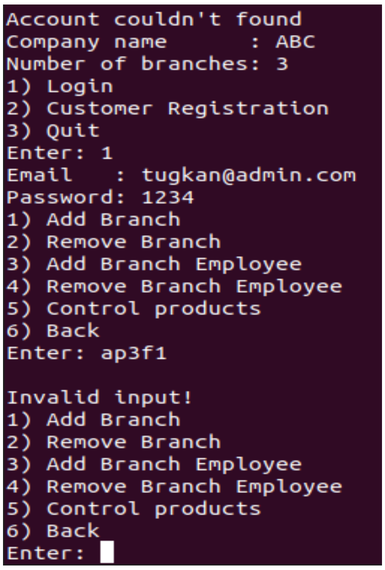


```
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email    : asd@employee.com
Password: 43211234
Account couldn't found.
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter:
```



```
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email    : tyz@customer.com
Password: afkl2343
Account couldn't found
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter:
```

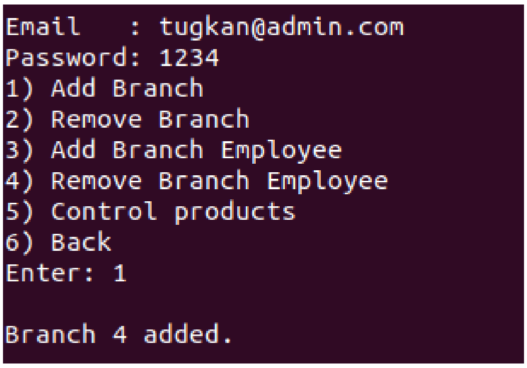
3) Compile / Login / Admin / **give invalid input**



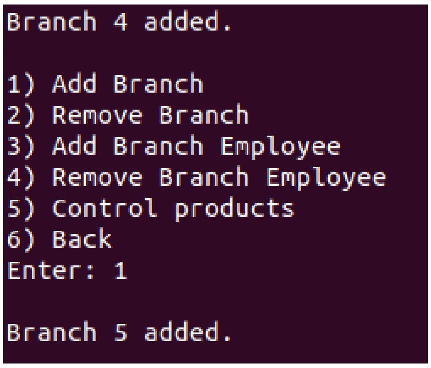
```
Account couldn't found
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email    : tugkan@admin.com
Password: 1234
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: ap3f1

Invalid input!
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter:
```

!! After this I will not include first compiled part to keep it simple

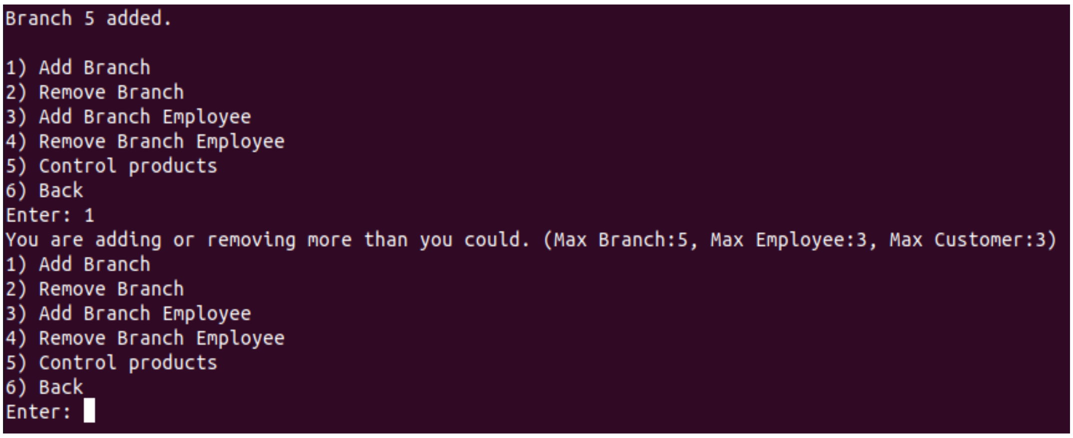
4) Compile / Login / Admin / Add Branch / **try to exceed 5**



```
Email   : tugkan@admin.com
Password: 1234
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 1

Branch 4 added.
```

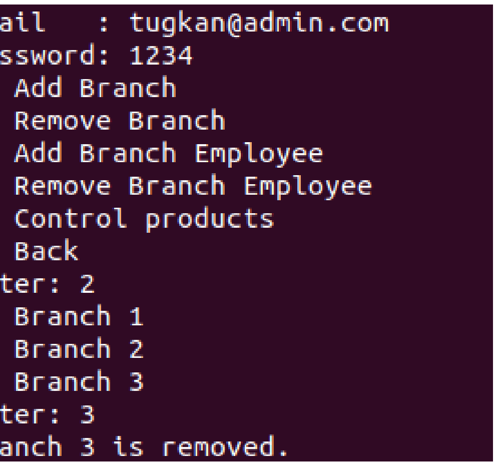


```
Branch 4 added.

1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 1

Branch 5 added.
```

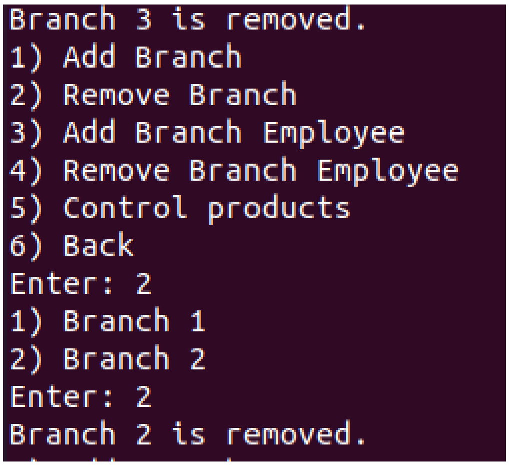


```
Branch 5 added.

1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 1
You are adding or removing more than you could. (Max Branch:5, Max Employee:3, Max Customer:3)
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter:
```

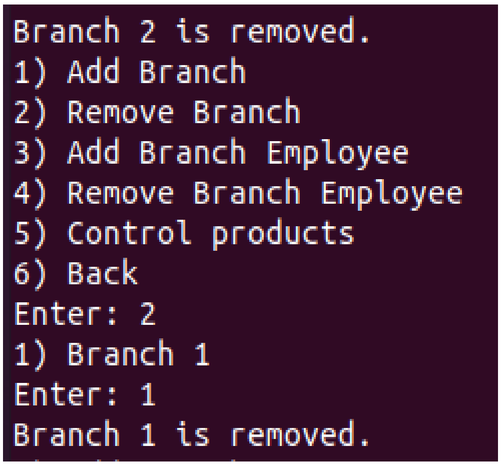
5) Compile / Login / Admin / Remove Branch / **try to remove while 0 branch**



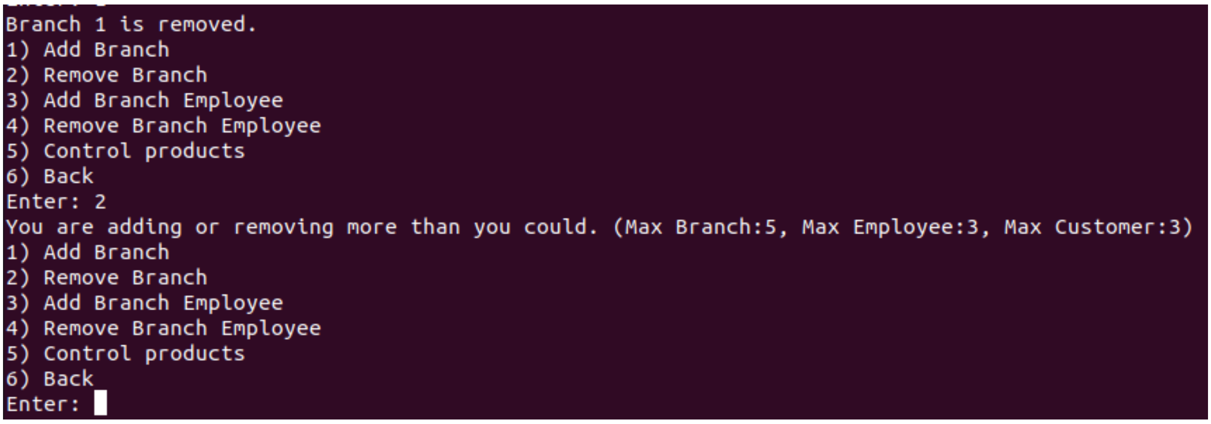
```
ail   : tugkan@admin.com
ssword: 1234
 Add Branch
 Remove Branch
 Add Branch Employee
 Remove Branch Employee
 Control products
 Back
ter: 2
 Branch 1
 Branch 2
 Branch 3
ter: 3
anch 3 is removed.
```



```
Branch 3 is removed.
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 2
1) Branch 1
2) Branch 2
Enter: 2
Branch 2 is removed.
```

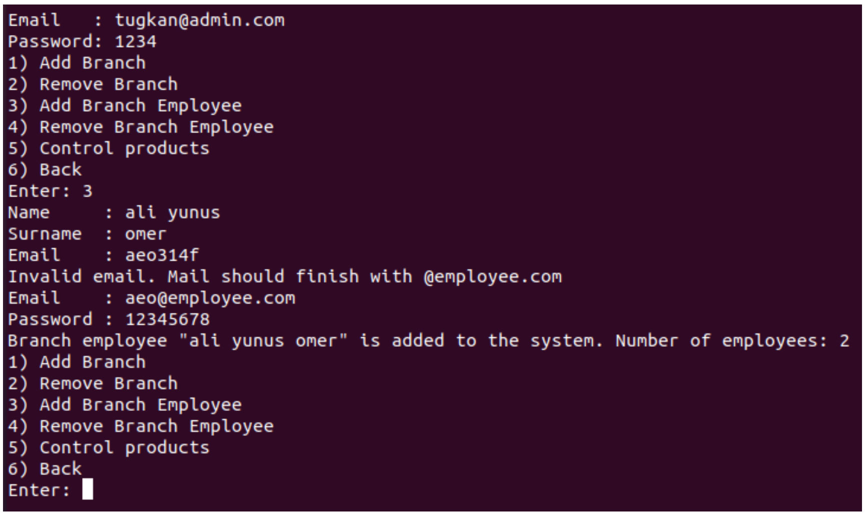


```
Branch 2 is removed.
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 2
1) Branch 1
Enter: 1
Branch 1 is removed.
```

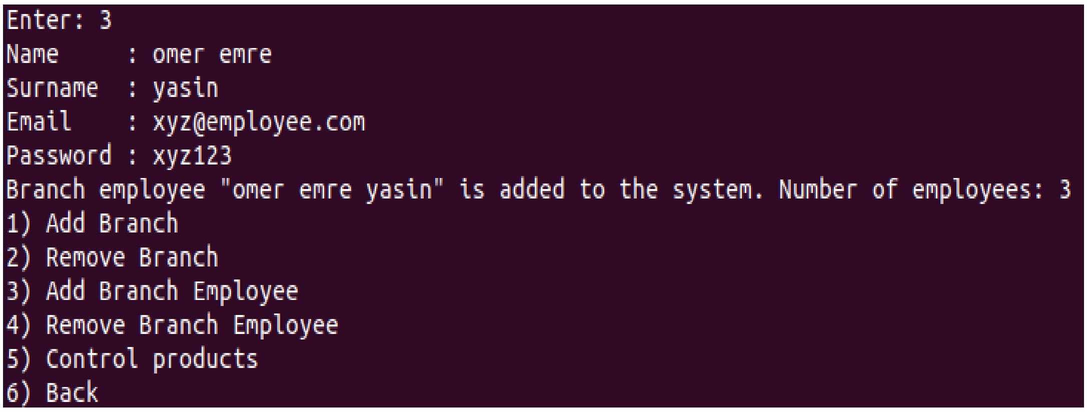


```
Branch 1 is removed.
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 2
You are adding or removing more than you could. (Max Branch:5, Max Employee:3, Max Customer:3)
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter:
```

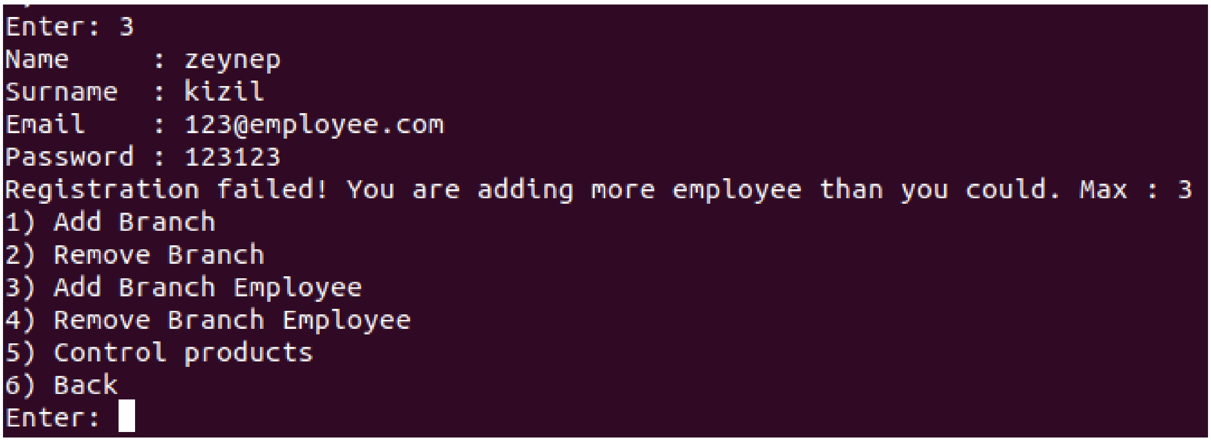
6) Compile / Login / Admin / Add Branch Employee / **try to exceed 3**



```
Email   : tugkan@admin.com
Password: 1234
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 3
Name     : ali yunus
Surname  : omer
Email    : aeo314f
Invalid email. Mail should finish with @employee.com
Email    : aeo@employee.com
Password : 12345678
Branch employee "ali yunus omer" is added to the system. Number of employees: 2
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 
```

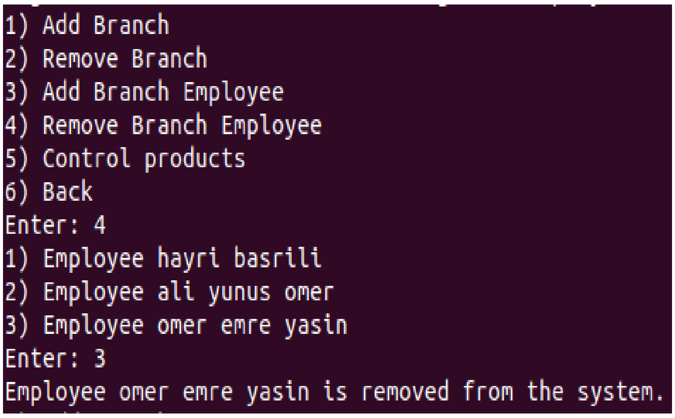


```
Enter: 3
Name     : omer emre
Surname  : yasin
Email    : xyz@employee.com
Password : xyz123
Branch employee "omer emre yasin" is added to the system. Number of employees: 3
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
```

