2025 CpSc H Q8

Section: Software Design & Development

Topic: Implementation (Data Types & Structures)

Question Summary

An online reaction-time game stores player attempts with fields: name, e-mail, attemptID, time. Tasks cover record definition, array of records, finding the fastest time, why numPlays is needed, converting a binary fraction to floating-point (16-bit mantissa incl. sign, 8-bit exponent), and creating attemptID from e-mail (characters before ‘@’) with numPlays.

(a)(i) Define a suitable record structure

SQA-style pseudocode

TYPE PlayerAttempt

name : STRING

email : STRING

attemptID : STRING

reactionTime : REAL // seconds to the nearest thousandth

END TYPE

(a)(ii) Declare an array variable for up to 10 000 attempts

DECLARE attempts : ARRAY[0..9999] OF PlayerAttempt

DECLARE numPlays : INTEGER INITIALLY 0 // number of attempts recorded today

(b)(i) Code to find the fastest time (uses (a) declarations)

IF numPlays = 0 THEN

OUTPUT "No plays today"

ELSE

SET fastest ← attempts[0].reactionTime

```
  FOR index FROM 1 TO numPlays−1 DO
  IF attempts[index].reactionTime < fastest THEN
  SET fastest ← attempts[index].reactionTime
  END IF
  END FOR
  OUTPUT "Fastest time today: ", fastest
  END IF
```

(b)(ii) Why numPlays is needed

The array has a capacity of 10 000, but only the first numPlays elements contain valid data for today. Traversing all 10 000 would waste time and may read uninitialised entries. Using numPlays restricts the loop to exactly the number of actual attempts recorded.

(c) Convert 0.001_2 (i.e. 0.125_{10}) to floating-point

Normalise with the SQA convention: mantissa is 16 bits (including sign) with the binary point immediately after the sign bit. A normalised positive mantissa therefore starts as 0.1… and the exponent shifts the point.

• $0.001_2 = 0.125_{10}$.

• Write 0.001_2 as 0.1×2^{-2} (since $0.1_2 = 0.5$ and $0.5 \times 2^{-2} = 0.125$).

• Therefore: sign = 0 (positive); mantissa = 0.1 followed by zeros to 16 bits (incl. sign); exponent = −2.

Bit patterns depend on the specific course convention for exponent encoding:

• Two’s complement 8-bit exponent: −2 = 11111110.

• Excess-128 (bias) exponent: −2 + 128 = 126 = 01111110.

For mantissa (16 bits including sign): sign=0; then binary point; leading 1 followed by zeros. A valid 16-bit mantissa field is 0 1000 0000 0000 000 (sign bit ‘0’ then ‘1’ then zeros).

In assessments, you’ll receive full marks for the correct normalisation and a consistent bit representation that matches your class convention

(either two’s-complement exponent or excess-bias).

(d) Build attemptID using characters before ‘@’ and numPlays

```
FUNCTION findCharIndex(value : STRING, character : STRING)
RETURNS INTEGER

DECLARE positionChar : INTEGER INITIALLY −1

FOR index FROM 0 TO length(value)−1 DO

IF value[index] = character THEN

SET positionChar ← index

END IF

END FOR

RETURN positionChar

END FUNCTION

// Line 70: assign position of '@' in email

SET position ← findCharIndex(email, "@")

// Line 71: characters before '@' concatenated with numPlays

IF position ≠ −1 THEN

SET localPart ← substring(email, 0, position)

SET attemptID ← localPart & toString(numPlays)

ELSE

// No '@' found — fallback or error handling

SET attemptID ← email & toString(numPlays)

END IF
```

Final Answer

• (a) Record and array declarations as shown.

• (b)(i) Single pass (0..numPlays−1) to find minimum reaction time; (ii) numPlays prevents scanning unused entries.

• (c) 0.001_2 normalises to 0.1×2^{-2}; sign=0; mantissa begins 0.1…; exponent = −2 (encode per your course convention).

• (d) Use findCharIndex to get position of ‘@’; attemptID is the local-part concatenated with numPlays.

