

Unit 2-7 Exercises

1. Performing Computations with DO Loops

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

a. Open the file **p207e01** and make the following changes:

- Insert a DO loop containing statements to calculate the estimated values of **Wages**, **Retire**, and **Medical**.
 - Use **Start** and **Stop** to control the values of the index variable, **Year**.
- Assume the estimated annual increase shown in the table below. For example, to calculate **Wages**, use this formula:

$$\text{wages} = \text{wages} * 1.06;$$


| Variable | Current Value | Estimated Annual Increase |
|----------|---------------|---------------------------|
| Wages | \$12,874,000 | 6.0% |
| Retire | 1,765,000 | 1.4% |
| Medical | 649,000 | 9.5% |

- Create another variable, **Total_Cost**, as the sum of that year's **Wages**, **Retire**, and **Medical** values.
- Output one observation for each year.

b. Print and verify your results.

PROC PRINT Output

| Obs | Year | Wages | Retire | Medical | Total_Cost |
|-----|------|---------------|--------------|--------------|---------------|
| 1 | 2009 | 13,646,440.00 | 1,789,710.00 | 710,655.00 | 16,146,805.00 |
| 2 | 2010 | 14,465,226.40 | 1,814,765.94 | 778,167.23 | 17,058,159.57 |
| 3 | 2011 | 15,333,139.98 | 1,840,172.66 | 852,093.11 | 18,025,405.76 |
| 4 | 2012 | 16,253,128.38 | 1,865,935.08 | 933,041.96 | 19,052,105.42 |
| 5 | 2013 | 17,228,316.09 | 1,892,058.17 | 1,021,680.94 | 20,142,055.20 |
| 6 | 2014 | 18,262,015.05 | 1,918,546.99 | 1,118,740.63 | 21,299,302.67 |
| 7 | 2015 | 19,357,735.95 | 1,945,406.64 | 1,225,020.99 | 22,528,163.59 |
| 8 | 2016 | 20,519,200.11 | 1,972,642.34 | 1,341,397.99 | 23,833,240.44 |
| 9 | 2017 | 21,750,352.12 | 2,000,259.33 | 1,468,830.80 | 25,219,442.24 |
| 10 | 2018 | 23,055,373.25 | 2,028,262.96 | 1,608,369.72 | 26,692,005.93 |

 The results above were generated on January 10, 2008. Your values for **Year** might differ.

- c.** Corporate income for last year was \$50,000,000. Income is projected to increase at one percent per year.

Modify the previous program so that the DO loop stops when the year's total costs exceed the year's income.

- d. 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 | 2009 | 50,500,000.00 | 16,146,805.00 |
| 2 | 2010 | 51,005,000.00 | 17,058,159.57 |
| 3 | 2011 | 51,515,050.00 | 18,025,405.76 |
| 4 | 2012 | 52,030,200.50 | 19,052,105.42 |
| 5 | 2013 | 52,550,502.51 | 20,142,055.20 |
| 6 | 2014 | 53,076,007.53 | 21,299,302.67 |
| 7 | 2015 | 53,606,767.61 | 22,528,163.59 |
| 8 | 2016 | 54,142,835.28 | 23,833,240.44 |
| 9 | 2017 | 54,684,263.63 | 25,219,442.24 |
| 10 | 2018 | 55,231,106.27 | 26,692,005.93 |
| 11 | 2019 | 55,783,417.33 | 28,256,519.13 |
| 12 | 2020 | 56,341,251.51 | 29,918,944.75 |
| 13 | 2021 | 56,904,664.02 | 31,685,647.29 |
| 14 | 2022 | 57,473,710.66 | 33,563,421.13 |
| 15 | 2023 | 58,048,447.77 | 35,559,520.91 |
| 16 | 2024 | 58,628,932.25 | 37,681,694.14 |
| 17 | 2025 | 59,215,221.57 | 39,938,216.30 |
| 18 | 2026 | 59,807,373.78 | 42,337,928.49 |
| 19 | 2027 | 60,405,447.52 | 44,890,278.01 |
| 20 | 2028 | 61,009,502.00 | 47,605,361.89 |
| 21 | 2029 | 61,619,597.02 | 50,493,973.81 |
| 22 | 2030 | 62,235,792.99 | 53,567,654.57 |
| 23 | 2031 | 62,858,150.92 | 56,838,746.30 |
| 24 | 2032 | 63,486,732.43 | 60,320,451.03 |
| 25 | 2033 | 64,121,599.75 | 64,026,893.56 |
| 26 | 2034 | 64,762,815.75 | 67,973,189.29 |



The results above were generated on January 10, 2008. Your values for **Year** might differ.

2. Using an Iterative DO Statement with a Conditional Clause

Orion's income last year was \$50,000,000 and expenses totaled \$38,750,000. Income is projected to increase at one percent per year and expenses are expected to increase at two percent per year.