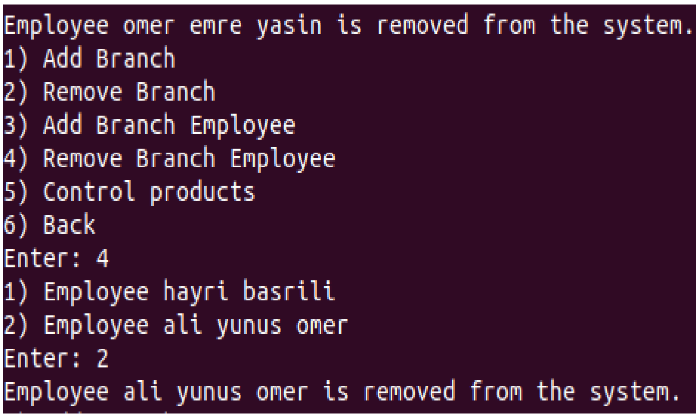


```
Enter: 3
Name     : zeynep
Surname  : kizil
Email    : 123@employee.com
Password : 123123
Registration failed! You are adding more employee than you could. Max : 3
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 
```

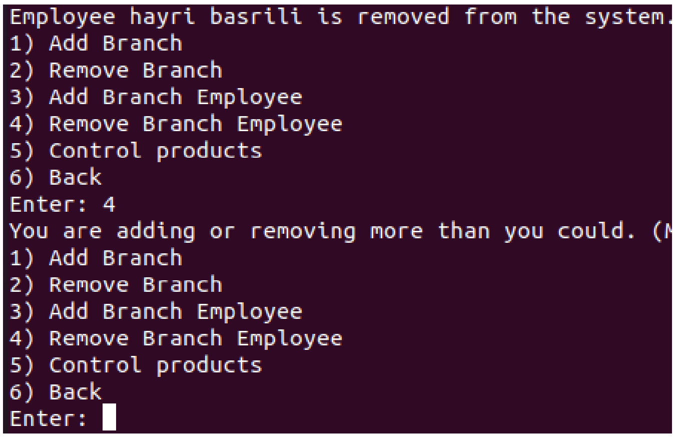
7) Compile / Login / Admin / Remove Branch / **try to remove while 0 employee**



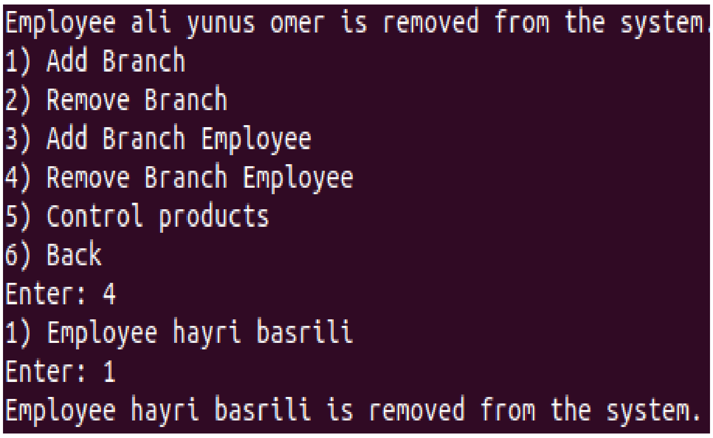
```
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 4
1) Employee hayri basrili
2) Employee ali yunus omer
3) Employee omer emre yasin
Enter: 3
Employee omer emre yasin is removed from the system.
```



```
Employee omer emre yasin is removed from the system.
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 4
1) Employee hayri basrili
2) Employee ali yunus omer
Enter: 2
Employee ali yunus omer is removed from the system.
```

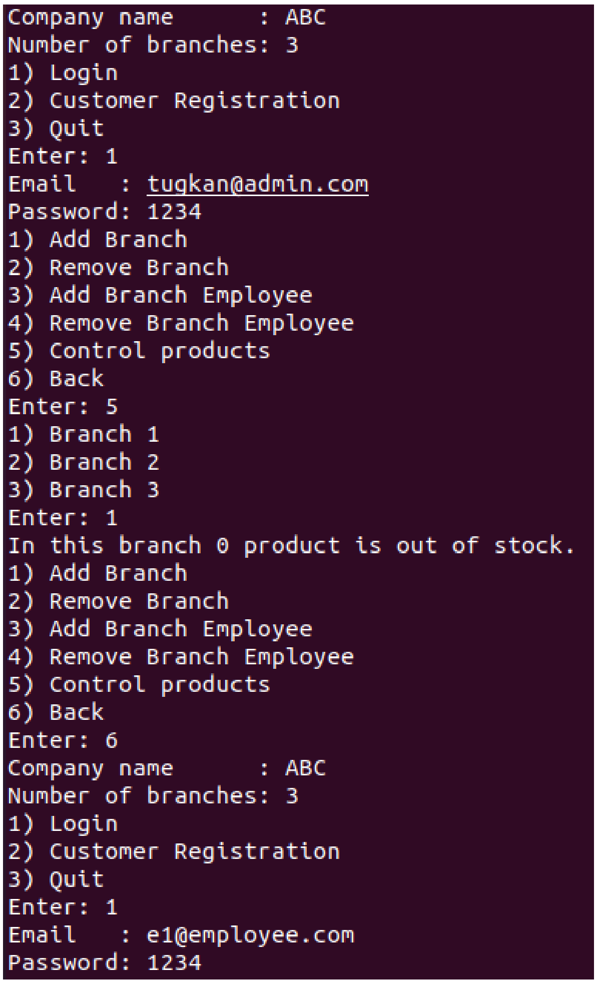


```
Employee hayri basrili is removed from the system.
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 4
You are adding or removing more than you could. (N
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 
```

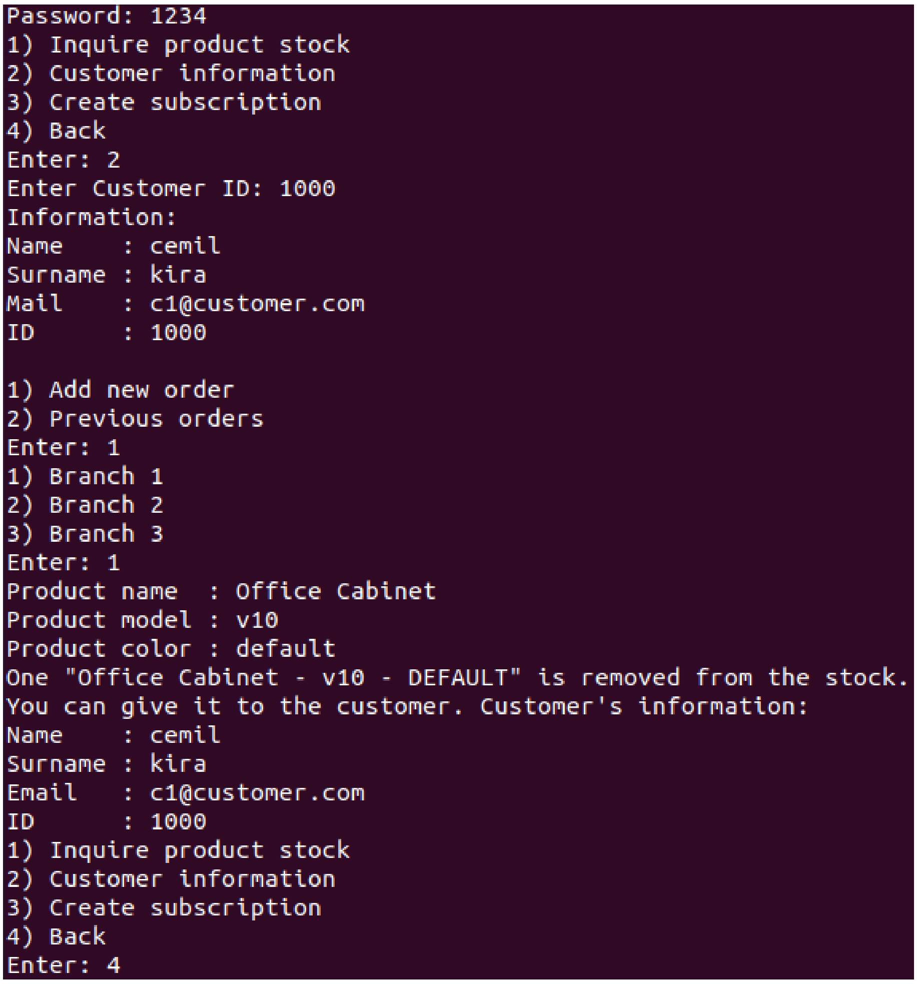


```
Employee ali yunus omer is removed from the system.
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 4
1) Employee hayri basrili
Enter: 1
Employee hayri basrili is removed from the system.
```

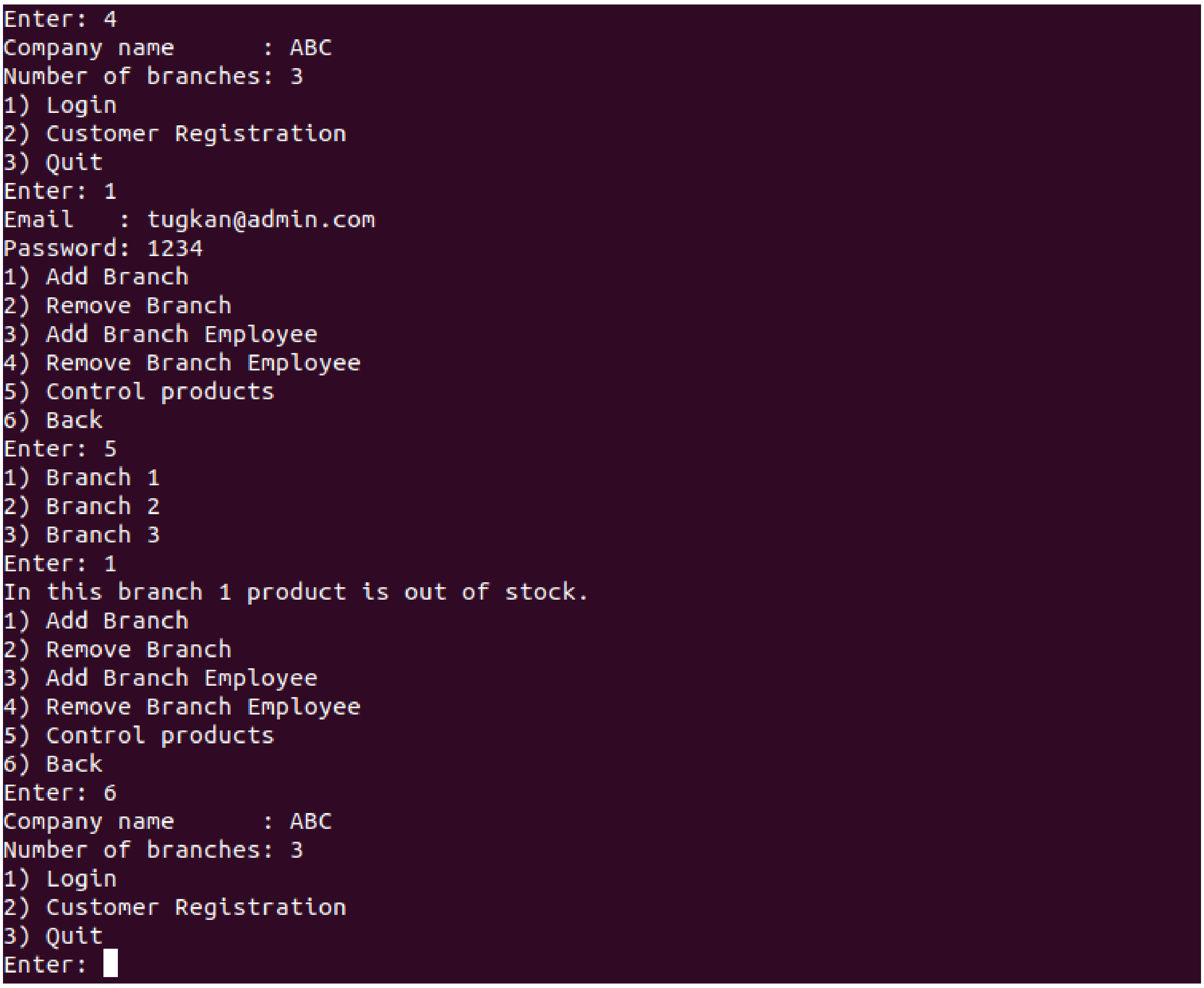
8) Compile / Login / Admin / Control Products / **first control stock in branch1 after that switch to employee and sell something in branch1. At last, login with admin and control stock in branch1 again**



```
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email   : tugkan@admin.com
Password: 1234
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 5
1) Branch 1
2) Branch 2
3) Branch 3
Enter: 1
In this branch 0 product is out of stock.
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 6
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email   : e1@employee.com
Password: 1234
```

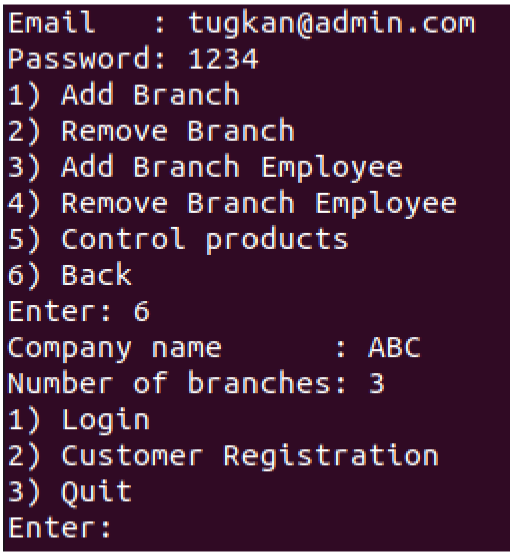


```
Password: 1234
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 2
Enter Customer ID: 1000
Information:
Name    : cemil
Surname : kira
Mail    : c1@customer.com
ID      : 1000

1) Add new order
2) Previous orders
Enter: 1
1) Branch 1
2) Branch 2
3) Branch 3
Enter: 1
Product name  : Office Cabinet
Product model : v10
Product color : default
One "Office Cabinet - v10 - DEFAULT" is removed from the stock.
You can give it to the customer. Customer's information:
Name    : cemil
Surname : kira
Email   : c1@customer.com
ID      : 1000
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 4
```



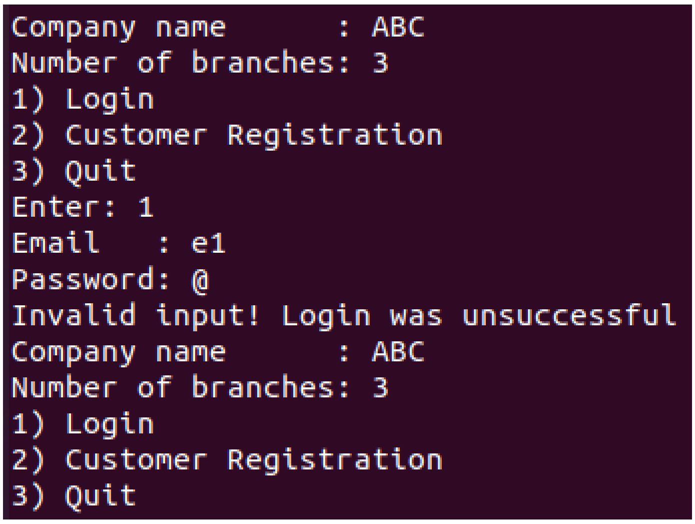
```
Enter: 4
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email   : tugkan@admin.com
Password: 1234
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 5
1) Branch 1
2) Branch 2
3) Branch 3
Enter: 1
In this branch 1 product is out of stock.
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 6
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 
```

