# CMPP 239

Assignment Four – Custom Problem

Martin Czerwinski and Nathan Faucher

Dec. 1, 2016

# Problem

To calculate discounts on a large order of computer hardware items. The discounts include a bulk discount and user entered promo codes. The program will display information about the customer’s order and savings.

# Analysis

**Functionality**

Prompt and get the infile name and the outfile name.

Determine the number of rows in the input file.

Extract data from the comma delimited input file lines and parse the input to the appropriate data type.

Tests to see if there are any zero values in the price column, if there are the system will exit.

Transliterates the item names to title case.

Calculates the cost of each item total before discounts are applied.

Applies bulk discount based on quantity of each item.

Prompts the user to enter promo code.

Applies custom promo code discounts to particular item prices in the price array.

Calculates the discounted cost of individual item totals.

Prints and Displays subtotals and totals in a table.

Calculates the highest and lowest savings per item, and the average savings per item.

Prints and Displays highest, lowest and average savings as a table.

**Inputs**

The infile name

The outfile name

User enters promo code names

code (String)

**Promo codes:**

FAMILY price \* 0.95

SPRING15 price \* 0.85

SUMMER15 price \* 0.85

WINTER15 price \* 0.85

FALL15 price \* 0.85

F3qytr Laptop price \* 0.85

Jxn4ns Keyboard price \* 0.75

Mouse price \* 0.75

udXyj6 Switch price \* 0.80

Router price \* 0.80

Gc47wX Ethernet cables \* 0.50

kc42Qj Desktop \* 0.90

Monitor \* 0.70

Keyboard \* 0.70

Mouse \* 0.70

DQB6HS Wireless \* 0.85

Ethernet cable \* 0.85

**Outputs**

itemName (String)

itemPrice (real 2 dp)

itemTotal (real 2 dp)

itemBulk (real 2 dp)

itemDiscount (real 2 dp)

itemQuantity (int)

itemSavings (real 2 dp)

highSavings (real 2 dp)

lowSavings (real 2 dp)

lowIndex (int)

highIndex (int)

avgSavings (real 2 dp)

**Formulas:**

totalPrice[i] = price[i] \* quantity[i]

totalPrice[index] = (price[index] \* specificDiscount) \* quantity[index]

itemSavings[i] = itemTotal[i] - itemDiscount[i]

totalSavings = totalSavings + itemSavings[i]

avgSavings = totalSavings / arraySize

# Test Data

**Input file**

laptop,673.60,15

keyboard,45.60,40

monitor,230.45,30

router,310.81,9

desktop comp,492.90,32

switch,168.46,16

ethernet cable,90.40,18

mouse,13.80,38

wireless router,80.59,9

**Promo code entered:**

Jxn4ns 25% off Mice and Keyboards

**Outputs**

ITEM NAME ITEM QUANTITY ORIGINAL TOTALS DISCOUNT TOTALS

Laptop 15 $10104.00 $ 9598.80

Keyboard 40 $ 1824.00 $ 1231.20

Monitor 30 $ 6913.50 $ 6222.15

Router 9 $ 2797.29 $ 2797.29

Desktop Comp 32 $15772.80 $14195.52

Switch 16 $ 2695.36 $ 2560.59

Ethernet Cable 18 $ 1627.20 $ 1545.84

Mouse 38 $ 524.40 $ 353.97

Wireless Router 9 $ 725.31 $ 725.31

Totals $42983.86 $39230.67

Today You Saved: $ 3753.19

Number of Records: 9

Average Savings per Item: $ 417.02

Highest Savings per Item 32 Desktop Comp $1577.28

Lowest Savings per Item 9 Router $0.00

# Algorithms

**Main**

Prompt and get infile path

Prompt and get outfile path

arraySize = getArraySize(inFilename);

while(i<arraySize)

{

String line=inputFile.nextLine();

String[] nextfield=line.split(",",-1);

itemName[i] = nextfield[0];

itemPrice[i] = Double.parseDouble(nextfield[1]);

itemQuantity[i] = Integer.parseInt(nextfield[2]);

i++;