- 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.
- 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 |

3. Using Other Loop Control Statements

Orion's income last year was \$50,000,000 and expenses totaled \$38,750,000. Income is projected to increase at one percent per year and expenses are expected to increase at two percent per year.

- Use an iterative DO statement to calculate projected income and expenses for the next 75 years.
 - Investigate SAS documentation for information on 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.
- 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 |

4. Using Arrays for Repetitive Computations

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 five-percent price decrease next year. He (the manager) wants to see how such a discount would affect this year's income.

Partial Listing of **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 |
|----|----|-------|---|---|---|---|-------|

- a. Create a data set, **discount_sales**, to reflect the five-percent discount.
 - Create an array, **Mon**, to access **Month1** through **Month6**.
 - Use a DO loop to adjust each customer's monthly data. Apply the five-percent discount.
- b. Print the resulting data set and verify your results.
 - Add an appropriate title.
 - Use the DOLLAR. format for the monthly sales amounts.

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 | . |
| 24 | \$88.35 | \$252.51 | . | . | . | \$79.52 |
| 27 | \$295.17 | \$743.76 | . | . | . | \$265.65 |
| 31 | \$1,410.09 | \$278.64 | . | . | . | \$1,269.08 |

5. Using Arrays for Repetitive Computations

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-percent 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 Listing of **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 | . |
| 24 | 93.00 | 265.80 | . | . | . | 83.70 |
| 27 | 310.70 | 782.90 | . | . | . | 279.63 |
| 31 | 1484.30 | 293.30 | . | . | . | 1335.87 |
| 34 | 642.50 | . | 86.3 | 94.93 | 71.20 | 578.25 |

- a. Create a data set, **special_offer**, including the 10-percent 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-percent 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**.

b. 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.

PROC PRINT Output

| Total Sales with 10% Discount in First Three Months | | | |
|---|-----------------|---------------------|------------|
| | Total_ Sales | Projected_ 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 |
| | \$86.00 | \$83.06 | \$2.94 |
| | \$442.50 | \$406.62 | \$35.88 |
| | \$1,373.23 | \$1,263.87 | \$109.36 |
| | \$3,113.47 | \$2,935.71 | \$177.76 |
| | \$1,473.18 | \$1,400.30 | \$72.88 |
| | \$1,289.33 | \$1,232.71 | \$56.62 |
| | \$1,177.16 | \$1,107.13 | \$70.03 |
| | \$47.12 | \$44.64 | \$2.48 |
| | \$928.07 | \$869.99 | \$58.08 |
| | \$37.90 | \$34.11 | \$3.79 |
| | \$1,027.85 | \$992.71 | \$35.14 |
| | \$47.85 | \$45.18 | \$2.67 |
| | \$925.42 | \$851.76 | \$73.66 |
| | \$1,334.97 | \$1,280.90 | \$54.07 |
| | \$501.57 | \$483.83 | \$17.74 |
| | \$697.60 | \$627.84 | \$69.76 |
| | \$4,306.28 | \$4,121.28 | \$185.00 |
| | \$147.10 | \$132.39 | \$14.71 |
| | \$177.84 | \$171.76 | \$6.08 |
| | \$1,518.95 | \$1,467.02 | \$51.93 |
| | | | ===== |
| | | | \$1,550.74 |

6. Terminating a DATA Step

Monthly order data for the first half of the year is stored in the **orion.orders_midyyear** data set. This data set is updated monthly and can contain data for one to 12 months.

Partial Listing of **orion.orders_midyyear**

| 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 |

- a. Orion Star decided to create a Frequent Shopper Program (FSP). The company wants to invite customers who placed orders in 50% of the months to date (assuming at least three months of data exist for that year), and who spent at least \$1000 since the beginning of the year.
- Open the input data set and verify that it contains at least three months of data.
 - If there is less than three months, write a message to the SAS log and stop the DATA step immediately.
 - If there are at least three months of data in the data set, create an array, **Mon**, to access the **Monthn** variables, regardless of how many **Monthn** variables exist in the data set.
Hint: Consider using a SAS Variable List to list the array elements.
 - Use a DO loop to examine each customer's data to determine if the customer qualifies for the Frequent Shopper Program.
 - Create a new data set, **fsp**, which contains an observation for each qualifying customer.
 - Use the report below to determine which variables to drop or keep.
- Hint: Use SAS documentation to investigate the use of the STOP statement and the DIM function.

- b. Print the resulting data set with an appropriate title and formats, and verify your results.