9) Compile / Login / Admin / **Back**

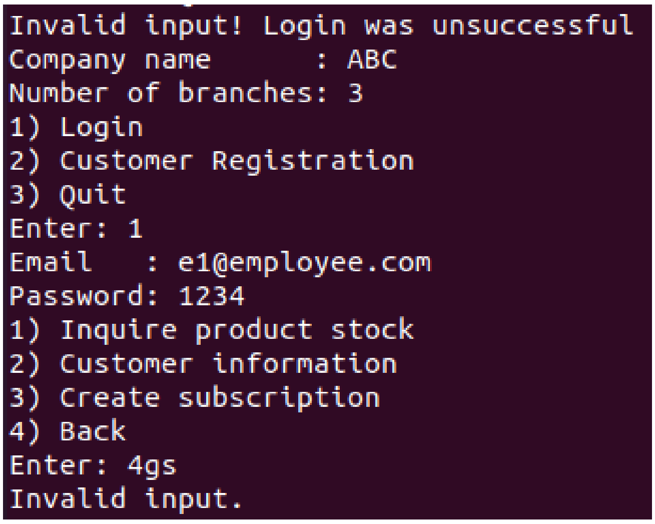


```
Email   : tugkan@admin.com
Password: 1234
1) Add Branch
2) Remove Branch
3) Add Branch Employee
4) Remove Branch Employee
5) Control products
6) Back
Enter: 6
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter:
```

10) Compile / Login / Employee / **give invalid input**

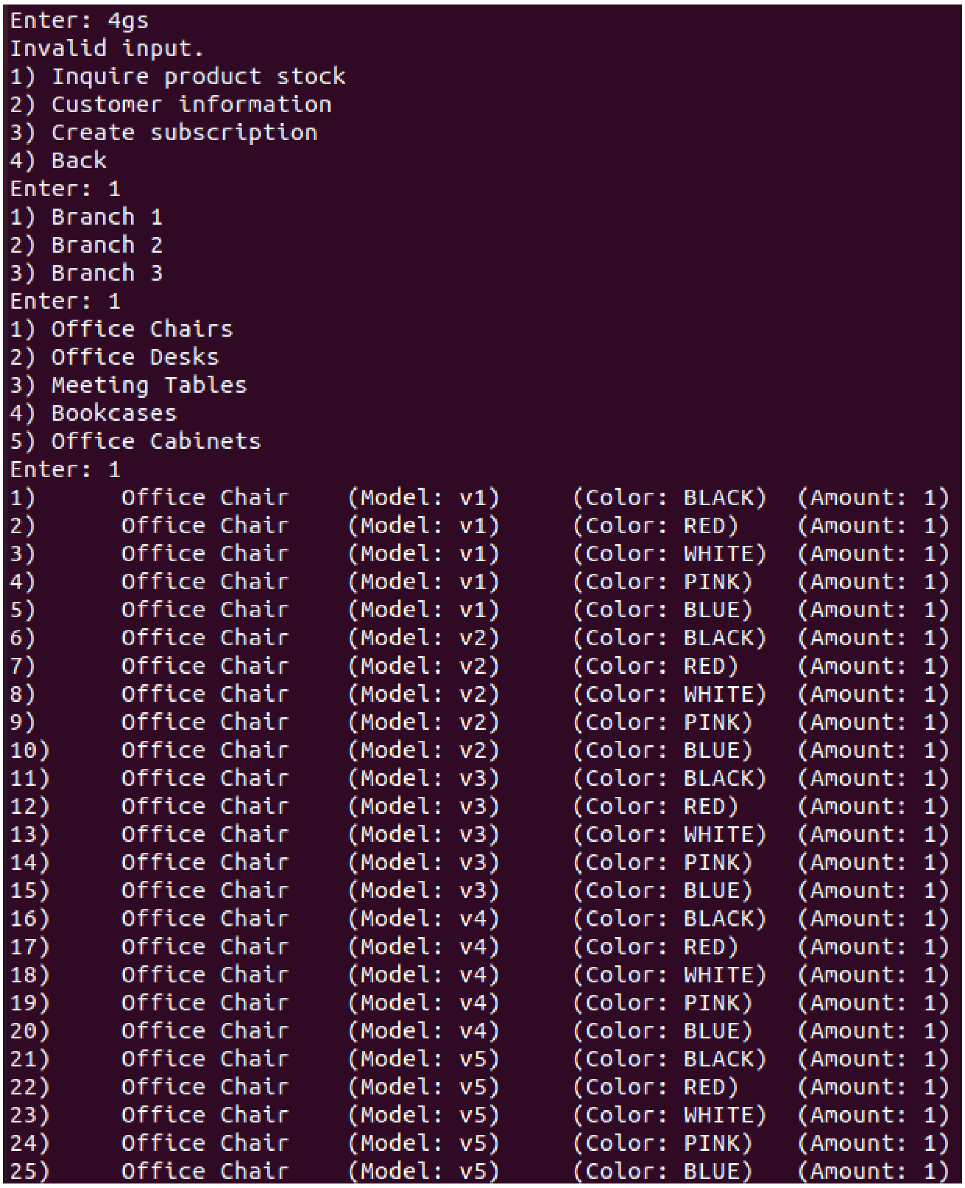


```
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email    : e1
Password: @
Invalid input! Login was unsuccessful
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
```

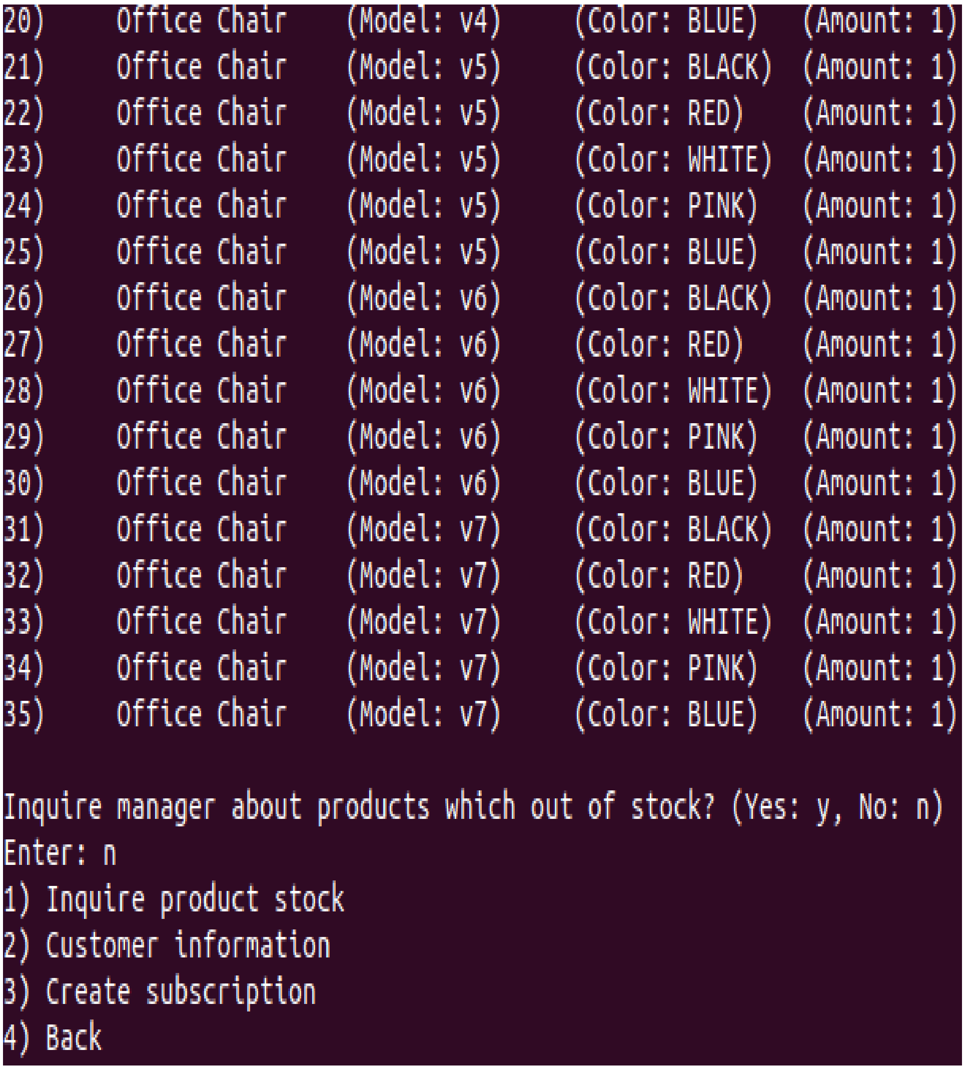


```
Invalid input! Login was unsuccessful
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email    : e1@employee.com
Password: 1234
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 4gs
Invalid input.
```

11) Compile / Login / Employee / Inquire Product / Branch1 / Office Desks / **don't inquire**

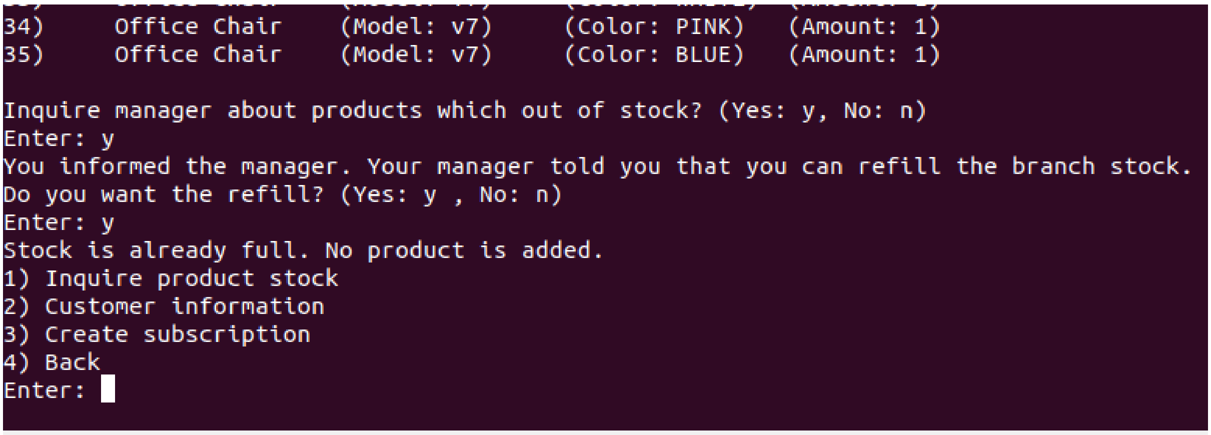


```
Enter: 4gs
Invalid input.
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 1
1) Branch 1
2) Branch 2
3) Branch 3
Enter: 1
1) Office Chairs
2) Office Desks
3) Meeting Tables
4) Bookcases
5) Office Cabinets
Enter: 1
1)      Office Chair    (Model: v1)     (Color: BLACK)  (Amount: 1)
2)      Office Chair    (Model: v1)     (Color: RED)    (Amount: 1)
3)      Office Chair    (Model: v1)     (Color: WHITE)  (Amount: 1)
4)      Office Chair    (Model: v1)     (Color: PINK)   (Amount: 1)
5)      Office Chair    (Model: v1)     (Color: BLUE)   (Amount: 1)
6)      Office Chair    (Model: v2)     (Color: BLACK)  (Amount: 1)
7)      Office Chair    (Model: v2)     (Color: RED)    (Amount: 1)
8)      Office Chair    (Model: v2)     (Color: WHITE)  (Amount: 1)
9)      Office Chair    (Model: v2)     (Color: PINK)   (Amount: 1)
10)     Office Chair    (Model: v2)     (Color: BLUE)   (Amount: 1)
11)     Office Chair    (Model: v3)     (Color: BLACK)  (Amount: 1)
12)     Office Chair    (Model: v3)     (Color: RED)    (Amount: 1)
13)     Office Chair    (Model: v3)     (Color: WHITE)  (Amount: 1)
14)     Office Chair    (Model: v3)     (Color: PINK)   (Amount: 1)
15)     Office Chair    (Model: v3)     (Color: BLUE)   (Amount: 1)
16)     Office Chair    (Model: v4)     (Color: BLACK)  (Amount: 1)
17)     Office Chair    (Model: v4)     (Color: RED)    (Amount: 1)
18)     Office Chair    (Model: v4)     (Color: WHITE)  (Amount: 1)
19)     Office Chair    (Model: v4)     (Color: PINK)   (Amount: 1)
20)     Office Chair    (Model: v4)     (Color: BLUE)   (Amount: 1)
21)     Office Chair    (Model: v5)     (Color: BLACK)  (Amount: 1)
22)     Office Chair    (Model: v5)     (Color: RED)    (Amount: 1)
23)     Office Chair    (Model: v5)     (Color: WHITE)  (Amount: 1)
24)     Office Chair    (Model: v5)     (Color: PINK)   (Amount: 1)
25)     Office Chair    (Model: v5)     (Color: BLUE)   (Amount: 1)
```



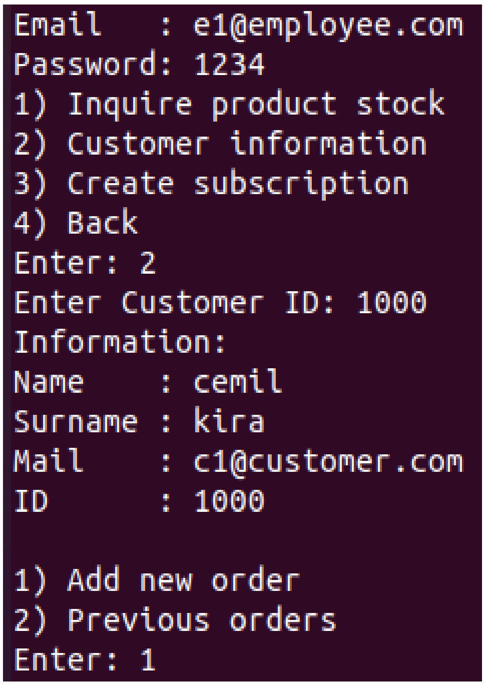
```
20)     Office Chair    (Model: v4)     (Color: BLUE)   (Amount: 1)
21)     Office Chair    (Model: v5)     (Color: BLACK)  (Amount: 1)
22)     Office Chair    (Model: v5)     (Color: RED)    (Amount: 1)
23)     Office Chair    (Model: v5)     (Color: WHITE)  (Amount: 1)
24)     Office Chair    (Model: v5)     (Color: PINK)   (Amount: 1)
25)     Office Chair    (Model: v5)     (Color: BLUE)   (Amount: 1)
26)     Office Chair    (Model: v6)     (Color: BLACK)  (Amount: 1)
27)     Office Chair    (Model: v6)     (Color: RED)    (Amount: 1)
28)     Office Chair    (Model: v6)     (Color: WHITE)  (Amount: 1)
29)     Office Chair    (Model: v6)     (Color: PINK)   (Amount: 1)
30)     Office Chair    (Model: v6)     (Color: BLUE)   (Amount: 1)
31)     Office Chair    (Model: v7)     (Color: BLACK)  (Amount: 1)
32)     Office Chair    (Model: v7)     (Color: RED)    (Amount: 1)
33)     Office Chair    (Model: v7)     (Color: WHITE)  (Amount: 1)
34)     Office Chair    (Model: v7)     (Color: PINK)   (Amount: 1)
35)     Office Chair    (Model: v7)     (Color: BLUE)   (Amount: 1)

Inquire manager about products which out of stock? (Yes: y, No: n)
Enter: n
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
```

12) Compile / Login / Employee / Inquire Product / Branch1 / Office Desks / **inquire / try to fill while stocks are full**

