Problem1 Creating Accumulating Totals （4 Points）

The data set **orion.order\_fact** contains information about orders for several years, sorted by **Order\_Date**. Each observation represents one order, and **Total\_Retail\_Price** contains the sales value for the order.

Partial Listing of **orion.order\_fact** (617 Total Observations, 12 Total Variables)

Order\_ Total\_Retail\_

Order\_ID Date Price

1230058123 11JAN2007 $16.50

1230080101 15JAN2007 $247.50

1230106883 20JAN2007 $28.30

1230147441 28JAN2007 $32.00

1230315085 27FEB2007 $63.60

* 1. Orion Star would like to examine growth in sales during the date range of 01Nov2008 to 14Dec2008.
* Open file **p203e01**. It creates and prints a data set named **work.mid\_q4** from **orion.order\_fact**. The DATA step uses the following WHERE statement to read only the observations in the specified date range:

**where '01nov2008'd <= Order\_Date <= '14dec2008'd;**

* Modify the program to create an accumulating total, **Sales2Dte**, to display the sales-to-date total.
* Also create an accumulating variable, **Num\_Orders**, indicating how many orders-to-date that total represents. Each observation counts as one order.
  1. Modify the PROC PRINT step to show your results.
* Display **Sales2Dte** with a DOLLAR10.2 format.
* Show only the columns **Order\_ID**, **Order\_Date**, **Total\_Retail\_Price**, **Sales2Dte**, and **Num\_Orders**.

Partial PROC PRINT Output

Orders from 01Nov2008 through 14Dec2008

Order\_ Total\_Retail\_ Num\_

Obs Order\_ID Date Price Sales2Dte Orders

1 1234033037 01NOV2008 $53.70 $53.70 1

2 1234092222 10NOV2008 $7.20 $60.90 2

3 1234133789 17NOV2008 $328.30 $389.20 3

4 1234186330 25NOV2008 $200.10 $589.30 4

...

1. 1234301319 11DEC2008 $105.60 $1,664.20 10

**Problem2 Creating Accumulating Totals with Conditional Logic （3 Points）**

The data set **orion.order\_fact** contains a group of orders across several years, sorted by **Order\_Date**.

Partial Listing of **orion.order\_fact** (617 Total Observations, 12 Total Variables)

Order\_ Order\_

Order\_ID Type Date Quantity

1230058123 1 11JAN2007 1

1230080101 2 15JAN2007 1

1230106883 2 20JAN2007 1

1230147441 1 28JAN2007 2

1230315085 1 27FEB2007 3

1. Orion Star would like to analyze 2009 data by creating accumulating totals for the number   
   of items sold from retail, catalog, and Internet channels.

* The value of **Order\_Type** indicates whether the sale was retail (=*1*), catalog (=*2*),   
  or Internet (=*3*).
* Create a data set named **work.typetotals** with accumulating totals for **TotalRetail**, **TotalCatalog**, and **TotalInternet**, as described above.

**🖉** The variable **Quantity** contains the number of items sold for each order.

* For testing your program in this step, read only the first 10 observations that satisfy the WHERE statement.

**🖉** Remember to process only those rows where **Order\_Date** occurs in 2009.

1. Continue testing your program by printing the results from part **a**. Print all the variables and verify that the program is correctly calculating values for the accumulating totals.

PROC PRINT Output

Order\_ Delivery\_ Order\_

Obs Customer\_ID Employee\_ID Street\_ID Date Date Order\_ID Type Product\_ID

1 195 120150 1600101663 02JAN2009 02JAN2009 1234437760 1 230100600028

2 36 99999999 9260128237 11JAN2009 14JAN2009 1234534069 3 240800100026

3 183 120121 1600100760 12JAN2009 12JAN2009 1234537441 1 240100200001

4 16 99999999 3940105865 12JAN2009 14JAN2009 1234538390 2 220200300015

...

10 183 120179 1600100760 31JAN2009 31JAN2009 1234727966 1 240700400004

Total\_Retail\_ CostPrice\_ Total Total Total

Obs Quantity Price Per\_Unit Discount Retail Catalog Internet

1 2 $193.40 $48.45 . 2 0 0

2 4 $525.20 $58.55 . 2 0 4

3 1 $16.00 $6.35 . 3 0 4

4 1 $115.00 $52.40 . 3 1 4

...

