**Problem 1. Writing Observations Explicitly (2 Points)**

The **orion.prices** data set contains price information for Orion Star products.

Partial **orion.prices** (50 Total Observations)

Product\_ID Unit\_Price Factor

210200100009 $34.70 1.01

210200100017 $40.00 1.01

210200200023 $19.80 1.01

210200600067 $67.00 1.01

210200600085 $39.40 1.01

* 1. Write a DATA step to create a new data set that forecasts unit prices for the next three years. This data set contains three observations for each input observation read from **orion.prices**.
* Open file **p202e01**. It reads **orion.prices** and creates a new data set named **work.price\_increase**.
* Use explicit OUTPUT statements to forecast unit prices for the next three years, using **Factor** as the annual rate of increase.
  1. Print the new data set.
* Include only **Product\_ID**, **Unit\_Price**, and **Year** in the report.
* Verify your results.

Partial PROC PRINT Output (150 Total Observations)

Obs Product\_ID Unit\_Price Year

1 210200100009 $35.05 1

2 210200100009 $35.40 2

3 210200100009 $35.75 3

4 210200100017 $40.40 1

1. 210200100017 $40.80 2

**Problem 2 Writing Observations Explicitly (2 Points)**

The data set **orion.discount** contains information about various discounts that Orion Star runs on its products.

Partial **orion.discount**

Start\_ Unit\_Sales\_

Product\_ID Date End\_Date Price Discount

210100100027 01MAY2011 31MAY2011 $17.99 70%

210100100030 01AUG2011 31AUG2011 $32.99 70%

210100100033 01AUG2011 31AUG2011 $161.99 70%

210100100034 01AUG2011 31AUG2011 $187.99 70%

210100100035 01MAY2011 31MAY2011 $172.99 70%

1. Due to excellent sales, all discounts from December 2011 are repeated in July 2012. Both the December 2011 and the July 2012 discounts are called the Happy Holidays promotion.

* Create a new data set named **work.extended** that contains all discounts for the Happy Holidays promotion.
* Use a WHERE statement to read-only observations with a start date of 01Dec2011.
* Create a new variable, **Promotion**, which has the value *Happy Holidays* for each observation.
* Create another new variable, **Season**, that has a value of *Winter* for the December observations
* Use an explicit OUTPUT statement to write the December observations.
* Specify a new start date of 01Jul2012 and an end date of 31Jul2012 for the July 2012 discounts.
* For the July observations, overwrite value for **Season** with *Summer*.
* Use an explicit OUTPUT statement to write the July observations.
* Drop the **Unit\_Sales\_Price** variable.

1. Print the new data set.

* Add an appropriate title
* Verify the results.

Partial PROC PRINT (332 Total Observations)

Start\_

Obs Product\_ID Date End\_Date Discount Promotion Season

1 210200100007 01DEC2011 31DEC2011 50% Happy Holidays Winter

2 210200100007 01JUL2012 31JUL2012 50% Happy Holidays Summer

3 210200300013 01DEC2011 31DEC2011 50% Happy Holidays Winter

4 210200300013 01JUL2012 31JUL2012 50% Happy Holidays Summer

5 210200300025 01DEC2011 31DEC2011 50% Happy Holidays Winter

Problem 3 Creating Multiple SAS Data Sets (4 Points)

The data set **orion.employee\_organization** contains information about employee job titles, departments, and managers.

Partial **orion.employee\_organization** (424 Total Observations)

Employee\_ Manager\_

ID Job\_Title Department ID

120101 Director Sales Management 120261

120102 Sales Manager Sales Management 120101

120103 Sales Manager Sales Management 120101

120104 Administration Manager Administration 120101

120105 Secretary I Administration 120101

1. Create a separate data set for each department.

Name the data sets **work.admin**, **work.stock**, and **work.purchasing**.

Use conditional logic and explicit OUTPUT statements to write to these data sets depending on whether the value of **Department** is*Administration*, *Stock & Shipping*, or *Purchasing*, respectively. Ignore all other **Department** values.

Hint: Be careful with capitalization and the spelling of the **Department** values.

1. Print **work.admin** and verify your results. Add an appropriate title.

Partial **work.admin** (34 Total Observations)

Administration Employees

Employee\_ Manager\_

Obs ID Job\_Title Department ID

1 120104 Administration Manager Administration 120101

2 120105 Secretary I Administration 120101

3 120106 Office Assistant II Administration 120104

4 120107 Office Assistant III Administration 120104

5 120108 Warehouse Assistant II Administration 120104

1. Print **work.stock** and verify your results. Add an appropriate title.

Partial **work.stock** (26 Total Observations)

Stock and Shipping Employees

Employee\_ Manager\_

Obs ID Job\_Title Department ID

1 120670 Shipping Manager Stock & Shipping 120659

2 120671 Shipping Agent III Stock & Shipping 120670

3 120672 Shipping Manager Stock & Shipping 120659

4 120673 Shipping Agent II Stock & Shipping 120672

5 120677 Shipping Manager Stock & Shipping 120659

1. Print **work.purchasing** and verify your results. Add an appropriate title.

Partial **work.purchasing** (18 Total Observations)

Purchasing Employees

Employee\_ Manager\_

Obs ID Job\_Title Department ID

1 120728 Purchasing Agent II Purchasing 120735

2 120729 Purchasing Agent I Purchasing 120735

3 120730 Purchasing Agent I Purchasing 120735

4 120731 Purchasing Agent II Purchasing 120735

5 120732 Purchasing Agent III Purchasing 120736

Problem 4 Creating Multiple SAS Data Sets with Derived Values (4 Points)

The data set **orion.orders** contains information about in-store, catalog, and Internet orders   
as well as delivery dates.