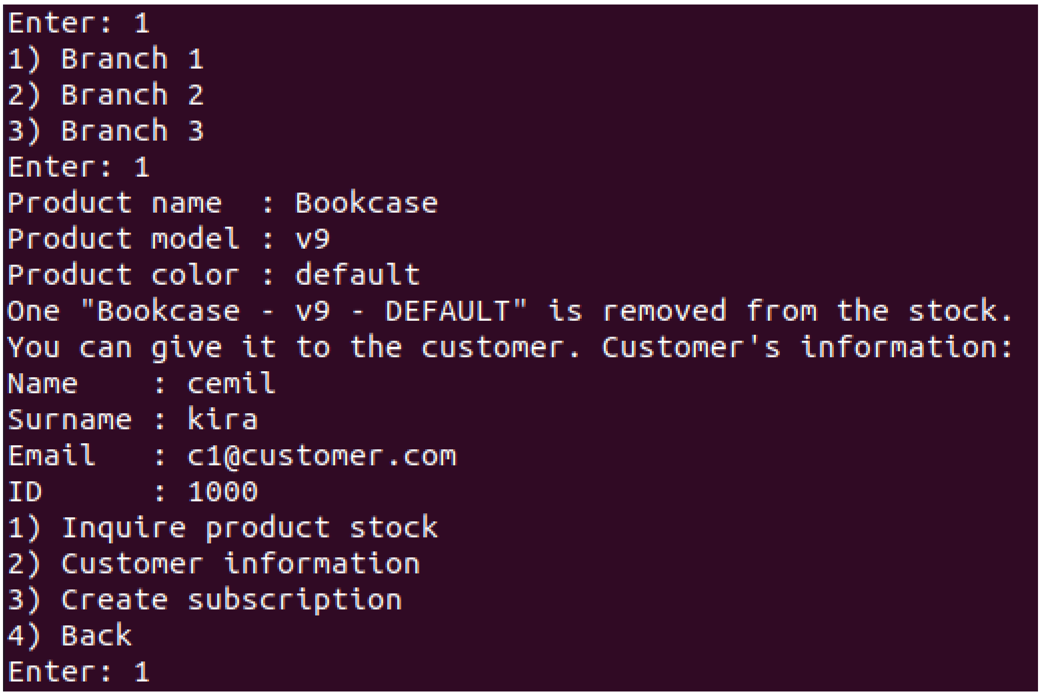


```
34)     Office Chair    (Model: v7)     (Color: PINK)   (Amount: 1)
35)     Office Chair    (Model: v7)     (Color: BLUE)   (Amount: 1)

Inquire manager about products which out of stock? (Yes: y, No: n)
Enter: y
You informed the manager. Your manager told you that you can refill the branch stock.
Do you want the refill? (Yes: y , No: n)
Enter: y
Stock is already full. No product is added.
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 
```

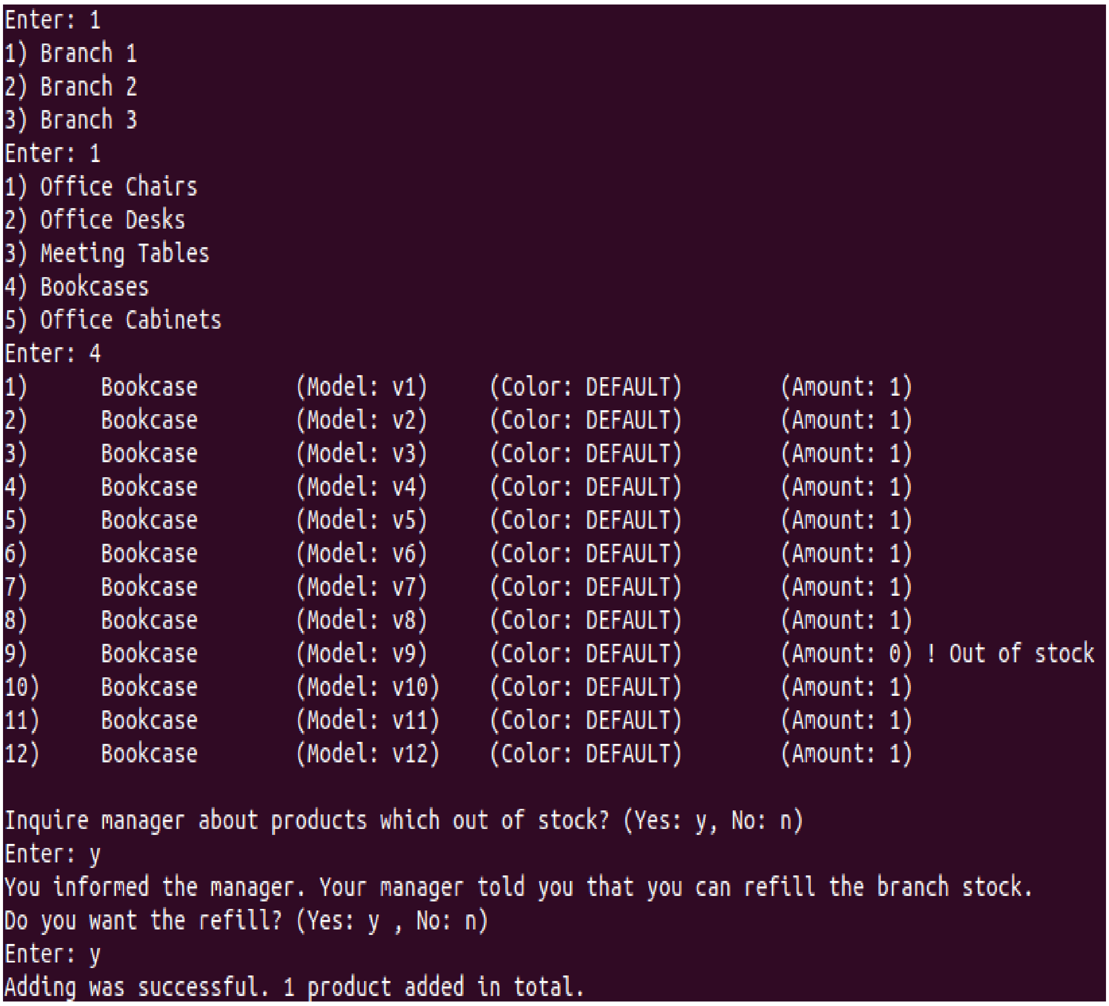
13) Compile / Login / Employee / **First Sell “Bookcase v9 default” to a customer then refill it**



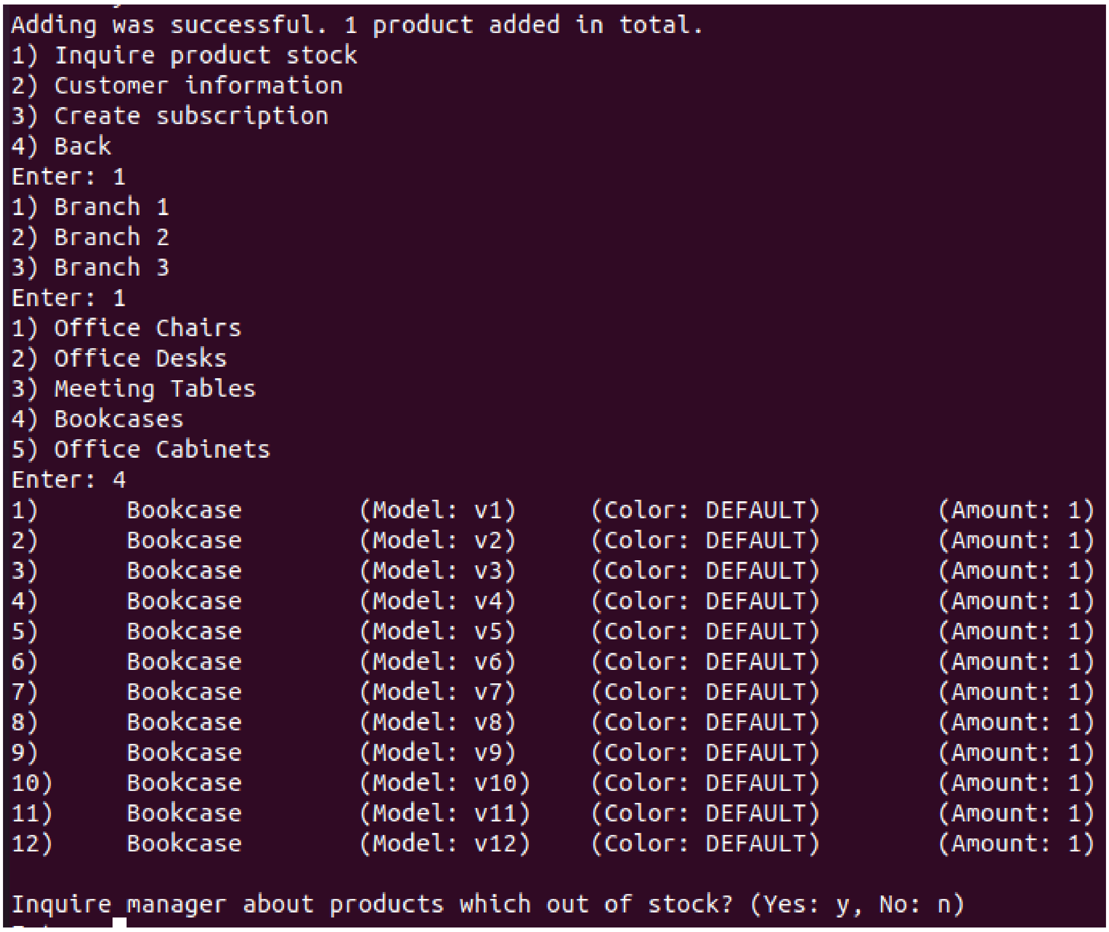
```
Email   : e1@employee.com
Password: 1234
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 2
Enter Customer ID: 1000
Information:
Name    : cemil
Surname : kira
Mail    : c1@customer.com
ID      : 1000

1) Add new order
2) Previous orders
Enter: 1
```



```
Enter: 1
1) Branch 1
2) Branch 2
3) Branch 3
Enter: 1
Product name  : Bookcase
Product model : v9
Product color : default
One "Bookcase - v9 - DEFAULT" is removed from the stock.
You can give it to the customer. Customer's information:
Name    : cemil
Surname : kira
Email   : c1@customer.com
ID      : 1000
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 1
```

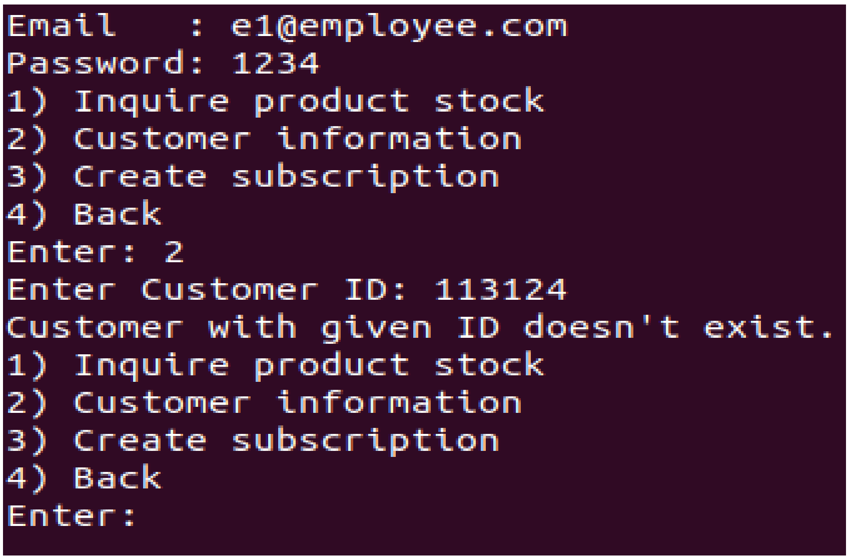


```
Enter: 1
1) Branch 1
2) Branch 2
3) Branch 3
Enter: 1
1) Office Chairs
2) Office Desks
3) Meeting Tables
4) Bookcases
5) Office Cabinets
Enter: 4
1)      Bookcase        (Model: v1)     (Color: DEFAULT)        (Amount: 1)
2)      Bookcase        (Model: v2)     (Color: DEFAULT)        (Amount: 1)
3)      Bookcase        (Model: v3)     (Color: DEFAULT)        (Amount: 1)
4)      Bookcase        (Model: v4)     (Color: DEFAULT)        (Amount: 1)
5)      Bookcase        (Model: v5)     (Color: DEFAULT)        (Amount: 1)
6)      Bookcase        (Model: v6)     (Color: DEFAULT)        (Amount: 1)
7)      Bookcase        (Model: v7)     (Color: DEFAULT)        (Amount: 1)
8)      Bookcase        (Model: v8)     (Color: DEFAULT)        (Amount: 1)
9)      Bookcase        (Model: v9)     (Color: DEFAULT)        (Amount: 0) ! Out of stock
10)     Bookcase        (Model: v10)    (Color: DEFAULT)        (Amount: 1)
11)     Bookcase        (Model: v11)    (Color: DEFAULT)        (Amount: 1)
12)     Bookcase        (Model: v12)    (Color: DEFAULT)        (Amount: 1)

Inquire manager about products which out of stock? (Yes: y, No: n)
Enter: y
You informed the manager. Your manager told you that you can refill the branch stock.
Do you want the refill? (Yes: y , No: n)
Enter: y
Adding was successful. 1 product added in total.
```



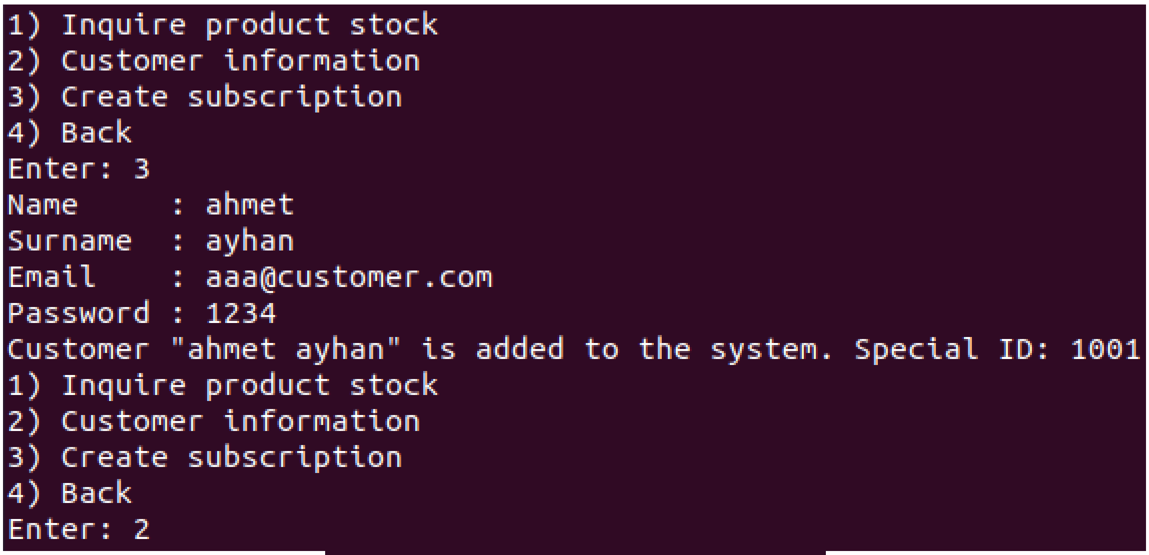
```
Adding was successful. 1 product added in total.
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 1
1) Branch 1
2) Branch 2
3) Branch 3
Enter: 1
1) Office Chairs
2) Office Desks
3) Meeting Tables
4) Bookcases
5) Office Cabinets
Enter: 4
1)      Bookcase        (Model: v1)     (Color: DEFAULT)        (Amount: 1)
2)      Bookcase        (Model: v2)     (Color: DEFAULT)        (Amount: 1)
3)      Bookcase        (Model: v3)     (Color: DEFAULT)        (Amount: 1)
4)      Bookcase        (Model: v4)     (Color: DEFAULT)        (Amount: 1)
5)      Bookcase        (Model: v5)     (Color: DEFAULT)        (Amount: 1)
6)      Bookcase        (Model: v6)     (Color: DEFAULT)        (Amount: 1)
7)      Bookcase        (Model: v7)     (Color: DEFAULT)        (Amount: 1)
8)      Bookcase        (Model: v8)     (Color: DEFAULT)        (Amount: 1)
9)      Bookcase        (Model: v9)     (Color: DEFAULT)        (Amount: 1)
10)     Bookcase        (Model: v10)    (Color: DEFAULT)        (Amount: 1)
11)     Bookcase        (Model: v11)    (Color: DEFAULT)        (Amount: 1)
12)     Bookcase        (Model: v12)    (Color: DEFAULT)        (Amount: 1)

Inquire manager about products which out of stock? (Yes: y, No: n)
```