PROC PRINT Output

| orion.orders_midyear: Frequent Shoppers | | | |
|---|-------------|----------------------------|--------------------|
| Obs | Customer_ID | Total_ Order_ Amount | Months_ Ordered |
| 1 | 5 | \$1,803.04 | 5 |
| 2 | 10 | \$9,186.41 | 6 |
| 3 | 27 | \$1,373.23 | 3 |
| 4 | 31 | \$3,113.47 | 3 |
| 5 | 34 | \$1,473.18 | 5 |
| 6 | 41 | \$1,289.33 | 6 |
| 7 | 45 | \$1,177.16 | 6 |
| 8 | 65 | \$1,027.85 | 3 |
| 9 | 90 | \$1,334.97 | 6 |
| 10 | 171 | \$4,306.28 | 6 |
| 11 | 908 | \$1,518.95 | 3 |

- c. Test your program using **orion.orders_qtr1**.

PROC PRINT Output

| orion.orders_qtr1: Frequent Shoppers | | | |
|--------------------------------------|-------------|----------------------------|--------------------|
| Obs | Customer_ID | Total_ Order_ Amount | Months_ Ordered |
| 1 | 10 | \$3,479.09 | 3 |
| 2 | 27 | \$1,093.60 | 2 |
| 3 | 31 | \$1,777.60 | 2 |
| 4 | 171 | \$1,849.99 | 3 |
| 5 | 2806 | \$1,506.90 | 3 |

- d. Test your program using **orion.orders_two_months**.

Partial SAS Log

```
Insufficient data for Frequent Shopper Program
NOTE: There were 1 observations read from the data set ORION.ORDERS_TWO_MONTHS.
NOTE: The data set WORK.FSP has 0 observations and 3 variables.
NOTE: DATA statement used (Total process time):

1066
1067  title 'orion.orders_two_months: Frequent Shoppers ';
1068  proc print data=fsp;
1069      format total_order_amount dollar10.2;
1070  run;

NOTE: No observations in data set WORK.FSP.
```

7. Using an Array for Table Lookup

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 Listing of `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 | . |
| 24 | 93.00 | 265.80 | . | . | . | 83.70 |
| 27 | 310.70 | 782.90 | . | . | . | 279.63 |
| 31 | 1484.30 | 293.30 | . | . | . | 1335.87 |
| 34 | 642.50 | . | 86.3 | 94.93 | 71.20 | 578.25 |
| 41 | 134.00 | 119.20 | 313.0 | 344.30 | 258.23 | 120.60 |
| 45 | 443.88 | 216.20 | 40.2 | 44.22 | 33.17 | 399.49 |

a. Open the file `p207e07` 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. Note: If the sales did not exceed the target for a given month, then do not perform this calculation.
- 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 only include **Customer_ID**, **Over1** through **Over6**, and **Total_Over**.

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

PROC PRINT Output

| Customer_ID | Over1 | Over2 | Over3 | Over4 | Over5 | Over6 | Total_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 |
| 65 | . | . | 51.4 | 286.54 | 189.91 | . | 527.85 |
| 90 | . | . | 96.9 | 336.59 | 227.44 | . | 660.93 |
| 171 | . | 134.60 | 941.4 | 1265.54 | 924.16 | . | 3265.70 |
| 908 | . | . | 219.3 | 471.23 | 328.42 | . | 1018.95 |

8. Using a Character Array for Table Lookup

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-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.

Listing of **orion.test_answers**

| 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 |
| 121038 | B | C | C | B | D | D | D | B | B | A |
| 120273 | C | C | C | B | E | E | E | B | B | A |
| 120759 | A | C | C | B | E | E | D | B | B | A |
| 120798 | | A | C | B | D | D | D | B | B | A |
| 121030 | C | C | C | C | E | E | D | B | B | B |
| 121017 | B | B | E | B | E | E | D | B | B | A |
| 121062 | A | C | C | B | E | E | D | B | B | A |
| 121119 | C | C | C | B | E | E | D | B | B | A |
| 120812 | A | C | C | B | E | E | E | B | B | A |
| 120756 | 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 |

- a. 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**.

- b. Print the **passed** data set to verify that it contains 12 observations.

PROC PRINT Output

| Passed | | | | | | | | | | | | |
|--------|-----------------|----|----|----|----|----|----|----|----|----|-----|-------|
| Obs | Employee_ 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 |
| 6 | 120759 | A | C | C | B | E | E | D | B | B | A | 10 |
| 7 | 121030 | C | C | C | C | E | E | D | B | B | B | 7 |
| 8 | 121017 | B | B | E | B | E | E | D | B | B | A | 7 |
| 9 | 121062 | A | C | C | B | E | E | D | B | B | A | 10 |
| 10 | 121119 | C | C | C | B | E | E | D | B | B | A | 9 |
| 11 | 120812 | A | C | C | B | E | E | E | B | B | A | 9 |
| 12 | 120756 | A | C | C | B | E | E | D | B | B | A | 10 |

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

PROC PRINT Output

| Failed | | | | | | | | | | | | |
|--------|-----------------|----|----|----|----|----|----|----|----|----|-----|-------|
| Obs | Employee_ 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 |