}

# int i=0;

# for (i=0; i<arraySize; i++)

# {

# checkZer0(itemPrice[i],i);

# itemName[i] = upperCaser(itemName[i]);

# 

# itemBulk[i] = discount.bulk(itemQuantity[i],itemPrice[i]);

# itemTotal[i] = getTotal(itemPrice[i], itemQuantity[i]);

# }

# 

# while(!code.equals(""))

# {

# Prompt and get code input

# discount.promo(code);

# }

# 

# itemDiscount = discount.total(itemName, itemBulk, itemQuantity);

# 

# printTitles();

# for (i=0; i<arraySize; i++)

# {

# printReceipt(itemName[i],itemQuantity[i],itemTotal[i],itemDiscount[i]);

# }

# for (i=0; i<arraySize; i++)

# {

# itemSavings[i] = itemTotal[i] - itemDiscount[i];

# 

# if (itemSavings[i] > highSavings)

# {

# highSavings = itemSavings[i];

# highIndex = i;

# }

# if (itemSavings[i] < lowSavings)

# {

# lowSavings = itemSavings[i];

# lowIndex = i;

# }

# }

# double avgSavings = calcAvgSavings(itemTotal, itemDiscount, arraySize);

# printTotal(itemTotal, itemDiscount, arraySize, avgSavings);

# printCalcs(itemQuantity[highIndex],itemQuantity[lowIndex],itemName[highIndex],itemName[lowIndex],highSavings,lowSavings);