14) Compile / Login / Employee / Customer Information / **give invalid ID**

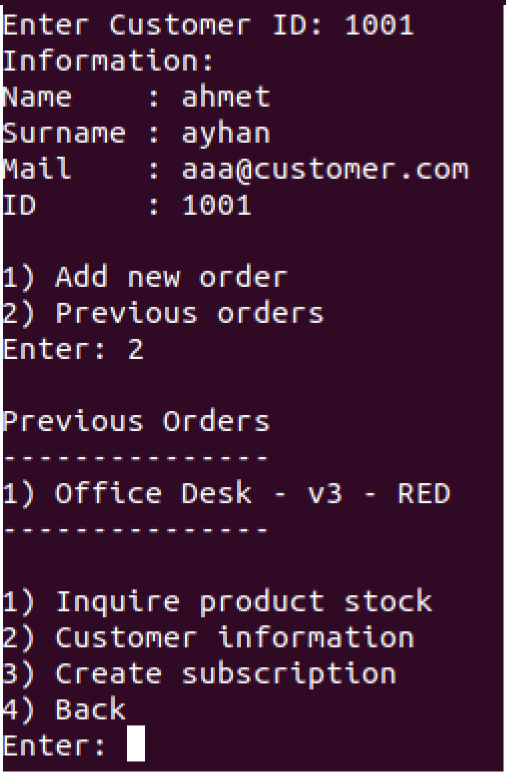


```
Email    : e1@employee.com
Password: 1234
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 2
Enter Customer ID: 113124
Customer with given ID doesn't exist.
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter:
```

15) Compile / Login / Employee / **create a customer subscription and sell something to it.Then look his/her previous orders**

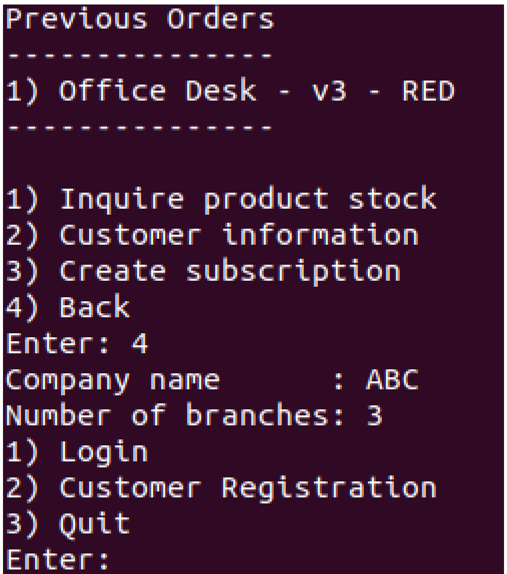


```
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 3
Name     : ahmet
Surname  : ayhan
Email    : aaa@customer.com
Password : 1234
Customer "ahmet ayhan" is added to the system. Special ID: 1001
1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 2
```



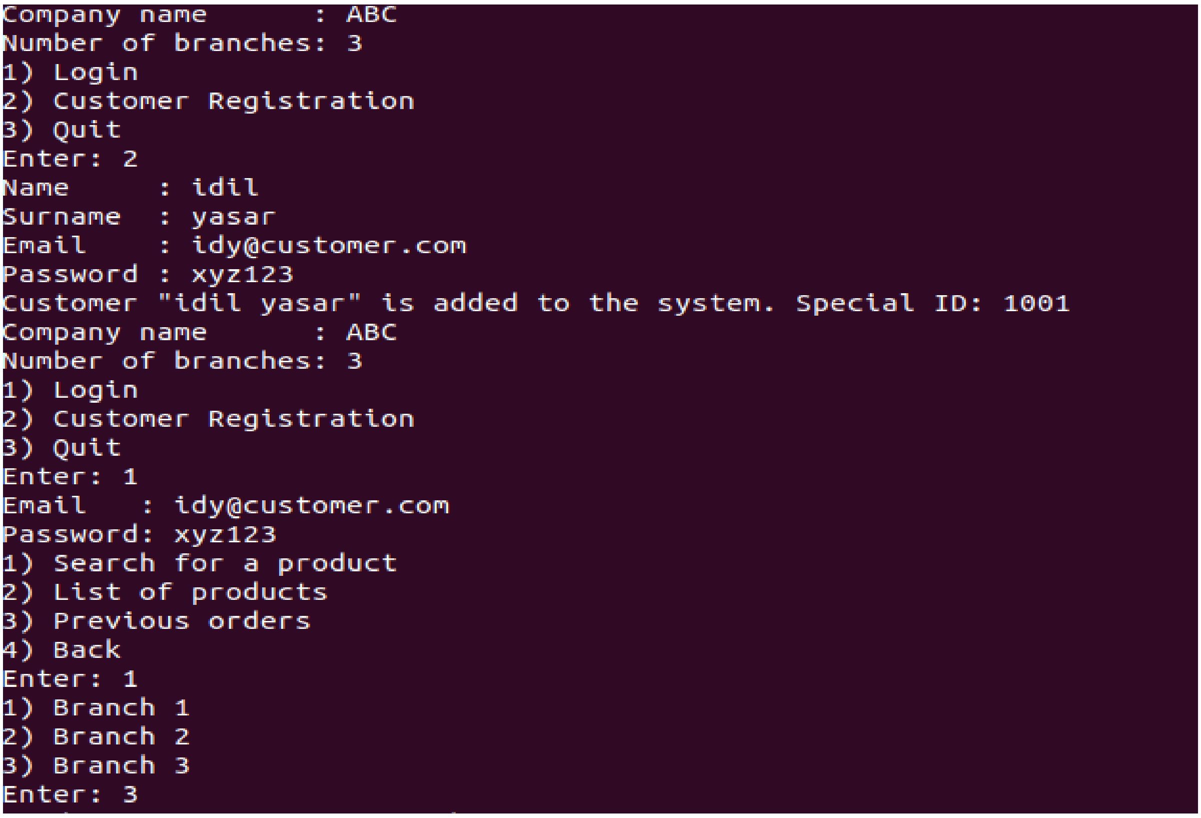
```
Enter Customer ID: 1001
Information:
Name    : ahmet
Surname : ayhan
Mail    : aaa@customer.com
ID      : 1001

1) Add new order
2) Previous orders
Enter: 2

Previous Orders
---------------
1) Office Desk - v3 - RED
---------------

1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 
```

16) Compile / Login / Employee / **Back**

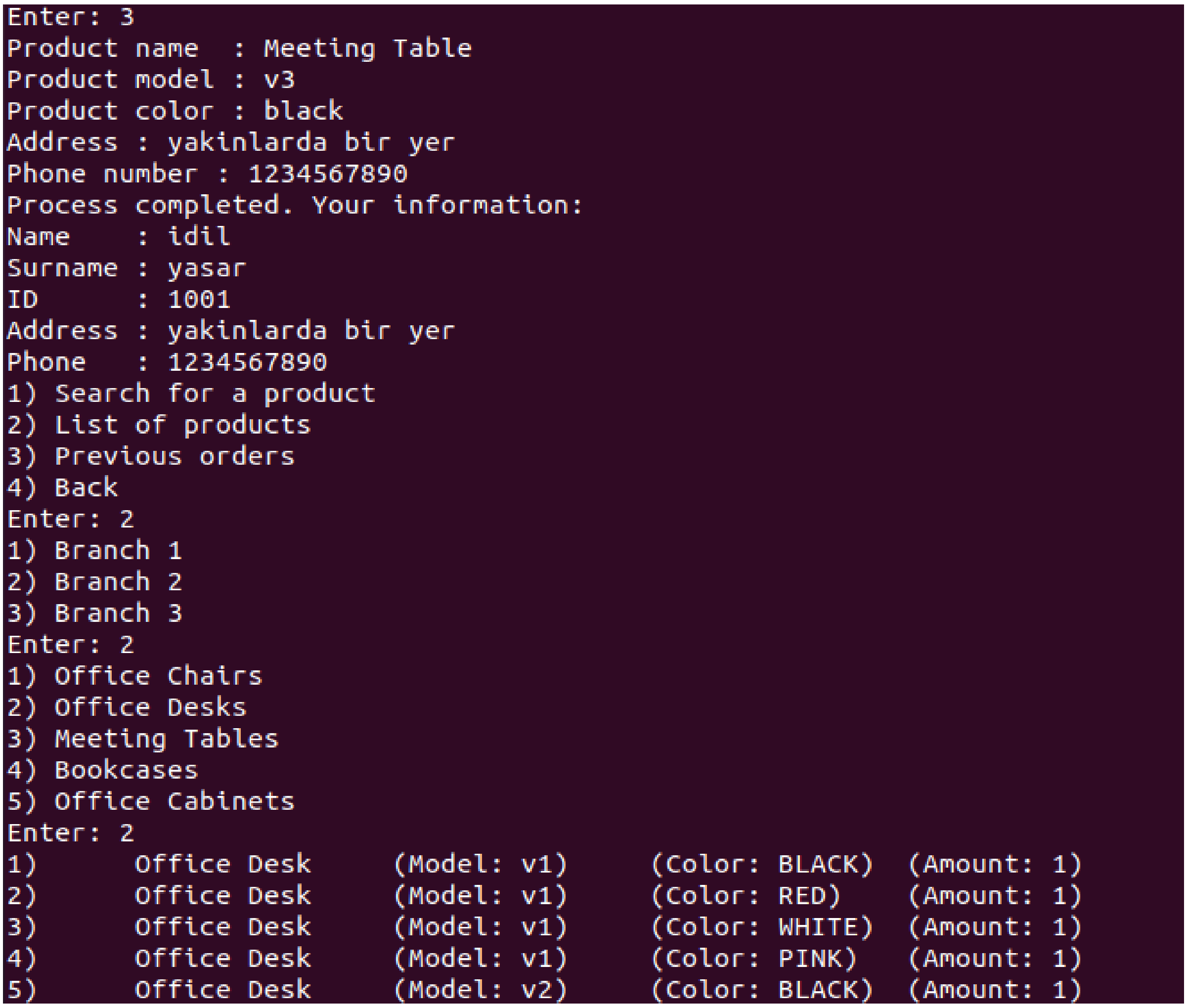


```
Previous Orders
---------------
1) Office Desk - v3 - RED
---------------

1) Inquire product stock
2) Customer information
3) Create subscription
4) Back
Enter: 4
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter:
```

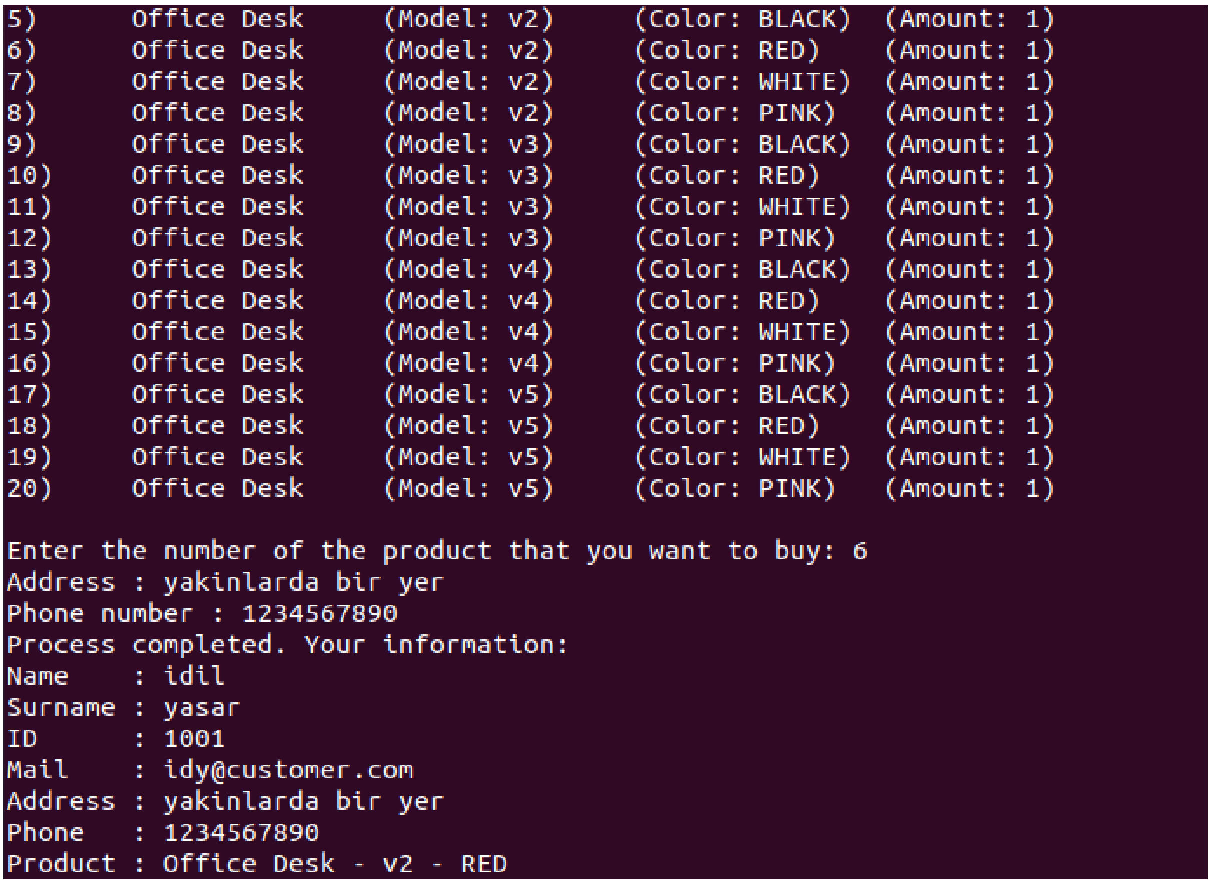
17) Compile / Register / **Login with new registered customer / give invalid input / Search for “Meeting Table v3 black” and buy it. / List “Branch 2- Office Desk” and buy v2 red / Look to your previous orders / Back**