10 1 $13.20 $5.95 . 4 6 6

1. When the results from parts **a** and **b** are correct, do the following:

* Modify the program to read all observations satisfying the WHERE statement. The resulting report contains 90 observations.
* Keep only the variables **Order\_Date**, **Order\_ID**, **TotalRetail**, **TotalCatalog**, and **TotalInternet**.
* Print your results with an appropriate title.

**Problem 3 Creating Accumulating Totals by Month (3 Points)**

The data set **orion.order\_fact** contains a group of orders across several years, sorted by **Order\_Date**.

Partial **orion.order\_fact** (617 Total Observations, 12 Total Variables)

Order\_ Total\_Retail\_

Order\_ID Date Price

1230058123 11JAN2007 $16.50

1230080101 15JAN2007 $247.50

1230106883 20JAN2007 $28.30

1230147441 28JAN2007 $32.00

1230315085 27FEB2007 $63.60

Orion Star would like to generate the following report showing all orders in 2011 along with an accumulating total:

* The accumulating total should reset to zero at the start of each new month.
* Remember to process only those rows where **Order\_Date** occurs in 2011.

Partial PROC PRINT Output (148 Total Observations)

Accumulating Totals by Month in 2011

Total\_

Order\_ Retail\_

Obs Date Order\_ID Price MonthSales

1 02JAN2011 1241054779 $195.60 $195.60

2 03JAN2011 1241063739 $160.80 $356.40

3 04JAN2011 1241066216 $306.20 $662.60

4 06JAN2011 1241086052 $37.80 $700.40

1. 13JAN2011 1241147641 $362.60 $1,063.00

**Problem 4 Summarizing Data Using the DATA Step (2 Points)**

The data set **orion.order\_summary** contains information about sales in a particular year for each customer, separated by month. For a given customer, there might be some months that he did not place an order.

Partial Listing of **orion.order\_summary** (101 Total Observations)

Order\_

Customer\_ID Month Sale\_Amt

5 5 478.00

5 6 126.80

5 9 52.50

5 12 33.80

10 3 32.60

1. Sort the input data set, **orion.order\_summary**, by **Customer\_ID**. Use the OUT= option to avoid overwriting the original data set. Name the output data set **work.sumsort**.
2. Create a new data set showing a total sales value for each customer.

* Name the new data set **work.customers**.
* Name the new variable **Total\_Sales**. This variable contains the total of sales across all months for each customer.

1. Print your result.

* Display **Total\_Sales** with a DOLLAR11.2 format.
* Add an appropriate title.

Partial PROC PRINT Output (37 Total Observations)

Total Sales to each Customer

Obs Customer\_ID Total\_Sales

1 5 $691.10

2 10 $3,479.09

3 11 $78.20

4 12 $253.20

1. 18 $29.40

**Problem 5 Summarizing and Grouping Data Using the DATA Step (5 Points)**

The data set **orion.order\_qtrsum** contains information about sales in a particular year for   
each customer, separated by month.

* For a given customer, there might be some months (and quarters) that the customer did not place an order.
* The variable **Order\_Qtr** contains the appropriate quarter.

Partial **orion.order\_qtrsum** (101 Total Observations)

Order\_ Order\_

Customer\_ID Qtr Month Sale\_Amt

69 4 10 3.2

70187 4 11 8.2

10 2 6 12.2

70079 4 10 14.6

70165 3 7 16.6

.

* 1. Create a data set named **work.qtrcustomers** that summarizes sales based on customer and quarter.
* The variable **Total\_Sales** should contain the total sales for each quarter within each **Customer\_ID** value.
* Create a variable named **Num\_Months** that counts the total months within each quarter that the customer had an order.
  1. Print your results.
* Display **Total\_Sales** with a DOLLAR11.2 format.
* Add an appropriate title.