# **Methods in Main**

# 

# getInFile() throws IOException

# {

# String filename = sc.nextLine();

# File file = new File(filename);

# inputFile = new Scanner(file);

# 

# return filename;

# }

# 

# getOutFile()

# {

# System.out.print("Please enter the path and filename to which you wish to save the output: ");

# String Out = sc.nextLine();

# outfile = new UtilityClass(Out);

# outfile.openFile();

# return Out;

# }

# 

# 

# getTotal(double itemPrice, int itemQuantity)

# {

# double itemTotal = itemPrice\*itemQuantity;

# 

# return itemTotal;

# }

# 

# upperCaser(String itemName)

# {

# String temp1 = itemName.toUpperCase().substring(0,1);

# itemName = temp1 + itemName.substring(1);

# 

# int whitespace = itemName.indexOf(" ");

# if (whitespace != -1)

# {

# String temp2 = itemName.toUpperCase().substring(whitespace+1,whitespace+2);

# itemName = itemName.substring(0,whitespace+1) + temp2 + itemName.substring(whitespace+2);

# }

# return itemName;

# }

# 

# 

# checkZer0(double price,int i)

# {

# if (price == 0)

# {

# System.out.printf("\nError: the price in line %d is zero.",i+1);

# outfile.writeLineToFile("\nError: the price in line %d is zero.",i+1);

# System.exit(0);

# }

# }

# getArraySize(String inFilename

# {

# int i = 0;

# File file = new File(inFilename);

# Scanner inputFile = new Scanner(file);

# 

# String line=inputFile.nextLine();

# 

# while(line!=null)

# {

# i++;

# try

# {

# line=inputFile.nextLine();

# }

# 

# catch (Exception e)

# {

# break;

# }

# 

# }

# return i;

# }

# calcAvgSavings(double [] itemTotal, double [] itemDiscount, int arraySize)

# {

# Double [] itemSavings = new Double[arraySize];

# double totalSavings=0;

# 

# for (int i=0; i<arraySize; i++)

# {

# itemSavings[i] = itemTotal[i] - itemDiscount[i];

# 

# totalSavings = totalSavings + itemSavings[i];

# }

# 

# double avgSavings = totalSavings / arraySize;

# return avgSavings;

# }

# printTitles()

# {

# System.out.println("\n\nITEM NAME\tITEM QUANTITY\tORIGINAL ITEM TOTALS\tDISCOUNTED ITEM TOTALS");

# outfile.writeLineToFile("\n\nITEM NAME\t\tITEM QUANTITY\tORIGINAL ITEM TOTALS\tDISCOUNTED ITEM TOTALS");

# }

# 

# printReceipt(String itemName, int itemQuantity, double itemTotal, double itemDiscount)

# {

# System.out.printf("\n%-10s\t%d\t\t$%8.2f\t\t$%8.2f", itemName, itemQuantity, itemTotal, itemDiscount);

# outfile.writeLineToFile("\n%-10s\t\t%d\t\t\t$%8.2f\t\t$%8.2f",itemName, itemQuantity, itemTotal,itemDiscount);

# }

# 

# printTotal(double[] itemTotal, double[] itemDiscount, int arraySize, double avgSavings)

# {

# double tempTotal = 0;

# double tempDiscount = 0;

# 

# System.out.printf("\n\nTotal..........................................................");

# outfile.writeLineToFile("\n\nTotal..........................................................");

# 

# for(int i = 0; i < itemTotal.length; i++)

# {

# tempTotal = tempTotal + itemTotal[i];

# tempDiscount = tempDiscount + itemDiscount[i];

# }

# 

# System.out.printf("\n\t\t\t\t$%8.2f\t\t$%8.2f", tempTotal, tempDiscount);

# outfile.writeLineToFile("\n\t\t\t\t\t\t$%8.2f\t\t\t\t$%8.2f", tempTotal, tempDiscount);

# 

# tempTotal = tempTotal - tempDiscount;

# 

# System.out.printf("\n Today You Saved:\t\t\t\t\t$%8.2f", tempTotal);

# outfile.writeLineToFile("\n Today You Saved:\t\t\t\t\t\t\t$%8.2f", tempTotal);

# System.out.printf("\n Number of Records:\t\t\t\t\t %d", arraySize);

# outfile.writeLineToFile("\n Number of Records:\t\t\t\t\t\t\t%d", arraySize);

# System.out.printf("\n Average Savings per Item:\t\t\t\t$%8.2f", avgSavings);

# outfile.writeLineToFile("\n Average Savings per Item:\t\t\t\t$%8.2f", avgSavings);

# System.out.printf("\n\n");

# outfile.writeLineToFile("\n\n");

# }

# 

# printCalcs(int highItemQuantity, int lowItemQuantity, String highItemName, String lowItemName, double highSavings, double lowSavings)

# {

# System.out.println("\n/~~~~~~~~~~~~~~~~~~ SAVINGS BREAKDOWN BY ITEM ~~~~~~~~~~~~~~~~~~~/");

# System.out.print("\\\t\t\t\t\t\t\t\t \\");

# System.out.print("\n/\t\t\t\t\t\t\t\t /");

# System.out.print("\n\\\t\t\tQUANTITY\tITEM NAME\tSAVINGS\t \\");

# System.out.print("\n/\t\t\t\t\t\t\t\t /");

# System.out.printf("\n\\ Highest\t\t%d\t\t%s\t%.2f\t \\", highItemQuantity, highItemName, highSavings);

# System.out.printf("\n/ Lowest\t\t%d\t\t%s\t\t%.2f\t /", lowItemQuantity, lowItemName, lowSavings);

# System.out.print("\n\\\t\t\t\t\t\t\t\t \\");

# System.out.println("\n/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/");

# 

# outfile.writeLineToFile("\n\t\tQUANTITY\tITEM NAME\tSAVINGS");

# outfile.writeLineToFile("\nHighest\t\t%d\t\t%s\t%.2f", highItemQuantity, highItemName, highSavings);

# outfile.writeLineToFile("\nLowest\t\t%d\t\t%s\t%.2f", lowItemQuantity, lowItemName, lowSavings);

# }

**Class Discount**

SimpleDateFormat monthFormat = new SimpleDateFormat("MM");

bulk(int quantity, double price)

{

double priceDisc=0;

if (quantity >= 100)

{

priceDisc = price\*0.80;

}

else if (quantity >= 50)

{

priceDisc = price\*0.85;

}

else if (quantity >= 25)

{

priceDisc = price\*0.90;

}

else if (quantity >= 10)

{

priceDisc = price\*0.95;

}

else

{

priceDisc = price;

}

return priceDisc;

}

**Method for determining the user input promo codes**

promo(String code)

{

Boolean flagSwitchActivated = false;

int month = 0;

