Problem 1 Performing Computations with Conditional DO Loops (2 Points)

The Orion Star Payroll Department must project total employee costs (wages, retirement benefits,   
and medical benefits) through future years, based on assumed increases.

* 1. Corporate income for last year was $50,000,000. Income is projected to increase at one %   
     per year.

Modify the program **p207\_PB1\_ORG** so that the DO loop stops when the year’s total costs exceed the year’s income.

* 1. Print **Year**, **Income**, and **Total\_Cost** and verify that total costs exceed income after 26 observations.

PROC PRINT Output

Obs Year Income Total\_Cost

1 2013 50,500,000.00 16,146,805.00

2 2014 51,005,000.00 17,058,159.57

3 2015 51,515,050.00 18,025,405.76

4 2016 52,030,200.50 19,052,105.42

...

26 2038 64,762,815.75 67,973,189.29

The results above were generated on February 14, 2012. Your values for **Year** might differ.

**Problem 2 Using an Iterative DO Statement with a Conditional Clause (2 Points)**

Orion’s income last year was $50,000,000 and expenses totaled $38,750,000. Income is projected   
to increase at 1% per year and expenses are expected to increase at 2% per year.

1. Create a SAS data set named **work.expenses** that contains each year’s projected income   
   and expenses.

* Use an iterative DO statement with a conditional clause.
* Stop the loop when expenses exceed income or after 30 years, whichever comes first.

1. Print the results and format **Income** and **Expenses** with a dollar sign and two decimal places.

Hint: Recall that an iterative DO statement with a conditional clause produces different results with DO WHILE and DO UNTIL statements. The results below were generated using a DO UNTIL statement.

PROC PRINT Output

Obs Income Expenses Year

1. $64,762,815.75 $64,844,951.93 26

**Problem 3 Using Other Loop Control Statements (2 Points)**

Orion’s income last year was $50,000,000 and expenses totaled $38,750,000. Income is projected   
to increase at 1% per year and expenses are expected to increase at 2% per year.

* 1. Use an iterative DO statement to calculate projected income and expenses for the next 75 years.
* **Investigate SAS documentation for information about other loop control statements such as CONTINUE and LEAVE.**
* Include the appropriate loop control statement (CONTINUE/LEAVE) to stop the loop when expenses exceed income.
  1. Print the results and format the values of **Income** and **Expenses** with dollar signs and two decimal places.

Hint: Recall that the DO WHILE and DO UNTIL statements might produce different results.

PROC PRINT Output

Obs Income Expenses Year

1. $64,762,815.75 $64,844,951.93 26

**Problem 4 Using Arrays for Repetitive Computations (2 Points)**

Monthly customer order data for the first half of the year is stored in the data set **orion.orders\_midyear**. The Orion Star Sales manager is considering a 5% price decrease next year. The manager wants to see how such a discount would affect this year’s income.

Partial **orion.orders\_midyear**

Obs Customer\_ID Month1 Month2 Month3 Month4 Month5 Month6

1 5 213.10 . 478.0 525.80 394.35 191.79

2 10 188.10 414.09 2876.9 3164.59 2373.44 169.29

3 11 78.20 . . . . 70.38

4 12 135.60 . 117.6 129.36 97.02 122.04

5 18 . . 29.4 32.34 24.26 .

6 24 93.00 265.80 . . . 83.70

7 27 310.70 782.90 . . . 279.63

8 31 1484.30 293.30 . . . 1335.87

9 34 642.50 . 86.3 94.93 71.20 578.25

10 41 134.00 119.20 313.0 344.30 258.23 120.60

11 45 443.88 216.20 40.2 44.22 33.17 399.49

12 49 24.80 . . . . 22.32

1. Create a data set, **discount\_sales**, to reflect the 5% discount.

* Create an array, **Mon**, to access **Month1** through **Month6**.
* Use a DO loop to adjust each customer’s monthly data. Apply the 5% discount.

