Standard Pseudocode Solutions

By Mark Stocks

December 4, 2024

Contents

Chapter 1

Pseudocode Problems and Solutions

1.1 Example Problem 1

Problem: Compare two temperature readings from different sensors and print both temperatures in ascending order.

```
READ temperature1
READ temperature2
IF temperature1 < temperature2 THEN
    PRINT temperature1, temperature2
ELSE
    PRINT temperature2, temperature1
ENDIF
```

1.2 Example Problem 2

Problem: Read pH levels from 10 water samples, calculate the total pH, and print the result with the count of samples.

```
Set totalPH to 0
Set sampleCount to 0
FOR each sample up to 10
    READ pH_level
    Add pH_level to totalPH
    Increment sampleCount by 1
END FOR
PRINT totalPH, sampleCount
```

1.3 Example Problem 3

Problem: Read a series of experimental readings until a specific outlier value (e.g., -999) is encountered. Print each valid reading and calculate the cumulative total of these readings, excluding the outlier.

```
Set totalReadings to 0
Set readingCount to 0
```

```
READ reading
WHILE reading != -999
    PRINT reading
    Add reading to totalReadings
    Increment readingCount by 1
    READ next reading
END WHILE
PRINT totalReadings
```

1.4   Example Problem 4

Problem: Read a dataset of geological samples labeled as "fossil" (code 'F') or "rock" (code 'R'). Print all fossil records only.

```
FOR each record in dataset
    READ sampleCode
    IF sampleCode = 'F' THEN
        PRINT record
    ENDIF
END FOR
```

1.5   Example Problem 5

Problem: Print a report listing fossil samples that are over 100 million years old and larger than 2 cm. The report should include age, location, and size.

```
FOR each record in dataset
    READ sampleCode, age, size, location
    IF sampleCode = 'F' AND age > 100000000 AND size > 2 THEN
        PRINT age, location, size
    ENDIF
END FOR
```

1.6   Example Problem 6

Problem: Print the number of fossil samples that meet the criteria (age over 100 million years and size over 2 cm) and the total number of fossil records.

```
Set selectedCount to 0
Set fossilCount to 0
FOR each record in dataset
    READ sampleCode, age, size
    IF sampleCode = 'F' THEN
        Increment fossilCount by 1
        IF age > 100000000 AND size > 2 THEN
            PRINT age, location, size
            Increment selectedCount by 1
```

```
        ENDIF
    ENDIF
END FOR
PRINT "Selected fossils:", selectedCount
PRINT "Total fossils:", fossilCount
```

1.7 Example Problem 7

Problem: Count the number of each type of sample (sediment, mineral, fossil, artifact, other) and the total weight for each type.

```
Set sedimentCount, mineralCount, fossilCount, artifactCount, otherCount to 0
Set sedimentWeight, mineralWeight, fossilWeight, artifactWeight to 0
FOR each record in dataset
    READ recordCode, sampleID, weight
    IF recordCode = 1 THEN
        Increment sedimentCount by 1
        Add weight to sedimentWeight
    ELSE IF recordCode = 2 THEN
        Increment mineralCount by 1
        Add weight to mineralWeight
    ELSE IF recordCode = 3 THEN
        Increment fossilCount by 1
        Add weight to fossilWeight
    ELSE IF recordCode = 4 THEN
        Increment artifactCount by 1
        Add weight to artifactWeight
    ELSE
        Increment otherCount by 1
    ENDIF
END FOR
PRINT "Sediments:", sedimentCount, "Total weight:", sedimentWeight
PRINT "Minerals:", mineralCount, "Total weight:", mineralWeight
PRINT "Fossils:", fossilCount, "Total weight:", fossilWeight
PRINT "Artifacts:", artifactCount, "Total weight:", artifactWeight
PRINT "Other records:", otherCount
```

1.8 Example Problem 8

Problem: Calculate the estimated biomass for biological sample records based on mass and a growth multiplier (5%, 7%, 10%) depending on growth stage (S, M, L).

```
FOR each record in dataset
    READ sampleID, mass, growthStage
    IF growthStage = 'S' THEN
        Set biomass to mass * 0.05
    ELSE IF growthStage = 'M' THEN
        Set biomass to mass * 0.07
```

```
    ELSE IF growthStage = 'L' THEN
        Set biomass to mass * 0.10
    ENDIF
    PRINT "Sample ID:", sampleID, "Mass:", mass, "Estimated Biomass:", biomass
END FOR
```

1.9   Example Problem 9

Problem: Calculate the average temperature and count the number of days with below-average temperatures using functions.

```
FUNCTION calculateAverage(temperatures)
    Set total to 0
    FOR each temperature in temperatures
        Add temperature to total
    END FOR
    RETURN total / len(temperatures)
END FUNCTION


FUNCTION countBelowAverage(temperatures, average)
    Set belowCount to 0
    FOR each temperature in temperatures
        IF temperature < average THEN
            Increment belowCount by 1
        ENDIF
    END FOR
    RETURN belowCount
END FUNCTION


MAIN PROGRAM
    Set dailyTemperatures to an empty list
    Print "Enter the daily temperatures for the month (-1 to end):"
    WHILE True
        Input temperature
        IF temperature = -1 THEN
            BREAK
        ENDIF
        Add temperature to dailyTemperatures
    END WHILE

    IF len(dailyTemperatures) = 0 THEN
        Print "No temperatures entered. Exiting program."
        EXIT
    ENDIF

    Set averageTemperature to calculateAverage(dailyTemperatures)
    Set belowAverageDays to countBelowAverage(dailyTemperatures, averageTemperature)
```

```
        Print "Average Temperature:", averageTemperature
        Print "Number of days with below-average temperatures:", belowAverageDays
END MAIN PROGRAM
```