Partial PROC PRINT Output (74 Total Observations)

Total Sales to each Customer for each Quarter

Order\_ Num\_

Obs Customer\_ID Qtr Total\_Sales Months

1 5 2 $604.80 2

2 5 3 $52.50 1

3 5 4 $33.80 1

4 10 1 $32.60 1

5 10 2 $342.80 3

1. **Summarizing Data and Conditional Output**

The data set **orion.usorders04** contains a group of orders from U.S. customers.

Partial **orion.usorders04** (83 Total Observations, 9 Total Variables)

Order\_ Total\_Retail\_

Order\_ID Customer\_ID Customer\_Name Type Price

1241054779 24 Robyn Klem 3 $195.60

1241063739 89 Wynella Lewis 1 $160.80

1241286432 27 Cynthia Mccluney 3 $174.40

1241359997 12 David Black 1 $117.60

1241461856 18 Tonie Asmussen 1 $29.40

1241623505 24 Robyn Klem 3 $46.90

1. Orion Star wants to reward customers who spent $100 or more through any particular sales channel (retail, catalog, or Internet).

* Create three data sets: **work.discount1**, **work.discount2**, and **work.discount3**.
* The value of **Order\_Type** indicates whether the sale was retail (=*1*), catalog (=*2*), or Internet (=*3*).
* The variable **Total\_Retail\_Price** contains the amount that the customer spent on each individual order.
* Create a variable named **TotSales** to hold the total sales to each customer by order type.
* Output to each of the three data sets based on the following table:

|  |  |
| --- | --- |
| **Customer spent $100 or more in…** | **Output to…** |
| Retail orders | **work.discount1** |
| Catalog orders | **work.discount2** |
| Internet orders | **work.discount3** |

**🖉** A given customer can output to more than one data set if he spent $100 or more in retail orders and the same in Internet orders, for example.

* Keep the variables **Customer\_ID**, **Customer\_Name**, and **TotSales**.
* Verify that the data sets **work.discount1**, **work.discount2**, and **work.discount3**   
  have 8, 2, and 5 observations, respectively.

1. Print your results from **work.discount1**.

* Format **TotSales** with a DOLLAR11.2 format.
* Add an appropriate title.

Partial **work.discount1**

Customers Spending $100 or more in Retail Orders

Customer\_ID Customer\_Name TotSales

5 Sandrina Stephano $213.10

10 Karen Ballinger $3,479.09

12 David Black $253.20

31 Cynthia Martinez $1,322.30

45 Dianne Patchin $700.28

**Problem 6 Identifying Extreme Values in Each Group of Data (3 Points)**

The data set **orion.customer\_dim** contains information about Orion Star customers.

Partial **orion.customer\_dim** (77 Total Observations, 11 Total Variables)

Customer\_

Customer\_ID Customer\_Name Customer\_Type BirthDate

4 James Kvarniq Orion Club members low activity 27JUN1978

5 Sandrina Stephano Orion Club Gold members medium activity 09JUL1983

9 Cornelia Krahl Orion Club Gold members medium activity 27FEB1978

10 Karen Ballinger Orion Club members high activity 18OCT1988

11 Elke Wallstab Orion Club members high activity 16AUG1978

Use First./Last. processing to create the report below. Show data on the oldest and youngest customers for each **Customer\_Type**.

* The variable **o\_ID** is the **Customer\_ID** value of the oldest customer and **y\_ID** is the **Customer\_ID** value of the youngest customer for each group.
* Create a variable named **agerange** to indicate the spread between these oldest and youngest customers.
* Use **Customer\_BirthDate**, rather than **Customer\_Age**, for all age determinations because this is more accurate.

Partial PROC PRINT Output (7 Total Observations)

Oldest and Youngest Customers of each Customer Type

Customer\_Type oldest youngest o\_ID y\_ID agerange

Internet/Catalog Customers 08JUL1938 18AUG1973 29 54655 35.1

Orion Club members high activity 28SEP1938 24OCT1990 89 46966 52.1

Orion Club members medium activity 20JAN1938 16SEP1992 70059 2806 54.7

Orion Club Gold members high activity 16JAN1938 25JUL1988 50 39 50.5

Orion Club Gold members low activity 19DEC1973 21JUL1992 70201 13 18.6