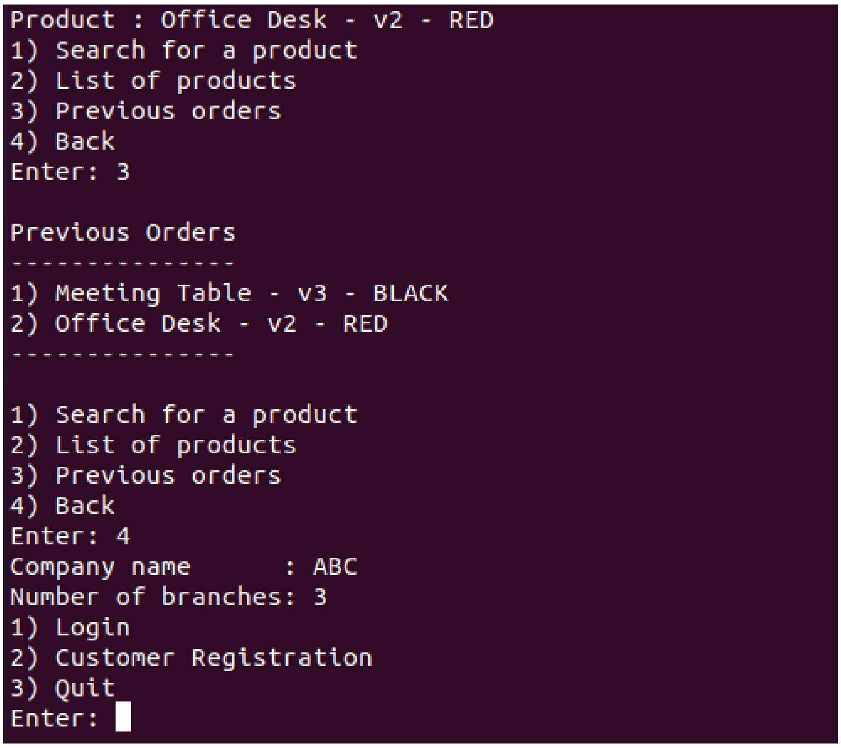
```
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 2
Name      : idil
Surname   : yasar
Email     : idy@customer.com
Password : xyz123
Customer "idil yasar" is added to the system. Special ID: 1001
Company name       : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 1
Email    : idy@customer.com
Password: xyz123
1) Search for a product
2) List of products
3) Previous orders
4) Back
Enter: 1
1) Branch 1
2) Branch 2
3) Branch 3
Enter: 3
```



```
Enter: 3
Product name  : Meeting Table
Product model : v3
Product color : black
Address : yakinlarda bir yer
Phone number : 1234567890
Process completed. Your information:
Name     : idil
Surname : yasar
ID       : 1001
Address : yakinlarda bir yer
Phone    : 1234567890
1) Search for a product
2) List of products
3) Previous orders
4) Back
Enter: 2
1) Branch 1
2) Branch 2
3) Branch 3
Enter: 2
1) Office Chairs
2) Office Desks
3) Meeting Tables
4) Bookcases
5) Office Cabinets
Enter: 2
1)      Office Desk     (Model: v1)     (Color: BLACK)  (Amount: 1)
2)      Office Desk     (Model: v1)     (Color: RED)    (Amount: 1)
3)      Office Desk     (Model: v1)     (Color: WHITE)  (Amount: 1)
4)      Office Desk     (Model: v1)     (Color: PINK)   (Amount: 1)
5)      Office Desk     (Model: v2)     (Color: BLACK)  (Amount: 1)
```

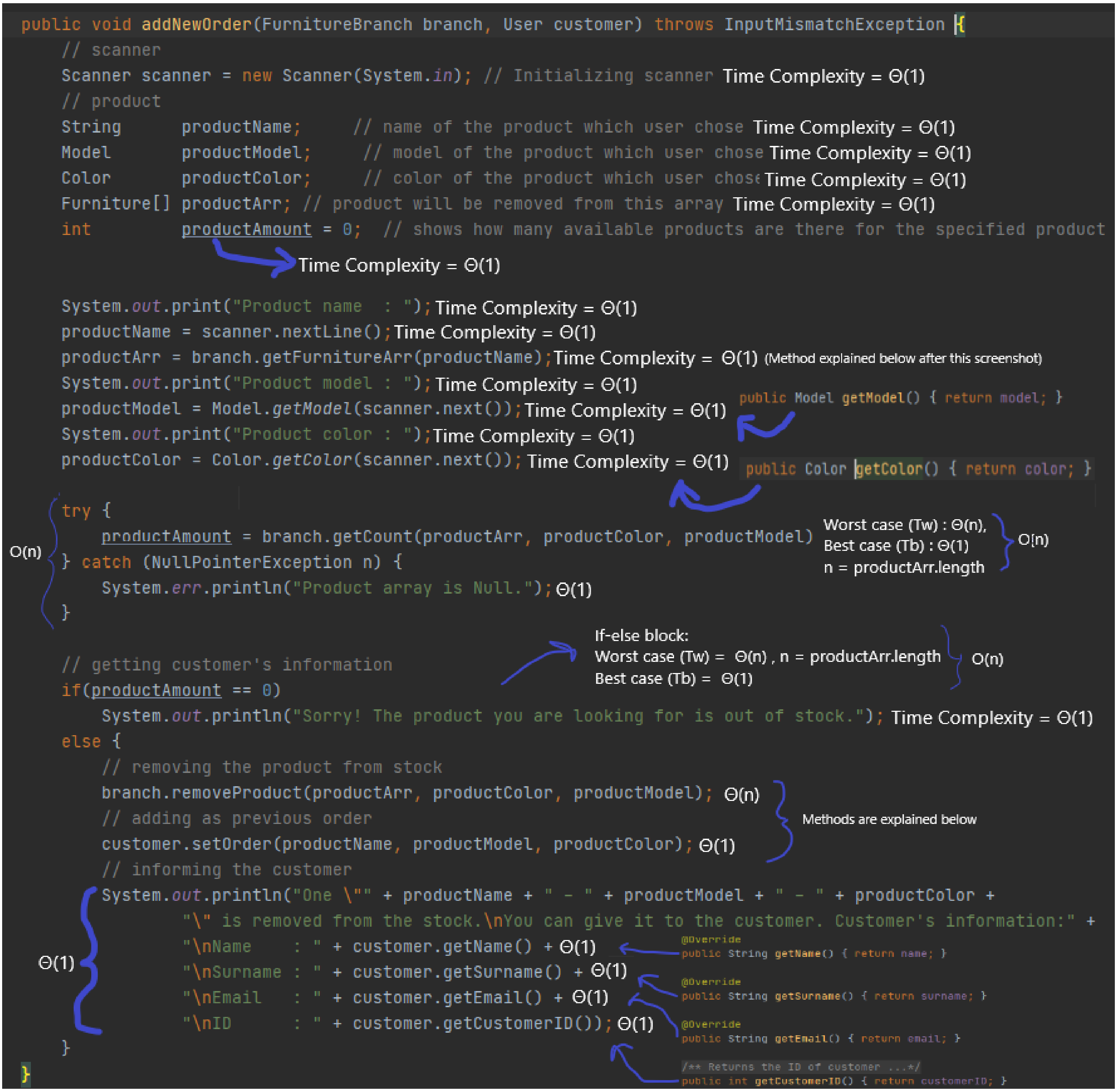


```
5)      Office Desk     (Model: v2)     (Color: BLACK)  (Amount: 1)
6)      Office Desk     (Model: v2)     (Color: RED)    (Amount: 1)
7)      Office Desk     (Model: v2)     (Color: WHITE)  (Amount: 1)
8)      Office Desk     (Model: v2)     (Color: PINK)   (Amount: 1)
9)      Office Desk     (Model: v3)     (Color: BLACK)  (Amount: 1)
10)     Office Desk     (Model: v3)     (Color: RED)    (Amount: 1)
11)     Office Desk     (Model: v3)     (Color: WHITE)  (Amount: 1)
12)     Office Desk     (Model: v3)     (Color: PINK)   (Amount: 1)
13)     Office Desk     (Model: v4)     (Color: BLACK)  (Amount: 1)
14)     Office Desk     (Model: v4)     (Color: RED)    (Amount: 1)
15)     Office Desk     (Model: v4)     (Color: WHITE)  (Amount: 1)
16)     Office Desk     (Model: v4)     (Color: PINK)   (Amount: 1)
17)     Office Desk     (Model: v5)     (Color: BLACK)  (Amount: 1)
18)     Office Desk     (Model: v5)     (Color: RED)    (Amount: 1)
19)     Office Desk     (Model: v5)     (Color: WHITE)  (Amount: 1)
20)     Office Desk     (Model: v5)     (Color: PINK)   (Amount: 1)

Enter the number of the product that you want to buy: 6
Address : yakinlarda bir yer
Phone number : 1234567890
Process completed. Your information:
Name    : idil
Surname : yasar
ID      : 1001
Mail    : idy@customer.com
Address : yakinlarda bir yer
Phone   : 1234567890
Product : Office Desk - v2 - RED
```



```
Product : Office Desk - v2 - RED
1) Search for a product
2) List of products
3) Previous orders
4) Back
Enter: 3

Previous Orders
---------------
1) Meeting Table - v3 - BLACK
2) Office Desk - v2 - RED
---------------

1) Search for a product
2) List of products
3) Previous orders
4) Back
Enter: 4
Company name      : ABC
Number of branches: 3
1) Login
2) Customer Registration
3) Quit
Enter: 
```

#########
# PART 2 #
#########



```
public void addNewOrder(FurnitureBranch branch, User customer) throws InputMismatchException {
    // scanner
    Scanner scanner = new Scanner(System.in); // Initializing scanner Time Complexity = Θ(1)
    // product
    String      productName;     // name of the product which user chose Time Complexity = Θ(1)
    Model       productModel;     // model of the product which user chose Time Complexity = Θ(1)
    Color       productColor;     // color of the product which user chos Time Complexity = Θ(1)
    Furniture[] productArr; // product will be removed from this array Time Complexity = Θ(1)
    int         productAmount = 0;  // shows how many available products are there for the specified product
                                Time Complexity = Θ(1)

    System.out.print("Product name  : ");Time Complexity = Θ(1)
    productName = scanner.nextLine();Time Complexity = Θ(1)
    productArr = branch.getFurnitureArr(productName);Time Complexity =  Θ(1) (Method explained below after this screenshot)
    System.out.print("Product model : ");Time Complexity = Θ(1)
    productModel = Model.getModel(scanner.next());Time Complexity = Θ(1)
    System.out.print("Product color : ");Time Complexity = Θ(1)
    productColor = Color.getColor(scanner.next()); Time Complexity = Θ(1)

    try {
        productAmount = branch.getCount(productArr, productColor, productModel)
    } catch (NullPointerException n) {
        System.err.println("Product array is Null.");Θ(1)
    }

    // getting customer's information
    if(productAmount == 0)
        System.out.println("Sorry! The product you are looking for is out of stock."); Time Complexity = Θ(1)
    else {
        // removing the product from stock
        branch.removeProduct(productArr, productColor, productModel); Θ(n)
        // adding as previous order
        customer.setOrder(productName, productModel, productColor); Θ(1)
        // informing the customer
        System.out.println("One \"" + productName + " - " + productModel + " - " + productColor +
                "\" is removed from the stock.\nYou can give it to the customer. Customer's information:" +
                "\nName    : " + customer.getName() + Θ(1)
                "\nSurname : " + customer.getSurname() + Θ(1)
                "\nEmail   : " + customer.getEmail() + Θ(1)
                "\nID      : " + customer.getCustomerID()); Θ(1)
    }
}
```

```
public Model getModel() { return model; }
```

```
public Color getColor() { return color; }
```

O(n)

Worst case (Tw) : Θ(n),
Best case (Tb) : Θ(1)
n = productArr.length

O(n)

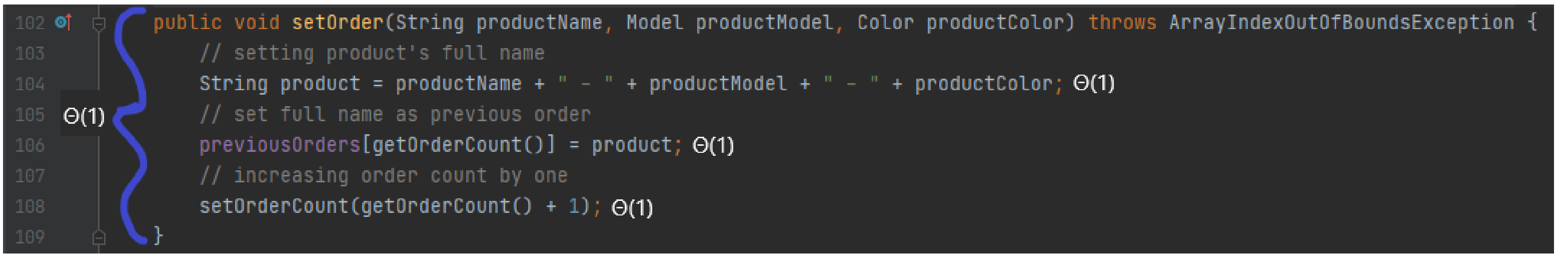
If-else block:
Worst case (Tw) = Θ(n) , n = productArr.length
Best case (Tb) = Θ(1)

O(n)

Methods are explained below

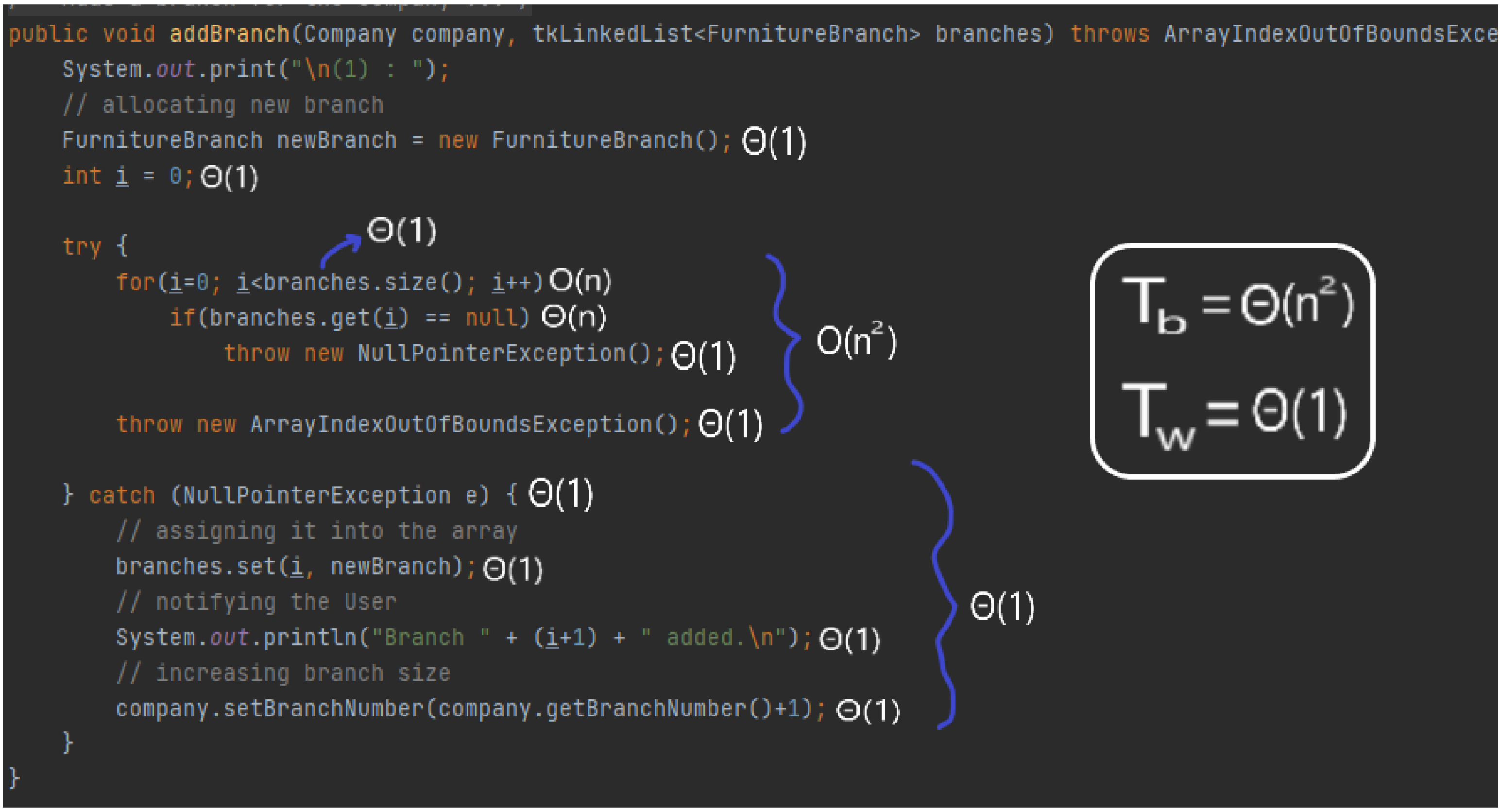
Θ(1)

```
@Override
public String getName() { return name; }

@Override
public String getSurname() { return surname; }

@Override
public String getEmail() { return email; }

/** Returns the ID of customer ...*/
public int getCustomerID() { return customerID; }
```



```
public void setOrder(String productName, Model productModel, Color productColor) throws ArrayIndexOutOfBoundsException {
    // setting product's full name
    String product = productName + " - " + productModel + " - " + productColor; Θ(1)
    // set full name as previous order
    previousOrders[getOrderCount()] = product; Θ(1)
    // increasing order count by one
    setOrderCount(getOrderCount() + 1); Θ(1)
}
```