Date date = new Date();

month = Integer.parseInt(monthFormat.format(date));

switch(code)

{

case "":

{

flagSwitchActivated = true;

break;

}

case "F3qytr":

{

if (flagCheckLaptop == false)

{

System.out.println("Laptop Code!");

System.out.println("15% off laptops");

System.out.println();

flagSwitchActivated = true;

flagCheckLaptop = true;

break;

}

else

{

System.out.println("You Have Already Entered The Laptop Code!");

System.out.println();

flagSwitchActivated = true;

break;

}

}

case "Jxn4ns":

{

if (flagCheckComp == false)

{

System.out.println("Computer Peripherals Code!");

System.out.println("25% off mice & keyboards");

System.out.println();

flagSwitchActivated = true;

flagCheckComp = true;

break;

}

else

{

System.out.println("You Have Already Entered The Computer Peripherals Code!");

System.out.println();

flagSwitchActivated = true;

break;

}

}

case "udXyj6":

{

if (flagCheckNetwork == false)

{

System.out.println("Networking Code!");

System.out.println("20% off switches & routers");

System.out.println();

flagSwitchActivated = true;

flagCheckNetwork = true;

break;

}

else

{

System.out.println("You Have Already Entered The Networking Code!");

System.out.println();

flagSwitchActivated = true;

break;

}

}

case "Gc47wX":

{

if (flagCheckCable == false)

{

System.out.println("Ethernet Cable Code!");

System.out.println("Half Off Ethernet Cables!");

System.out.println();

flagSwitchActivated = true;

flagCheckCable = true;

break;

}

else

{

System.out.println("You Have Already Entered The Ethernet Cable Code!");

System.out.println();

flagSwitchActivated = true;

break;

}

}

case "kc42Qj":

{

if (flagCheckOffice == false)

{

System.out.println("Home Office Starter Pack!");

System.out.println("10% off Desktops, 30% off mice, keyboards & monitors!");

System.out.println();

flagCheckOffice = true;

flagSwitchActivated = true;

break;

}

else

{

System.out.println("You Have Already Entered The Home Office Starter Pack Code!");

System.out.println();

flagSwitchActivated = true; break;

}

}

case "DQB6HS":

{

if (flagCheckHomeNet == false)

{

System.out.println("Home Networking Starter Pack!");

System.out.println("15% off Wireless Access Points & Ethernet Cables!");

System.out.println();

flagCheckHomeNet = true;

flagSwitchActivated = true;

}

else

{

System.out.println("You Have Already Entered The Home Networking Starter Pack Code!");

System.out.println();

flagSwitchActivated = true;

break;

}

}

default:

{

System.out.println("Invalid Option");

System.out.println();

break;

}

}

if(flagSwitchActivated==false) //if the first switch was not activated

{

switch(code.toUpperCase())

{

case "FULLCOMMIE":

{

System.out.println("Glory to the motherland!");

System.out.println();

break;

}

case "SPRING15":

{

if(month==3 || month==4 || month==5)

{

if(flagCheckSeason == false)

{

System.out.println("Spring 15% off");

System.out.println();

flagCheckSeason = true;

break;

}

else

{

System.out.println("You have already entered the season discount.");

System.out.println();

break;

}

}

else

{

System.out.println("Invalid Option");

System.out.println();

}

break;

}

case "SUMMER15":

{

if(month==6 || month==7 || month==8)

{

if(flagCheckSeason == false)

{

System.out.println("Summer 15% off");

System.out.println();

flagCheckSeason =true;

break;

}

else

{

System.out.println("You have already entered the season discount.");

System.out.println();

break;

}

}

else

{

System.out.println("Invalid Option");

System.out.println();

}

break;

}

case "FALL15":

{

if(month==9 || month==10 || month==11)

{

if(flagCheckSeason == false)

{

System.out.println("Fall 15% off");

System.out.println();

flagCheckSeason = true;

break;

}

else

{

System.out.println("You have already entered the season discount.");

System.out.println();

break;

}

}

else

{

System.out.println("Invalid Option");

System.out.println();

}

break;

}

case "WINTER15":

{

if(month==12 || month==1 || month==2)

