



STRUCTURING APHARMACY
DATASTORING SYSTEMTHATCONTAINS
TWOEMPLOYEES THE FIRST IS THEPHARMACIST
,AND THE SECOND IS THEASSISTANTWHO
HELPSCUSTOMERS TOCHOOSE PRODUCTS FORM
SEVERAL SECTIONSBEAUTY,CARE, MEDICAL
SUPPLIES,MEDICATIONS
EACH SECTION HAS SPECIFICPRODUCTS.


```
* @author Danah Alramadani, Sarah Alqufi, Haneen Almalki, Sara Alfaifi, Tala Aqeel
public static void main(String[] args) {
   //Array of sections in the pharmacy
   String[] sections = {"Beauty", "Care", "Medical Supplies", "Medications"};
   String[] beautySection = {"Dakeup", "Aails products", "Bair dye", "Cyes lenses"};
   String[] careSection = {"Products for body", "Products for skin care", "Products for hair", "Makeup removers"};
   String[] medicalSuppliesSection = {"First aid materials", "Wheel chairs", "Measuring devices", "Hot&Cold Bags"};
   String[] medicationsSection = {"Prescribed medications", "Non-Prescribed medications", "Sore throat Candies", "Vitamins");
   Employees employee1 = new Employees(name: "Omar Ahmad", ID: 1112234560, job: "Pharmacist", gender: 'M');
   Employees employee2 = new Employees(name: "Anas Turki", ID: 1113325406, 300: "Assistant", gender: 'M');
   System.out.println(x:"
                                                         [Welcome to DS pharmacy]
                                                                                                                \n");
   System.out.println(x: "Here is a list of our provided products and their sections: \n");
   System.out.println("Sections: " + sections[0] + ", " + sections[1] + ", " + sections[2] + ", " + sections[3] + ". |");
   System.out.println("Beauty Section has: " + beautySection[0] + ", " + beautySection[1] + ", " + beautySection[2] + ", " + beautySection[3] + ".|\n");
   System.out.println("Care Section has: " + careSection[0] + ", " + careSection[1] + ", " + careSection[2] + ", " + careSection[3] + ". |\n");
   System.out.println("Medical Supplies Section has: " + medicalSuppliesSection[0] + ", " + medicalSuppliesSection[1] + ", " + medicalSuppliesSection[2] + ", "
       + medicalSuppliesSection[3] + ".|\n");
   System.out.println("Medications Section has: " + medicationsSection[0] + ", " + medicationsSection[1] + ", " + medicationsSection[2] + ", "
      + medicationsSection[3] + ".|");
   System.out.println("Our " + employee1.job + " \"" + employee1.name + "\" AND our " + employee2.job + " \"" + employee2.name + "\" will be happy to serve you!~");
```

class

```
public class Employees {
    //Employees Infos
    String name;
    long ID:
    char gender;
    String job;
    public Employees() {
    ^{\star} Constructor to assign values for the employee information
    * @param name the name of the employee who serving the customer.
     ^{\star} \mbox{\em e} param ID the ID of the employee who serving the customer.
     * @param job The job of the employee who serves the customer.
     * @param gender the Gender of the employee who serving the customer.
    public Employees (String name, long ID, String job, char gender) {
        this.name = name;
        this.ID = ID;
        this.gender = gender;
        this.job = job;
}//End class
```



```
System.out.println(x: "\n_
                                                                     _[Implementing sort & search]_
   System.out.println(x: " ~~~~~~");
   System.out.println(x:"|insertion sort|");
   System.out.println(x: " ------);
   System.out.println(x: "-The sections array after sorting: ");
   Product.insertionSort(products sections);
   System.out.println(x: "\n-The Beauty Section array after sorting: ");
   Product.insertionSort(products:beautySection);
   System.out.println(x: "\n-The careSection array after sort: ");
   Product.insertionSort(products: careSection);
   System.out.println(x:"\n-The medicalSuppliesSection array after sorting: ");
   Product.insertionSort(products:medicalSuppliesSection);
   System.out.println(x: "\n-The medicationsSection array after sort: ");
   Product.insertionSort(products:medicationsSection);
   System.out.println(x: "
   //Implementing binary search algorithm
   System.out.println(x: " ~~~~~~~");
   System.out.println(x: "| binary search |");
   System.out.println(x: " ------);
   System.out.println(x: "if exist shows the index, -1 if not");
   System.out.println(x:"\n-Search for \"Snacks\" section: "); //Do we have a snacks section? in which index?
   System.out.println(x: Product.bSearch(products: sections, searchRey: "Snacks"));
   System.out.println(x: "~Search for \"Nails products\" in Beauty section: "); //Do we have nail products? in which index?
   System.out.println(x:Product.bSearch(products:beautySection, searchRey:"Nails products"));
   System.out.println(x: "-Search for \"Products for fitness\" in careSection: ");
   System.out.println(x: Product.bSearch(products: careSection, searchKey: "Products for fitness")); //Do we have products for fitness? in which index?
   System.out.println(x: "-Search for \"First aid materials\" in Medical supplies section: "); //Do we have First aid materials? in which index?
   System.out.println(x: Product.bSearch(products: medical Supplies Section, searchEap: "First aid materials"));
   System.out.println(x: "-Search for Non-Prescribed medications in Medications section: "); //Do we have non-Prescribed medications? in which index?
   System.out.println(x:Product.bSearch(products:medicationsSection, searchEast) "Non-Prescribed medications"));
      System.out.println(x: "_
      Product s = new Product();
      System.out.println(x: "If you bought from beautySection");
      System.out.println(x:s.details(ex: "beautySection"));
      System.out.println(x: "\nIf you bought from careSection");
      System.out.println(x:s.details(*x:"careSection"));
      System.out.println(x:"\nIf you bought from medicalSuppliesSection");
      System.out.println(x:s.details(ex: "medicalSuppliesSection"));
      System.out.println(x: "\nIf you bought from medicationsSection");
      System.out.println(x:s.details(xx: "medicationsSection"));
- }//End of main
```

