

2025 CpSc H Q7

Section: Software Design & Development

Topic: Implementation (Data Types and Structures) — Parallel 1D Arrays

Question Summary

CSV storm records are read into parallel 1D arrays: name(), region(), month(), year(). Tasks: (a) complete data-flow for Steps 2-4; (b) design an algorithm to count autumn storms for a chosen year; (c) write code to check whether a proposed storm name is already present.

Worked Solution

(a) Data-flow for Steps 2-4

Step	IN	OUT
1	name(), region(), month(), year()	— (arrays populated from file)
2	month(), year(), chosenYear	autumnCount (displayed)
3	year()	numStormsYear() (e.g., counts for 2020..2024)
4	numStormsYear()	yearWithMost (displayed)

(b) Design an algorithm (Step 2)

Pseudocode (structured English)

INPUT chosenYear

SET autumnCount $\leftarrow$ 0

FOR index FROM 0 TO length(year)–1 DO

IF year[index] = chosenYear THEN

IF month[index] IN {"September", "October", "November"} THEN

SET autumnCount $\leftarrow$ autumnCount + 1

END IF

END IF

END FOR

OUTPUT "Autumn storms in ", chosenYear, ": ", autumnCount

(c) Code to check for duplicate storm name

Language-neutral, Python-style pseudocode that matches the arrays used:

```
INPUT proposedName
SET found ← FALSE
FOR index FROM 0 TO length(name)–1 DO
  IF lower(name[index]) = lower(proposedName) THEN
    SET found ← TRUE
    BREAK
  END IF
END FOR
IF found = TRUE THEN
  OUTPUT "Name already present"
ELSE
  OUTPUT "Name is available"
END IF
```

Notes: using lower(...) gives a case-insensitive check; using BREAK avoids unnecessary comparisons once found.

Final Answer

- (a) Data-flow table as given (Step 2 uses month()+year()+chosenYear → autumnCount; Step 3 builds numStormsYear(); Step 4 outputs yearWithMost).
- (b) Algorithm loops once through arrays, counting autumn months (Sep/Oct/Nov) for chosenYear; outputs autumnCount.
- (c) Duplicate-name check loops through name() and reports present/available (case-insensitive).

Revision Tips

- Parallel arrays: keep indexes aligned — the i-th elements across arrays refer to the same record.
- When counting by condition, initialise a counter to 0 and increment inside the appropriate IF.

- Use clear IN/OUT for data-flow: Inputs are the arrays and any user selection; Outputs are the new values or displays.
- Prefer clear, readable month names (or map to numbers) to avoid spelling mismatches.

name(), region(),
month(), year

```
graph TD; A["name(), region(), month(), year"] --> B["Chosen year<br/>Count autumn storms<br/>Display"]; B --> C["year()<br/>numStormsYear()"]; C --> D["numStormsYear '<br/>yearWithMost<br/>Display"];
```

Chosen year
Count autumn storms
Display

year()
numStormsYear()

numStormsYear '
yearWithMost
Display