1. Print the resulting data set and verify your results.

* Add an appropriate title.
* Use the DOLLAR format for the monthly sales amounts.

Partial PROC PRINT Output (24 Total Observations)

Monthly Sales with 5% Discount

Customer\_ID Month1 Month2 Month3 Month4 Month5 Month6

5 $202.45 . $454.10 $499.51 $374.63 $182.20

10 $178.70 $393.39 $2,733.06 $3,006.36 $2,254.77 $160.83

11 $74.29 . . . . $66.86

12 $128.82 . $111.72 $122.89 $92.17 $115.94

18 . . $27.93 $30.72 $23.04 .

**Problem 5 Using Arrays for Repetitive Computations (2 Points)**

Monthly customer order data for the first half of the year is stored in the data set **orion.orders\_midyear**. Orion Star Sales management is considering a 10% price decrease during the first three months of the upcoming year. Management wants to see how such a discount would affect this year’s sales.

Partial **orion.orders\_midyear**

Customer\_ID Month1 Month2 Month3 Month4 Month5 Month6

5 213.10 . 478.0 525.80 394.35 191.79

10 188.10 414.09 2876.9 3164.59 2373.44 169.29

11 78.20 . . . . 70.38

12 135.60 . 117.6 129.36 97.02 122.04

18 . . 29.4 32.34 24.26 .

* 1. Create a data set, **special\_offer**, including the 10% discount in months 1 through 3.
* Create an array, **Mon**, to access **Month1** through **Month3**.
* Use a DO loop to adjust each customer’s monthly data to include the 10% discount (**Month1** through **Month3** only).
* Create three new variables:
* **Total\_Sales** – the total of current sales over the six months
* **Projected\_Sales** – the total of the adjusted sales over the six months (including the   
  three discounted months)
* **Difference** – the difference between **Total\_Sales** and **Projected\_Sales**
* Keep only the new variables: **Total\_Sales**, **Projected\_Sales**, and **Difference**.
  1. Print the resulting data set and verify your results.
* Suppress the session start date, page number, and observation column.
* Add an appropriate title.
* Use the SUM statement to display a total for the **Difference** variable.
* Use the DOLLAR. format for all variables.

Partial PROC PRINT Output (24 Total Observations)

Total Sales with 10% Discount in First Three Months

Total\_ Projected\_

Sales Sales Difference

$1,803.04 $1,733.93 $69.11

$9,186.41 $8,838.50 $347.91

$148.58 $140.76 $7.82

$601.62 $576.30 $25.32

...

$1,518.95 $1,467.02 $51.93

==========

$1,550.74

**Problem 6 Using an Array for Table Lookup (2 Points)**

The manager of the Sales Department wants to identify preferred customers for an upcoming promotion. Use the **orion.orders\_midyear** data set and a lookup table to create a new data set, **preferred\_cust**.

Partial **orion.orders\_midyear**

Customer\_ID Month1 Month2 Month3 Month4 Month5 Month6

5 213.10 . 478.0 525.80 394.35 191.79

10 188.10 414.09 2876.9 3164.59 2373.44 169.29

11 78.20 . . . . 70.38

12 135.60 . 117.6 129.36 97.02 122.04

18 . . 29.4 32.34 24.26 .

1. Open the file **p207\_PB6\_ORG** and make the following changes:

* Create a temporary lookup table, **Target**, to hold the target sales amount for each month:   
   200, 400, 300, 100, 100, 200
* Create new variables, **Over1**, **Over2**, …,**Over6**, to hold the amount that a customer spent above the corresponding month’s target.
* Use a DO loop to calculate the values of **Over1** through **Over6** when the corresponding month’s sales amount exceeds the target.
* Store the sum of **Over1** through **Over6** in another new variable, **Total\_Over**.
* Write an observation only if **Total\_Over** is greater than 500.
* The new data set should include only **Customer\_ID**, **Over1** through **Over6**, and **Total\_Over**.