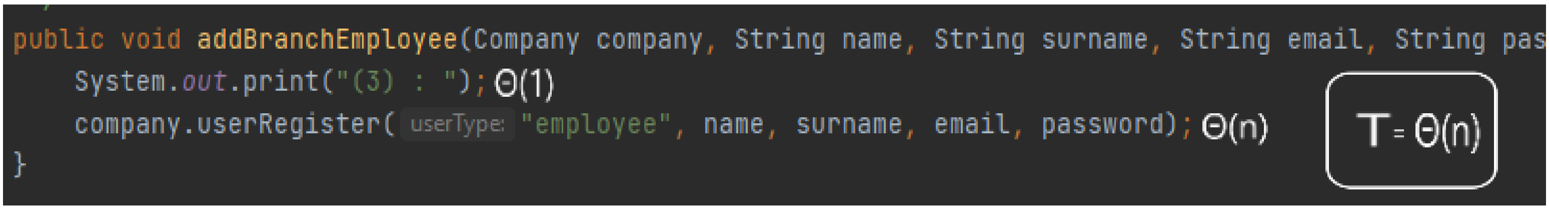
Θ(1)



```
public void addBranch(Company company, tkLinkedList<FurnitureBranch> branches) throws ArrayIndexOutOfBoundsExce
    System.out.print("\n(1) : ");
    // allocating new branch
    FurnitureBranch newBranch = new FurnitureBranch(); Θ(1)
    int i = 0; Θ(1)
                             Θ(1)
    try {
        for(i=0; i<branches.size(); i++) O(n)
            if(branches.get(i) == null) Θ(n)
                throw new NullPointerException(); Θ(1)       O(n²)

        throw new ArrayIndexOutOfBoundsException(); Θ(1)

    } catch (NullPointerException e) { Θ(1)
        // assigning it into the array
        branches.set(i, newBranch); Θ(1)
        // notifying the User                                          Θ(1)
        System.out.println("Branch " + (i+1) + " added.\n"); Θ(1)
        // increasing branch size
        company.setBranchNumber(company.getBranchNumber()+1); Θ(1)
    }
}
```

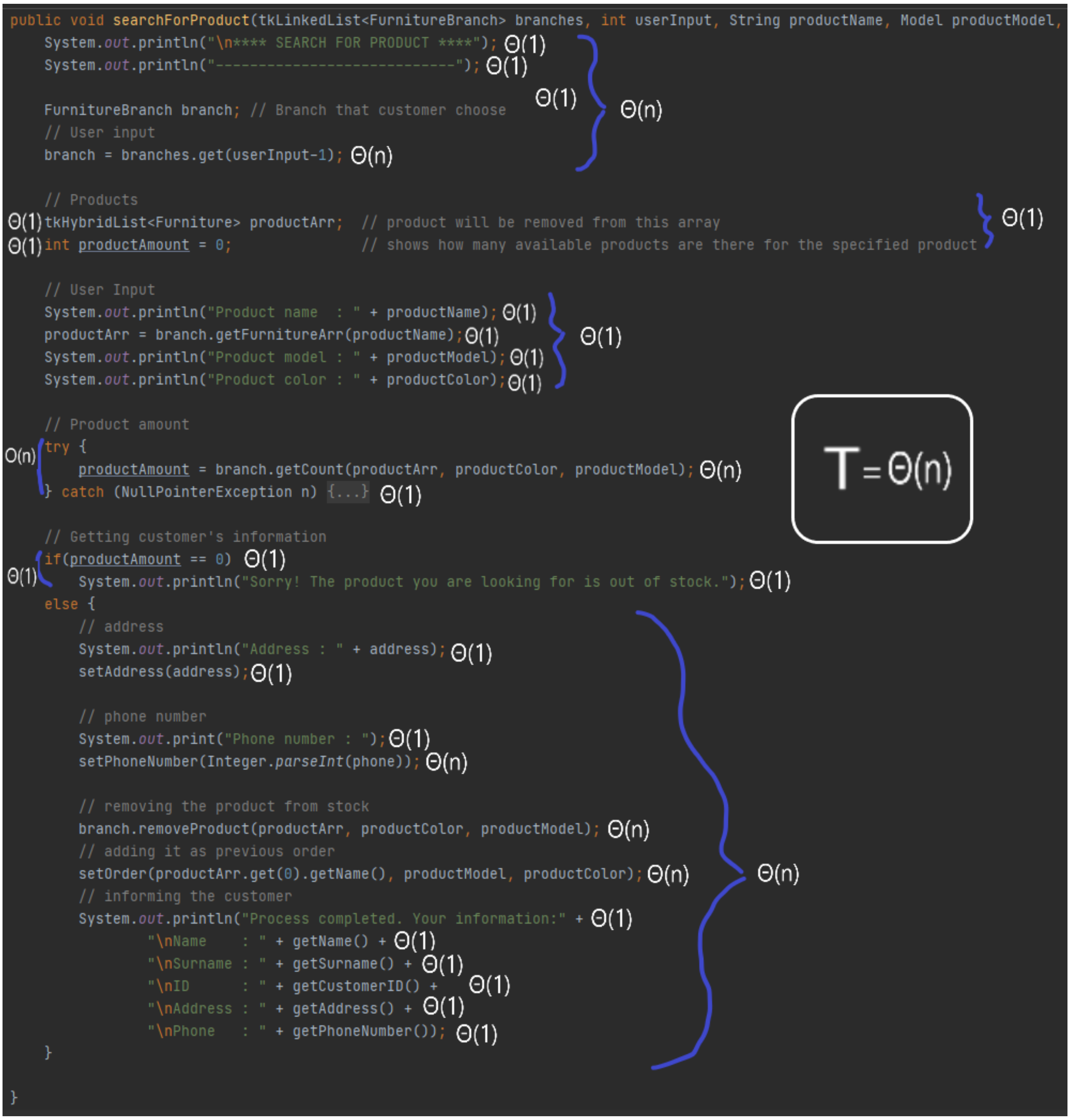
$T_b = \Theta(n^2)$

$T_w = \Theta(1)$



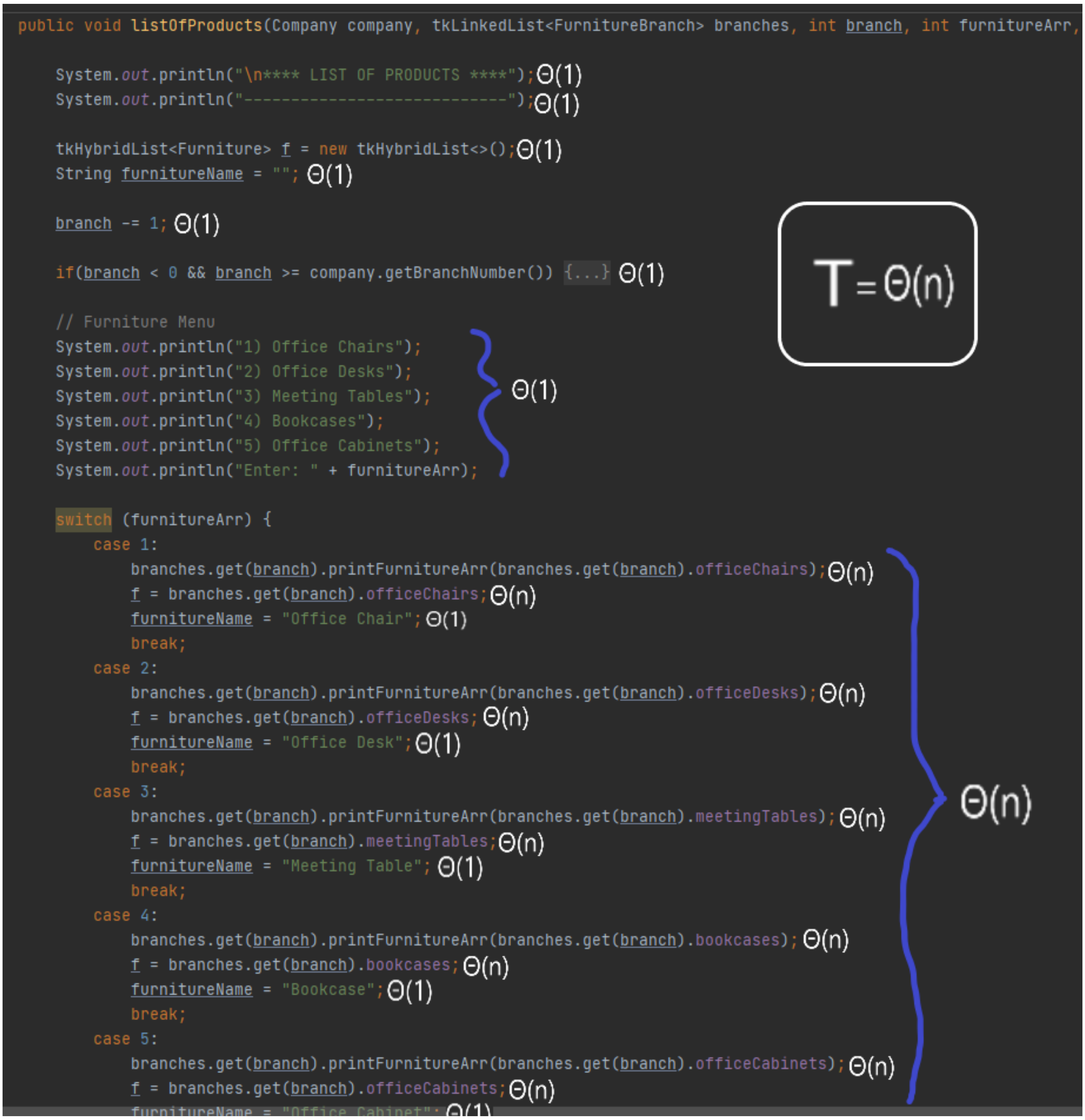
```
public void addBranchEmployee(Company company, String name, String surname, String email, String pas
    System.out.print("(3) : "); Θ(1)
    company.userRegister( userType: "employee", name, surname, email, password); Θ(n)
}
```

$T = \Theta(n)$



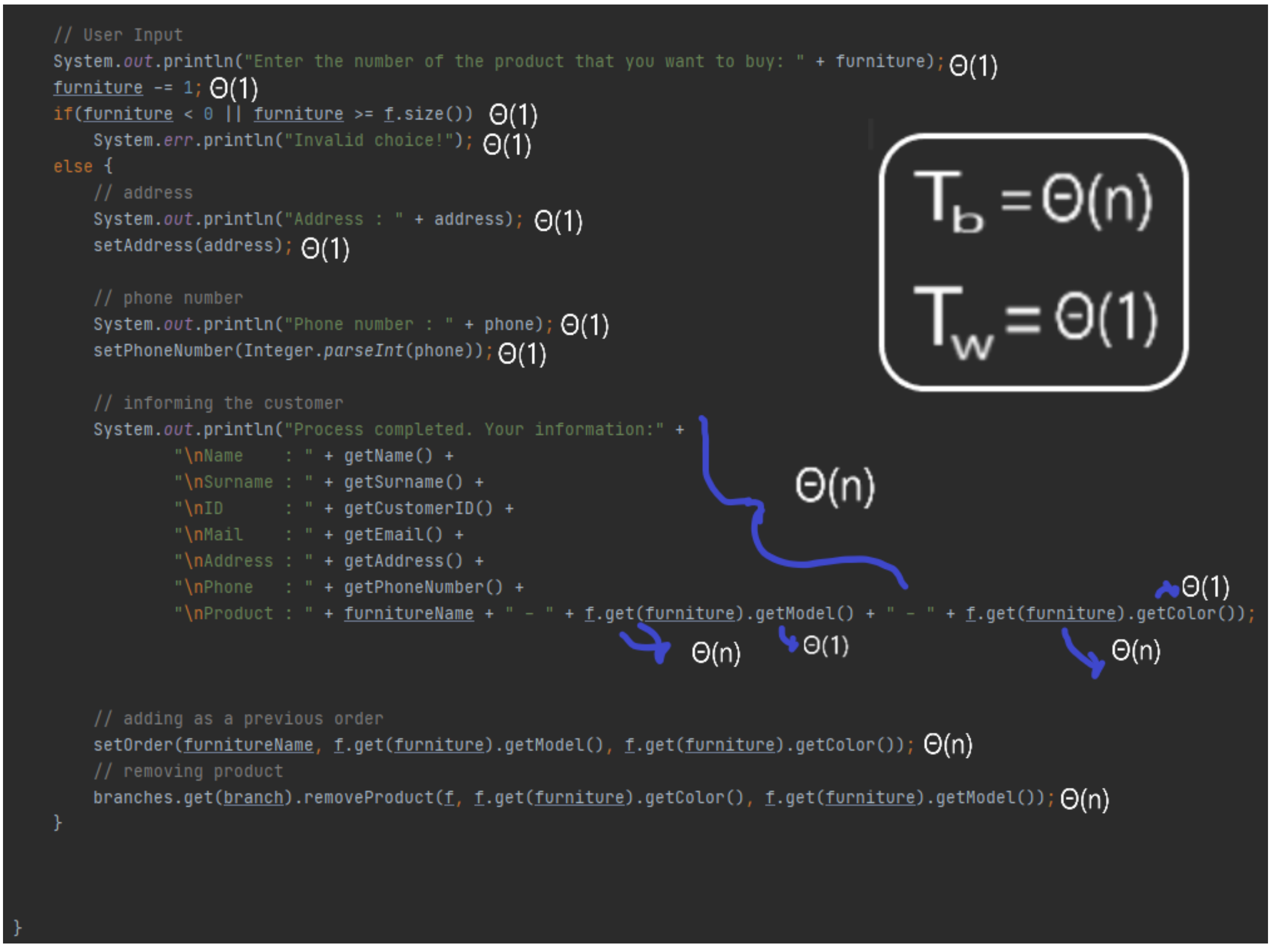
```
public void searchForProduct(tkLinkedList<FurnitureBranch> branches, int userInput, String productName, Model productModel,
    System.out.println("\n**** SEARCH FOR PRODUCT ****"); Θ(1)
    System.out.println("--------------------------"); Θ(1)
                                                            Θ(1)     Θ(n)
    FurnitureBranch branch; // Branch that customer choose
    // User input
    branch = branches.get(userInput-1); Θ(n)

    // Products
Θ(1)tkHybridList<Furniture> productArr;  // product will be removed from this array                    Θ(1)
Θ(1)int productAmount = 0;               // shows how many available products are there for the specified product

    // User Input
    System.out.println("Product name  : " + productName); Θ(1)
    productArr = branch.getFurnitureArr(productName); Θ(1)      Θ(1)
    System.out.println("Product model : " + productModel); Θ(1)
    System.out.println("Product color : " + productColor); Θ(1)

    // Product amount
O(n)try {
        productAmount = branch.getCount(productArr, productColor, productModel); Θ(n)
    } catch (NullPointerException n) {...} Θ(1)

    // Getting customer's information
    if(productAmount == 0)  Θ(1)
Θ(1)    System.out.println("Sorry! The product you are looking for is out of stock."); Θ(1)
    else {
        // address
        System.out.println("Address : " + address); Θ(1)
        setAddress(address); Θ(1)

        // phone number
        System.out.print("Phone number : "); Θ(1)
        setPhoneNumber(Integer.parseInt(phone)); Θ(n)

        // removing the product from stock
        branch.removeProduct(productArr, productColor, productModel); Θ(n)
        // adding it as previous order
        setOrder(productArr.get(0).getName(), productModel, productColor); Θ(n)     Θ(n)
        // informing the customer
        System.out.println("Process completed. Your information:" + Θ(1)
                "\nName    : " + getName() + Θ(1)
                "\nSurname : " + getSurname() + Θ(1)
                "\nID      : " + getCustomerID() +    Θ(1)
                "\nAddress : " + getAddress() + Θ(1)
                "\nPhone   : " + getPhoneNumber()); Θ(1)
    }

}
```