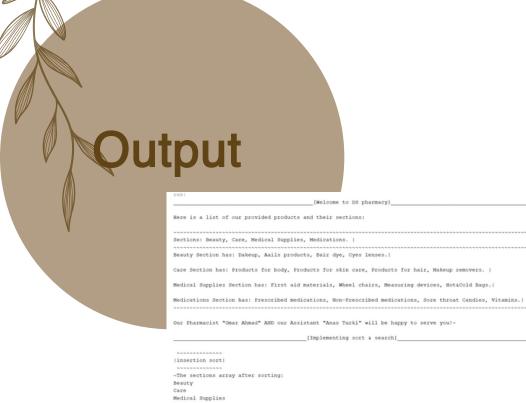
class

```
public class Product (
   string end_date;
   double price;
   public Product() [
   )
   * Constructor to assign values for the Products information
    * 8param end_date the Product Expiry Date .
    * Sparam price the Product Price .
   public Product(String end_date, double price) (
      this.end_date = end_date;
       this.price = price;
   // @param end_date Setters to set the products values depending on the type of product
   public String details(String ex) [
       switch (ex) (
           case "beautySection":
          return this.end_date = "The expiry will be from 6-12 months after openning the package";
          case "careSection":
          return this.end_date = "The expiry will be from 4-6 months after openning the package";
          case "medicalSuppliesSection":
          return this.end_date = "The expiry will be from 12-24 months after openning the package";
           case "medicationsSection":
          return this.end_date = "The expiry will be from 12-24 months after openning the package";
       }//End switch
      return"null" ;
```

class Product

```
public static void insertionSort(String products[]) {
         String temp;
         int i, j, k;
         for (i = 1; i < products.length; i++) {
             temp = products[i]; //st
             for (j = 0; j < i; j++) {
                 if (products[j].compareTo(anotherString:temp) > 0) ( // >0 products[j] First
                    break;
             for (k = i; k > j; k--) [
                products[k] = products[k - 1];
             products[j] = temp;
         for (i = 0; i < products.length; i++) (
             System.out.println(products[i]);
     1//End method
     public static int bSearch(String products[], String searchKey) {
         int left = 0, right = products.length - 1;
         while (left <= right) (
             int mid = (left + right) / 2;
             if (products[mid].compareTo(anotherString: searchKey) == 0) {
                 return mid;
             } else {
                 if (products[mid].compareTo(anotherString: searchKey) < 0) {
                 left = mid + 1;
                ) else {
                 right = mid - 1;
         return -1;
. ]//End methodSs
 1// end class
```

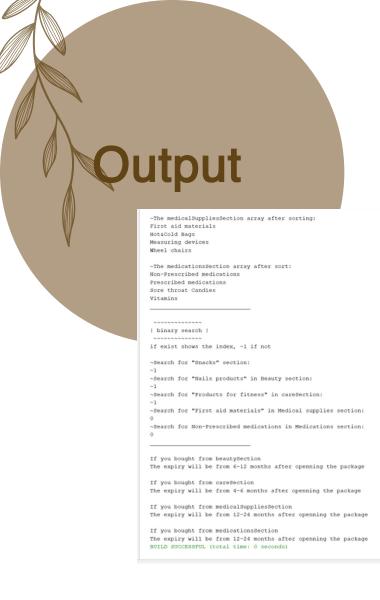
....



Medications

-The Beauty Section array after sorting:
Amils products
Bair dye
Cyes lenses
Dakeup

-The caresection array after sort:
Makeup removers
Products for body
Products for hair
Products for skin care





سارة الفيفي .. 443002606 حنين الهالكي .. 443016595 سارة العوفي .. 443006431 دانة الرهضاني .. 442013346 تالة عقيل .. 442013216

Presented to: Dr. Manal Al-Harbi Prepared by: Group 16