{

if(flagCheckSeason == false)

{

System.out.println("Winter 15% off");

System.out.println();

flagCheckSeason=true;

break;

}

else

{

System.out.println("You have already entered the season discount.");

System.out.println();

break;

}

}

else

{

System.out.println("Invalid Option");

System.out.println();

}

break;

}

case "FAMILY":

{

if (flagCheckFamily == false)

{

System.out.println("You Entered The Friends and Family Discount");

System.out.println();

flagCheckFamily = true;

break;

}

else

{

System.out.println("You Have Already Entered The Friends and Family Code!");

System.out.println();

break;

}

}

default:

{

System.out.println("Invalid Option");

System.out.println();

break;

}

}

}

flagSwitchActivated = false;

}

**Method for Promo Discount Calcualtions**

total(String[] names, double[] price, int[] quantity)

{

int index = 0;

double[] totalPrice = new double[price.length];

for(int i = 0; i < totalPrice.length; i++)

{

totalPrice[i] = price[i] \* quantity[i];

}

if(flagCheckLaptop == true)

{

index = ary.getIndex(names, "Laptop");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.85) \* quantity[index];

}

else

{

System.out.println("There are no Laptops in your order.");

}

}

if(flagCheckComp == true)

{

index = ary.getIndex(names, "Mouse");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.75) \* quantity[index];

}

else

{

System.out.println("There are no Mice in your order.");

}

index = ary.getIndex(names, "Keyboard");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.75) \* quantity[index];

}

else

{

System.out.println("There are no Keyboards in your order.");

}

}

if(flagCheckNetwork == true)

{

index = ary.getIndex(names, "Switch");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.80) \* quantity[index];

}

else

{

System.out.println("There are no Switches in your order.");

}

index = ary.getIndex(names, "Router");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.80) \* quantity[index];

}

else

{

System.out.println("There are no Routers in your order.");

}

}

if(flagCheckCable == true)

{

index = ary.getIndex(names, "Ethernet Cable");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.50) \* quantity[index];

}

else

{

System.out.println("There are no Ethernet Cables in your order.");

}

}

if(flagCheckOffice == true)

{

index = ary.getIndex(names, "Desktop Comp");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.90) \* quantity[index];

}

else

{

System.out.println("There are no Desktop Computers in your order.");

}

index = ary.getIndex(names, "Monitor");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.70) \* quantity[index];

}

else

{

System.out.println("There are no Monitors in your order.");

}

index = ary.getIndex(names, "Mouse");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.70) \* quantity[index];

}

else

{

System.out.println("There are no Mice in your order.");

}

index = ary.getIndex(names, "Keyboard");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.70) \* quantity[index];

}

else

{

System.out.println("There are no Keyboards in your order.");

}

}

if(flagCheckHomeNet == true)

{

index = ary.getIndex(names, "Ethernet Cable");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.70) \* quantity[index];

}

else

{

System.out.println("There are no Ethernet Cables in your order.");

}

index = ary.getIndex(names, "Wireless Router");

if (index != -1)

{

totalPrice[index] = (price[index] \* 0.70) \* quantity[index];

}

else

{

System.out.println("There are no Wireless Routers in your order.");

}

}

if(flagCheckSeason == true)

{

for(int i = 0; i < totalPrice.length; i++)

{

totalPrice[i] = (price[i]\*0.85) \* quantity[i];

}

}

if(flagCheckFamily == true)

{

for(int i = 0; i < totalPrice.length; i++)

{

totalPrice[i] = (price[i]\*0.95) \* quantity[i];

}

}

return totalPrice;

}

**Class ArrayUtil**

public static int getIndex(String[] array, String search)

{

int index = -1;

Boolean wasFound = false;

for(int i = 0; i < array.length; i++)

{

if(search.equals(array[i]) && wasFound == false

{

index = i;

wasFound = true;

}

else if(search.equals(array[i]) && wasFound == true)

{

System.out.printf("\nFATAL: Two Instances of %s Were Found Within The Array", search);

System.out.printf("\nFATAL: Check Your Infile For Duplicate Names");

System.out.printf("\nFATAL: Program Will Now Halt");