1. Print the new data set and verify your results.

Partial PROC PRINT Output (9 Total Observations)

Total\_

Customer\_ID Over1 Over2 Over3 Over4 Over5 Over6 Over

5 13.1 . 178.0 425.80 294.35 . 911.25

10 . 14.09 2576.9 3064.59 2273.44 . 7929.02

27 110.7 382.90 . . . 79.63 573.23

31 1284.3 . . . . 1135.87 2420.17

34 442.5 . . . . 378.25 820.75

**Problem 7 Using a Character Array for Table Lookup (2 Points)**

The Public Safety Department at Orion Star wants all employees to be aware of the new policies   
and procedures regarding customer incidents in their retail stores.

* Each employee must participate in web-based training and then take a multiple-choice test that consists of 10 questions.
* Each question has five choices (A through E).
* The test results from each testing session are entered into the SAS data set **orion.test\_answers** as shown below.
* Each observation in **orion.test\_answers** contains a single person’s answers.

Partial **orion.test\_answers** (15 Total Observations)

Employee\_

ID Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10

121044 A C C B D E D B B A

120145 B C C E E D B A A

120761 A C C B D D E B B C

120656 B C C A D B B C A D

121107 A C C B E E D B B A

* The correct answers for the questions are shown below:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Question: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Answer: | A | C | C | B | E | E | D | B | B | A |

* 1. Read **orion.test\_answers** and determine whether each person passed or failed the test.
* Compute a variable **Score** that contains the total correct answers for each person.

**🖉** Create a temporary array for the answer key.

* If an employee scores 7 or higher, write the observation to a data set named **passed**.
* If an employee scores less than 7, write the observation to a data set named **failed**.
  1. Print the **passed** data set to verify that it contains 12 observations.

Partial PROC PRINT Output (12 Total Observations)

Passed

Employee\_

Obs ID Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Score

1 121044 A C C B D E D B B A 9

2 120145 B C C E E D B A A 7

3 121107 A C C B E E D B B A 10

4 121038 B C C B D D D B B A 7

5 120273 C C C B E E E B B A 8

* 1. Print the **failed** data set to verify that it contains three observations.

PROC PRINT Output

Failed

Employee\_

Obs ID Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Score

1 120761 A C C B D D E B B C 6

2 120656 B C C A D B B C A D 2

3 120798 A C B D D D B B A 6

**Problem 8 Rotating a Data Set (2 Points)**

The data set **orion.orders\_midyear** contains an observation for each customer,   
with the total retail value of the customer’s monthly orders for the first half of the year.

Partial Listing of **orion.orders\_midyear** (24 Total Observations)

Obs Customer\_ID Month1 Month2 Month3 Month4 Month5 Month6

1 5 213.10 . 478.0 525.80 394.35 191.79

2 10 188.10 414.09 2876.9 3164.59 2373.44 169.29

3 11 78.20 . . . . 70.38

4 12 135.60 . 117.6 129.36 97.02 122.04

5 18 . . 29.4 32.34 24.26 .

* 1. Rotate **orion.orders\_midyear** to create an output data set, **sixmonths**, containing one observation per month for each customer.
* The data set should contain **Customer\_ID**, **Month**, and **Sales**.
* Do not output an observation if the monthly total is missing.
  1. Print the new data set. Verify the results.

Partial PROC PRINT Output (88 Total Observations)

Obs Customer\_ID Month Sales

1 5 1 213.10

2 5 3 478.00

3 5 4 525.80

4 5 5 394.35

5 5 6 191.79

**Problem 8 Rotating a Data Set and Using a Lookup Table (2 Points)**

The data set **orion.travel\_expense** contains an observation for each employee business trip and includes **Trip\_ID**, **Employee\_ID**, and up to five expenses, **Exp1-Exp5**. The table below shows the type of each expense:

|  |  |
| --- | --- |
| **Exp1** | Airfare |
| **Exp2** | Hotel |
| **Exp3** | Meals |
| **Exp4** | Transportation |
| **Exp5** | Miscellaneous |

Partial **orion.travel\_expense** (10 Total Observations)

Employee\_

Obs Trip\_ID ID Exp1 Exp2 Exp3 Exp4 Exp5

1 1044-1 121044 345.97 568.54 235.00 320.00 .