Partial **orion.orders** (490 Total Observations)

Order\_ Order\_ Delivery\_

Order\_ID Type Employee\_ID Customer\_ID Date Date

1230058123 1 121039 63 11JAN2007 11JAN2007

1230080101 2 99999999 5 15JAN2007 19JAN2007

1230106883 2 99999999 45 20JAN2007 22JAN2007

1230147441 1 120174 41 28JAN2007 28JAN2007

1230315085 1 120134 183 27FEB2007 27FEB2007

1. Orion Star wants to study catalog and Internet orders that were delivered quickly, as well   
   as those that went slowly.

* Create three data sets named **work.fast**, **work.slow**, and **work.veryslow**.
* Write a WHERE statement to read only the observations with **Order\_Type** equal   
  to *2* (catalog) or *3* (Internet).
* Create a variable named **ShipDays** that is the number of days between when the order   
  is placed and when the order is delivered.
* Handle the output as follows:
* Output to **work.fast** when the value of **ShipDays** is less than *3*.
* Output to **work.slow** when the value of **ShipDays** is *5* to *7*.
* Output to **work.veryslow** when the value of **ShipDays** is greater than *7*.
* Do not output an observation when the value of **ShipDays** is *3* or *4*.
* Drop the variable **Employee\_ID**.
* There should be 80 observations in **work.fast**, 69 observations in **work.slow**, and   
  5 observations in **work.veryslow**.

**🖉** Of the 490 observations in **orion.orders**, only 230 are read due to the WHERE statement.

1. Print your results from **work.veryslow** with an appropriate title.

**work.veryslow**

Orders taking more than 7 days to deliver

Order\_ Order\_ Delivery\_ Ship

Obs Order\_ID Type Customer\_ID Date Date Days

1 1231305521 2 16 27AUG2007 04SEP2007 8

2 1236483576 2 70108 22JUL2009 02AUG2009 11

3 1236965430 3 70165 08SEP2009 18SEP2009 10

4 1237165927 3 79 27SEP2009 08OCT2009 11

5 1241298131 2 2806 29JAN2011 08FEB2011 10

Problem 5 Using a SELECT Group (4 Points)

Write a solution to the previous exercise using SELECT logic instead of IF-THEN/ELSE logic. Refer to SAS documentation to explore the use of a compound expression in a SELECT statement. Print the data set **work.veryslow**.

Problem 6 Specifying Variables and Observations (4 Points)

The data set **orion.orders** contains information about in-store, catalog, and Internet orders   
as well as delivery dates.

Partial **orion.orders** (490 Total Observations)

Order\_ Order\_ Delivery\_

Order\_ID Type Employee\_ID Customer\_ID Date Date

1230058123 1 121039 63 11JAN2007 11JAN2007

1230080101 2 99999999 5 15JAN2007 19JAN2007

1230106883 2 99999999 45 20JAN2007 22JAN2007

1230147441 1 120174 41 28JAN2007 28JAN2007

1230315085 1 120134 183 27FEB2007 27FEB2007

1230333319 2 99999999 79 02MAR2007 03MAR2007

1. Create two data sets, **work.instore** and **work.delivery**, to analyze in-store sales.

* Use a WHERE statement to read-only observations with **Order\_Type** equal to 1.
* Create a variable **ShipDays** that is the number of days between when the order was placed and when the order was delivered.
* Output to **work.instore** when **ShipDays** is equal to 0.
* Output to **work.delivery** when **ShipDays** is greater than 0.
* The **work.instore** data set should contain three variables (**Order\_ID**, **Customer\_ID**, and **Order\_Date**).

The **work.delivery** data set should contain four variables (**Order\_ID**, **Customer\_ID**, **Order\_Date**, and **ShipDays**).

* Test this program by reading the first 30 observations that satisfy the WHERE statement. Check the SAS log to verify that no warnings or errors were reported.

1. Modify the program to read the full **orion.orders** data set. Of the 490 observations in **orion.orders**, only 260 are read due to the WHERE statement.
2. Print your results from **work.delivery** with an appropriate title.

Partial **work.delivery** (10 Total Observations)

Deliveries from In-store Purchases

Order\_ Ship

Obs Order\_ID Customer\_ID Date Days

1 1231468750 52 25SEP2007 5

2 1231657078 63 29OCT2007 4

3 1232648239 49 07APR2008 8

4 1241063739 89 03JAN2011 1

5 1241235281 171 23JAN2011 7

1. Use PROC FREQ to display the number of orders per year in **work.instore**. Add an appropriate title.

Hint:Format the variable **Order\_Date** with a YEAR. format. Restrict the analysis to the variable O**rder\_Date** with a TABLES statement.

PROC FREQ Output

In-stock Store Purchases, By Year

The FREQ Procedure

Date Order was placed by Customer

Cumulative Cumulative

Order\_Date Frequency Percent Frequency Percent

ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ

2007 43 17.20 43 17.20

2008 50 20.00 93 37.20

2009 27 10.80 120 48.00

2010 67 26.80 187 74.80

2011 63 25.20 250 100.00