System.exit(-1);

}

}

return index;

}

checkDuplicate(String[] array)

{

for(int i = 0; i < array.length; i++)

{

for(int f = 0; f < array.length; f++)

{

if(array[i].equals(array[f]) && i != f)

{

System.out.printf("\nFATAL: Two Instances of \"%s\" Were Found Within The Array at line %d & line %d", array[i], (i+1), (f+1));

System.out.printf("\nFATAL: Check Your Infile For Duplicate Names");

System.out.printf("\nFATAL: Program Will Now Halt");

System.exit(-1);

}

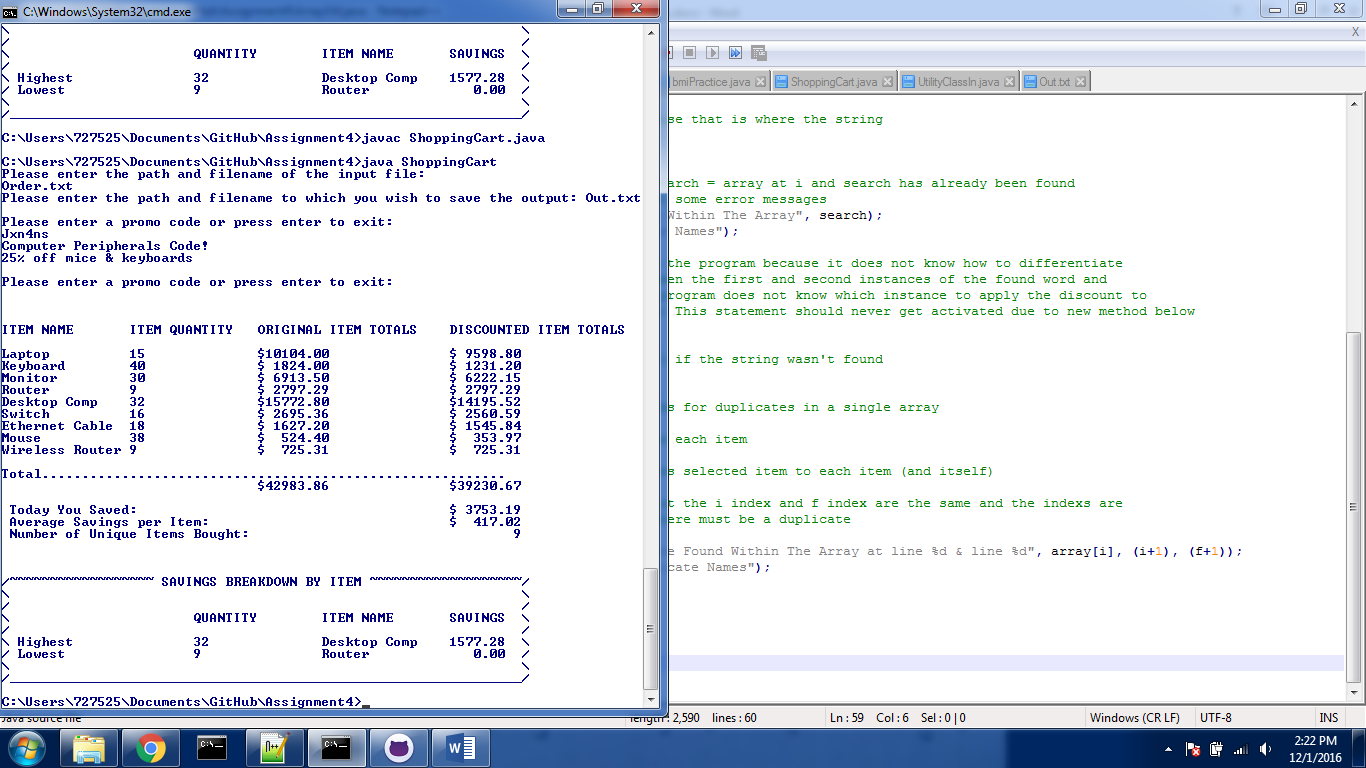
}

}

}

# Screenshots

Output with Test Data



Output with Zero Value in Price Column