2 0145-1 120145 256.00 675.90 343.25 125.00 67.50

3 0656-1 120656 312.26 . 236.98 325.00 45.00

4 1119-1 121119 597.80 780.99 345.87 195.00 50.75

5 0812-1 120812 345.24 865.45 534.20 430.50 76.75

1. Rotate **orion.travel\_expense** to create an output data set named **travel** that contains one observation per nonmissing travel expense.

* The new data set should contain the variables **Trip\_ID**, **Employee\_ID**, **Expense\_Type**, and **Amount**.
* Use two arrays in your solution: one to refer to each of the five expenses in an observation and a second array to store the expense types as a lookup table.

1. Print the new data set.

* Format **Amount** with dollar signs, commas, and two decimal places.
* Compare your results to the partial listing below.

Partial PROC PRINT Output (44 Total Observations)

Employee\_

Obs Trip\_ID ID Expense\_Type Amount

1 1044-1 121044 Airfare $345.97

2 1044-1 121044 Hotel $568.54

3 1044-1 121044 Meals $235.00

4 1044-1 121044 Transportation $320.00

5 0145-1 120145 Airfare $256.00

**Problem 9 Performing a Match-Merge on Two Data Sets (10 Points)**

The data set **orion.web\_products** contains an observation for every product available for sale on Orion Star’s wholesale website.

Partial **orion.web\_products** (20 Total Observations)

Product\_ID Price Product\_Name

120400304333 114.36 Smasher Super Rq Ti 350 Tennis Racket

120400305288 53.26 Knife

120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes

120400308766 40.96 Big Guy Men's Packable Hiking Shorts

120400308849 12.23 Wood Box for 6 Balls

The data set **orion.web\_orders** contains a list of orders made in a single day from the website. Each observation contains the product ID, the quantity ordered, and the customer’s name.

Partial **orion.web\_orders** (43 Total Observations)

Product\_ID Quantity Customer

120400305288 16 Carglar Aydemir

120400305288 19 Sanelisiwe Collier

120400305846 13 Candy Kinsey

120400305846 13 Cynthia Martinez

120400305846 10 Rolf Robak

* 1. Create ***three***data sets:
* A data set named **revenue** contains the product code (**Product\_ID**), the price (**Price**), the quantity sold (**Quantity**), the product name (**Product\_Name**), the customer name (**Customer**), and the revenue generated from each sale (**Revenue**). **Revenue** is a new variable that is equal to **Price\*Quantity**.
* A data set named **notsold** contains the product code (**Product\_ID**), price (Price), and product name (**Product\_Name**) for each product that was not sold.
* A data set named **invalidcode** contains the product code (**Product\_ID**), quantity (**Quantity**), and customer name (**Customer**) for each observation in the **web\_orders** data set that does not have a corresponding product code in the **web\_products** data set.
  1. Print the three data sets with appropriate titles. The data sets should contain 39, 7, and 4 observations, respectively.

Partial **work.revenue** (39 Observations)

Revenue from Orders

Product\_ID Price Product\_Name Quantity Customer Revenue

120400305288 53.26 Knife 16 Carglar Aydemir 852.16

120400305288 53.26 Knife 19 Sanelisiwe Collier 1011.94

120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes 13 Candy Kinsey 1400.62

120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes 13 Cynthia Martinez 1400.62

120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes 10 Rolf Robak 1077.40

Partial **work.notsold** (7 Observations)

Products Not Ordered

Product\_ID Price Product\_Name

120400304333 114.36 Smasher Super Rq Ti 350 Tennis Racket

120400308849 12.23 Wood Box for 6 Balls

120400311211 69.16 Tipee Summer Sleeping Bag

120400317183 164.82 Smasher Rd Ti 70 Tennis Racket

120400329978 114.47 Tipee Twin Blue/Orange

Partial **work.invalidcode** (4 Observations)