T = Θ(n)



```
public void listOfProducts(Company company, tkLinkedList<FurnitureBranch> branches, int branch, int furnitureArr,

    System.out.println("\n**** LIST OF PRODUCTS ****");Θ(1)
    System.out.println("--------------------------");Θ(1)

    tkHybridList<Furniture> f = new tkHybridList<>();Θ(1)
    String furnitureName = ""; Θ(1)

    branch -= 1; Θ(1)

    if(branch < 0 && branch >= company.getBranchNumber()) {...} Θ(1)

    // Furniture Menu
    System.out.println("1) Office Chairs");
    System.out.println("2) Office Desks");
    System.out.println("3) Meeting Tables");          Θ(1)
    System.out.println("4) Bookcases");
    System.out.println("5) Office Cabinets");
    System.out.println("Enter: " + furnitureArr);

    switch (furnitureArr) {
        case 1:
            branches.get(branch).printFurnitureArr(branches.get(branch).officeChairs);Θ(n)
            f = branches.get(branch).officeChairs;Θ(n)
            furnitureName = "Office Chair";Θ(1)
            break;
        case 2:
            branches.get(branch).printFurnitureArr(branches.get(branch).officeDesks);Θ(n)
            f = branches.get(branch).officeDesks;Θ(n)
            furnitureName = "Office Desk";Θ(1)
            break;
        case 3:                                                                      Θ(n)
            branches.get(branch).printFurnitureArr(branches.get(branch).meetingTables);Θ(n)
            f = branches.get(branch).meetingTables;Θ(n)
            furnitureName = "Meeting Table"; Θ(1)
            break;
        case 4:
            branches.get(branch).printFurnitureArr(branches.get(branch).bookcases); Θ(n)
            f = branches.get(branch).bookcases;Θ(n)
            furnitureName = "Bookcase";Θ(1)
            break;
        case 5:
            branches.get(branch).printFurnitureArr(branches.get(branch).officeCabinets);Θ(n)
            f = branches.get(branch).officeCabinets;Θ(n)
            furnitureName = "Office Cabinet";Θ(1)
```

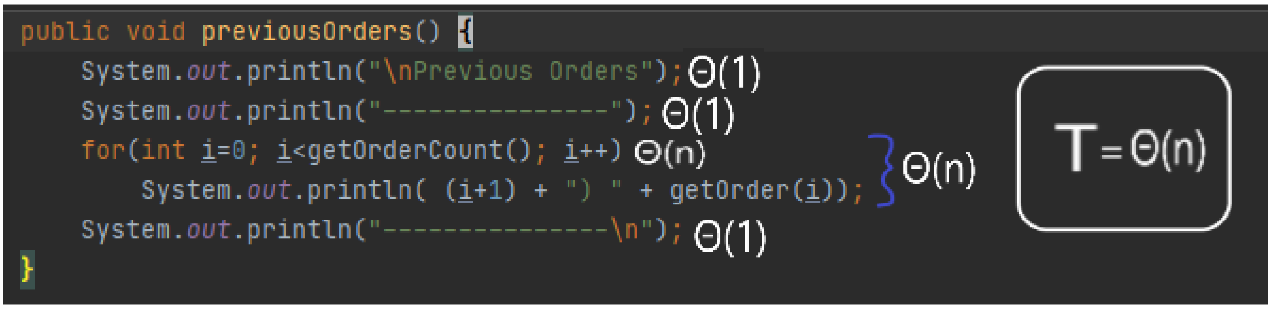
T = Θ(n)



```
    // User Input
    System.out.println("Enter the number of the product that you want to buy: " + furniture); Θ(1)
    furniture -= 1; Θ(1)
    if(furniture < 0 || furniture >= f.size())  Θ(1)
        System.err.println("Invalid choice!"); Θ(1)
    else {
        // address
        System.out.println("Address : " + address); Θ(1)
        setAddress(address); Θ(1)

        // phone number
        System.out.println("Phone number : " + phone); Θ(1)
        setPhoneNumber(Integer.parseInt(phone)); Θ(1)

        // informing the customer
        System.out.println("Process completed. Your information:" +
                "\nName    : " + getName() +                          Θ(n)
                "\nSurname : " + getSurname() +
                "\nID      : " + getCustomerID() +
                "\nMail    : " + getEmail() +
                "\nAddress : " + getAddress() +
                "\nPhone   : " + getPhoneNumber() +                                                     Θ(1)
                "\nProduct : " + furnitureName + " - " + f.get(furniture).getModel() + " - " + f.get(furniture).getColor());
                                                          Θ(n)      Θ(1)                                   Θ(n)

        // adding as a previous order
        setOrder(furnitureName, f.get(furniture).getModel(), f.get(furniture).getColor()); Θ(n)
        // removing product
        branches.get(branch).removeProduct(f, f.get(furniture).getColor(), f.get(furniture).getModel()); Θ(n)
    }


}
```

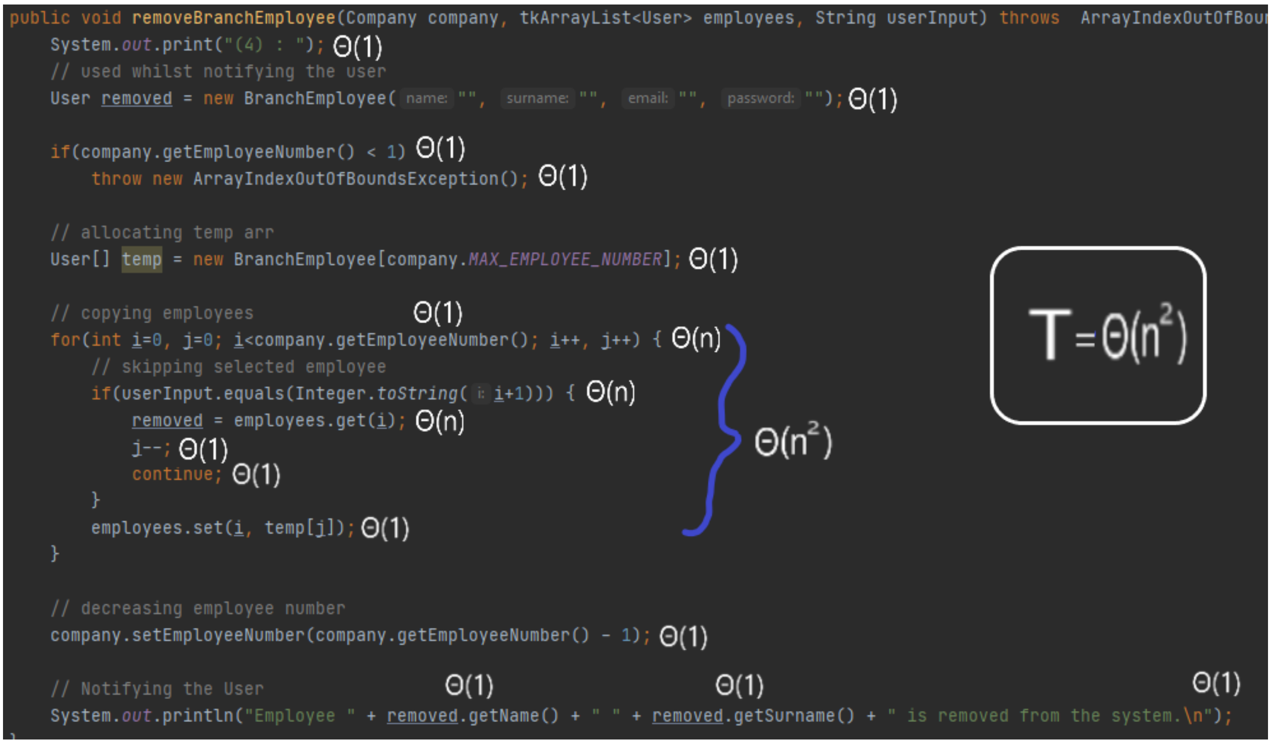
$T_b = \Theta(n)$

$T_w = \Theta(1)$



```
public void previousOrders() {
    System.out.println("\nPrevious Orders"); Θ(1)
    System.out.println("---------------"); Θ(1)
    for(int i=0; i<getOrderCount(); i++) Θ(n)
        System.out.println( (i+1) + ") " + getOrder(i)); } Θ(n)
    System.out.println("---------------\n"); Θ(1)
}
```

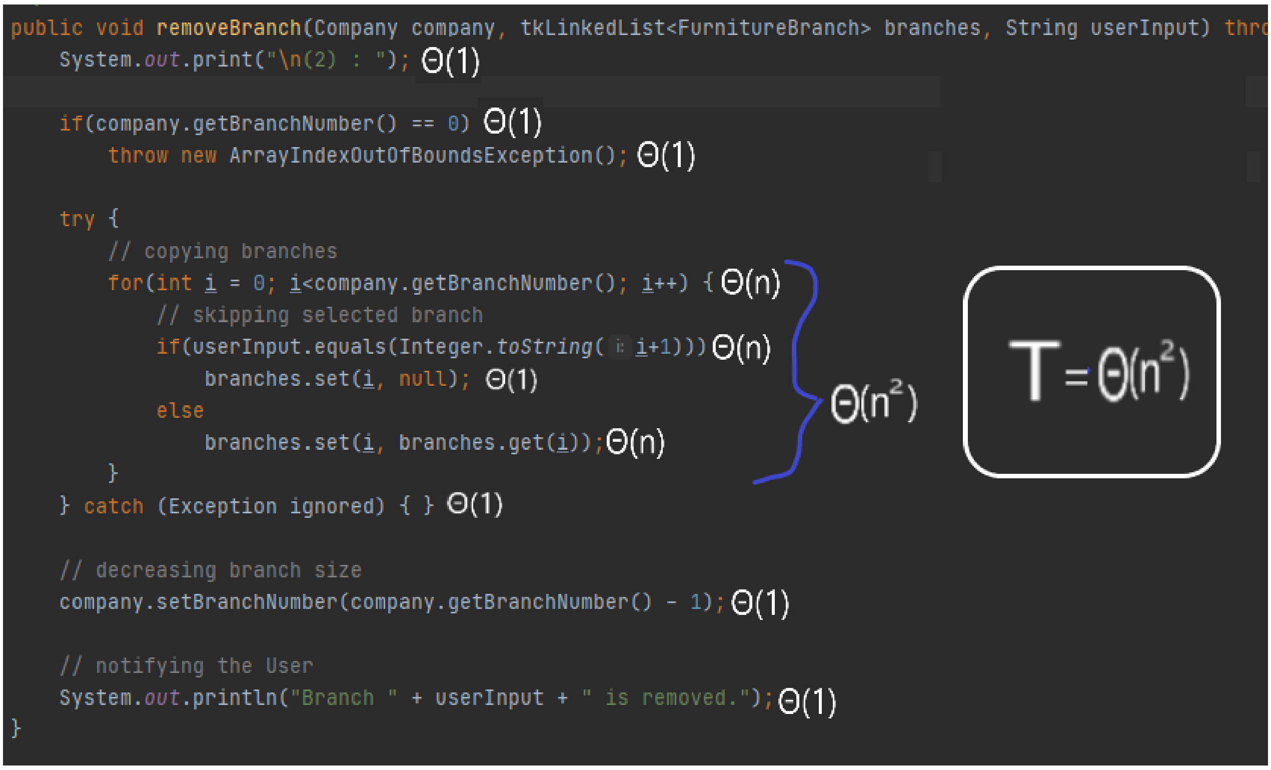
$T = \Theta(n)$



```
public void removeBranchEmployee(Company company, tkArrayList<User> employees, String userInput) throws  ArrayIndexOutOfBou
    System.out.print("(4) : "); Θ(1)
    // used whilst notifying the user
    User removed = new BranchEmployee( name: "",  surname: "",  email: "",  password: ""); Θ(1)

    if(company.getEmployeeNumber() < 1) Θ(1)
        throw new ArrayIndexOutOfBoundsException(); Θ(1)

    // allocating temp arr
    User[] temp = new BranchEmployee[company.MAX_EMPLOYEE_NUMBER]; Θ(1)

    // copying employees              Θ(1)
    for(int i=0, j=0; i<company.getEmployeeNumber(); i++, j++) { Θ(n)
        // skipping selected employee
        if(userInput.equals(Integer.toString( i: i+1))) { Θ(n)
            removed = employees.get(i); Θ(n)
            j--; Θ(1)
            continue; Θ(1)
        }
        employees.set(i, temp[j]); Θ(1)
    }

    // decreasing employee number
    company.setEmployeeNumber(company.getEmployeeNumber() - 1); Θ(1)

    // Notifying the User              Θ(1)                      Θ(1)                                       Θ(1)
    System.out.println("Employee " + removed.getName() + " " + removed.getSurname() + " is removed from the system.\n");
}
```

$\Theta(n^2)$

$T = \Theta(n^2)$



```
public void removeBranch(Company company, tkLinkedList<FurnitureBranch> branches, String userInput) thro
    System.out.print("\n(2) : "); Θ(1)

    if(company.getBranchNumber() == 0) Θ(1)
        throw new ArrayIndexOutOfBoundsException(); Θ(1)

    try {
        // copying branches
        for(int i = 0; i<company.getBranchNumber(); i++) { Θ(n)
            // skipping selected branch
            if(userInput.equals(Integer.toString( i: i+1))) Θ(n)
                branches.set(i, null); Θ(1)
            else
                branches.set(i, branches.get(i)); Θ(n)
        }
    } catch (Exception ignored) { } Θ(1)

    // decreasing branch size
    company.setBranchNumber(company.getBranchNumber() - 1); Θ(1)

    // notifying the User
    System.out.println("Branch " + userInput + " is removed."); Θ(1)
}
```

$\Theta(n^2)$

$T = \Theta(n^2)$