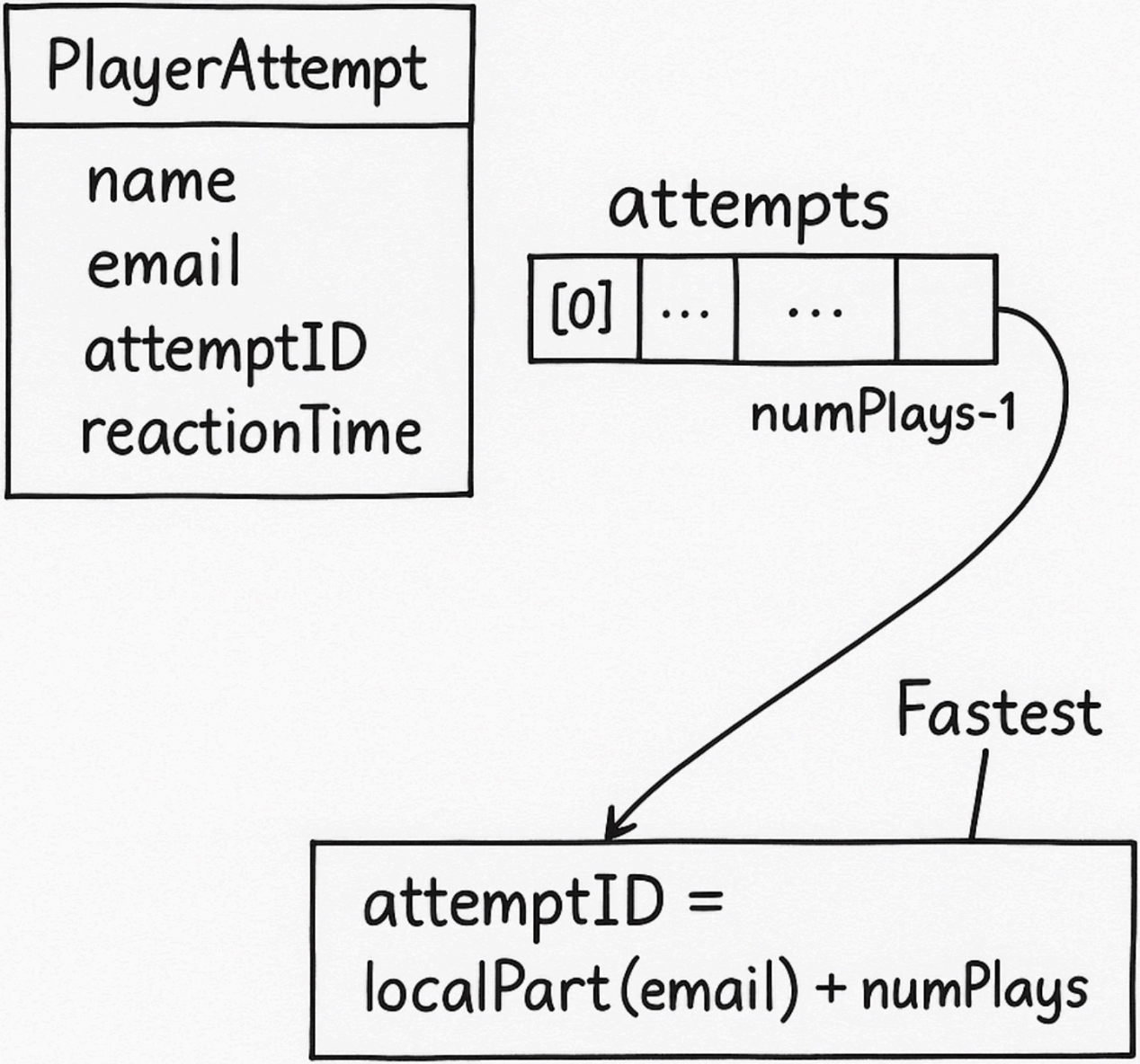
Revision Tips

• Records group related fields; an array of records stores many attempts.

• Always track a count (numPlays) to avoid scanning unused array elements.

• For floating-point: normalise mantissa to start 0.1… (SQA convention) and encode the exponent consistently.

• String operations with indices (find, substring) are typical 1–2 mark tasks — write them clearly.


PlayerAttempt
name
email
attemptID
reactionTime
attempts
[0]
...
...
numPlays-1
Fastest
attemptID =
localPart(email) + numPlays