Invalid Orders

Product\_ID Quantity Customer

120400311465 13 Thomas Leitmann

120400312556 7 Robyn Klem

120400315078 23 Tonie Asmussen

120400326278 10 Theunis Brazier

**Problem 10 Handling Same-Named Variables and Different Data Types for BY Variables (2 Points)**

The data set **orion.web\_products2** contains an observation for every product available   
for sale on Orion Star’s wholesale website.

Partial **orion.web\_products2** (20 Total Observations)

Product\_ID Price Name

120400304333 114.36 Smasher Super Rq Ti 350 Tennis Racket

120400305288 53.26 Knife

120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes

120400308766 40.96 Big Guy Men's Packable Hiking Shorts

120400308849 12.23 Wood Box for 6 Balls

The data set **orion.web\_orders2** contains a list of orders made in a single day from the   
website. Each observation contains the product ID, the quantity ordered, and the customer’s   
name.

Partial **orion.web\_orders2** (43 Total Observations)

Product\_ID Quantity Name

120400305288 16 Carglar Aydemir

120400305288 19 Sanelisiwe Collier

120400305846 13 Candy Kinsey

120400305846 13 Cynthia Martinez

120400305846 10 Rolf Robak

1. Create a new data set, **web\_converted**, from the **orion.web\_products2** data set. Change the type of **Product\_ID** to character. (Use the data set **web\_converted** to merge with **orion.web\_orders2** in the next step.)

Hint: Use the RENAME= data set option to change **Product\_ID** to some other name, such as **nProduct\_ID**, the LENGTH statement to declare a new character variable named **Product\_ID**, and an assignment statement with a PUT function to explicitly convert the numeric value in **nProduct\_ID** into a character value in **Product\_ID**.

1. Create ***three*** new data sets:

* A data set named **revenue** contains the product code, the price, the quantity sold, the product name, the customer name and the revenue generated from each sale. **Revenue** is calculated as **Price\*Quantity**.

**🖉** The **Name** variable in **web\_converted** refers to the product name and the **Name** variable in **web\_orders2** refers to the customer name. Give each variable an appropriate name in the **revenue** data set.

* A data set named **notsold** contains the product code, price, and product name for each product that was not sold.
* A data set named **invalidcode** contains the product code, quantity, and customer name for each observation in the **web\_orders2** data set that does not have a corresponding product code in the **web\_products2** data set.

1. Print the three data sets with appropriate titles. The data sets should contain 39, 7, and 4 observations, respectively. (The data sets you create might have different variable names   
   than the ones shown here.)

Partial **work.revenue** (39 Observations)

Revenue from Orders

Product\_ID Price Product\_Name Quantity Customer Revenue

120400305288 53.26 Knife 16 Carglar Aydemir 852.16

120400305288 53.26 Knife 19 Sanelisiwe Collier 1011.94

120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes 13 Candy Kinsey 1400.62

120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes 13 Cynthia Martinez 1400.62

120400305846 107.74 Big Guy Men's Air Deschutz Viii Shoes 10 Rolf Robak 1077.40

Partial **work.notsold** (7 Observations)

Products Not Ordered

Product\_ID Price Name

120400304333 114.36 Smasher Super Rq Ti 350 Tennis Racket

120400308849 12.23 Wood Box for 6 Balls

120400311211 69.16 Tipee Summer Sleeping Bag

120400317183 164.82 Smasher Rd Ti 70 Tennis Racket

120400329978 114.47 Tipee Twin Blue/Orange

Listing of **invalidcode** (4 Observations)

Invalid Orders

Product\_ID Quantity Name

120400311465 13 Thomas Leitmann

120400312556 7 Robyn Klem

120400315078 23 Tonie Asmussen

120400326278 10 Theunis Brazier