```
public class Product {
  private String name;
  private int price;
  private int amount;


  public Product(String name, int price, int amount) {
    this.name = name;
    this.price = price;
    this.amount = amount;
  }

  public Product(String name, int price) {
    ... // A2
  }

  public String getName() {
    return name;
  }

  public int getPrice() {
    return price;
  }

  public int getAmount() {
    return amount;
  }

  public int setPrice(int newPrice) {
     ... // A4
  }

  public int addAmount(int amount) {
     ...  // A5
  }

  public boolean sell() {
     ...  // A6
  }


  public static void main(String[] args) {
    Product prod = new Product("Banan", 6);
    System.out.println("toString : " + prod);
    System.out.println("sell     :"+ prod.sell());
    System.out.println("addAmount: " + prod.addAmount(5));
    System.out.println("toString : " + prod);
    System.out.println("sell     : " + prod.sell());
    System.out.println("sell     : " + prod.sell());
    System.out.println("toString : " + prod);
    System.out.println("setPrice : " + prod.setPrice(7));
    System.out.println("toString : " + prod);
  }
}

/* Output:
toString : Banan 6 0
sell     : false
addAmount: 5
toString : Banan 6 5
sell     : true
sell     : true
toString : Banan 6 3
setPrice : 6
toString : Banan 7 3

*/
```

```
import java.util.ArrayList;
import java.util.Scanner;
import java.io.*;

public class VendingMachine {
  private ArrayList<Product> products;
  private int size;              // Maximal number of different products
  private int cash;              // Sum of all successful sales

  public VendingMachine(int size) {
     ... // A7
  }

  public String toString() {
    return size + " " + products.toString();
  }

  public void listProducts() {
    ... // A8
  }

  public Product find(String name) {
    ... // A9
  }

  public boolean cashReceived(int amount) {
    // Skall inte skrivas
  }

  public boolean addOrUpdate(String name, int price, int amount) {
    ... // B1
  }

  public void save(String filename)
    throws IOException
  {
    PrintWriter pw = new PrintWriter(filename);
    for (Product prod: products) {
      pw.println(prod.toString());
    }
    pw.close();

  }

  public void sell(String name) {
     ... // B2
  }

  public ArrayList<Product> load(String filename)
      throws IOException
  {
    Scanner scan = new Scanner(new BufferedReader(new FileReader(filename)));
    ... // B3
    scan.close();
    return result;
  }



  public static void main(String[] args)
    throws IOException
  {
    VendingMachine vm = new VendingMachine(5);
    System.out.println("addOrUpdate(banan) :" + vm.addOrUpdate("banan", 8, 5));
    System.out.println("addOrUpdate(apelsin):" + vm.addOrUpdate("apelsin", 9, 3));
    System.out.println("addOrUpdate(melon) :" + vm.addOrUpdate("melon", 15, 2));
    System.out.println("addOrUpdate(komquat):" + vm.addOrUpdate("kumquat",9, 1));
    System.out.println("addOrUpdate(citron) :" + vm.addOrUpdate("citron", 7, 3));
    System.out.println("toString :" + vm.toString());
```



```
    System.out.println("addOrUpdate(citron) :" + vm.addOrUpdate("citron", 8, 2));
    System.out.println("toString :" + vm.toString());
    System.out.println("addOrUpdate(mandarin):" + vm.addOrUpdate("mandarin", 8, 2));
    System.out.println("toString :" + vm.toString());
    vm.listProducts();
    vm.sell("banan");
    vm.sell("melon");
    vm.sell("kumquat");
    vm.sell("citron");
    vm.sell("kumquat");
    vm.sell("kumquat");
    vm.sell("citron");
    System.out.println("toString :" + vm.toString());
    System.out.println(vm.load("vendingMachine.txt").toString());
  }
}

/* Output:
addOrUpdate(banan)   : true
addOrUpdate(apelsin) : true
addOrUpdate(melon)   : true
addOrUpdate(komquat) : true
addOrUpdate(citron)  : true
toString   : 5 [banan 8 5, apelsin 9 3, melon 15 2, kumquat 9 1, citron 7 3]
addOrUpdate(citron)  : true
toString   : 5 [banan 8 5, apelsin 9 3, melon 15 2, kumquat 9 1, citron 8 5]
addOrUpdate(mandarin): false
toString   : 5 [banan 8 5, apelsin 9 3, melon 15 2, kumquat 9 1, citron 8 5]
banan     8 kr
apelsin   9 kr
melon    15 kr
kumquat   9 kr
citron    8 kr
Betala 8
banan levereras.
Betala 15
melon levereras.
Betala 9
kumquat levereras.
Betala 8
citron levereras.
*** kumquat finns ej i lager
*** kumquat finns ej i lager
Betala 8
citron levereras.
toString   : 5 [banan 8 4, apelsin 9 3, melon 15 1, kumquat 9 0, citron 8 3]
*** Can't load aprikos
*** Can't load fikon
[aprikos 4 7, fikon 3 4]
*/
```