2024 CpSc Q8

Section: Software Design and Development

Topic: Unique random integers — boundaries, debugging a Boolean function, efficiency, parameters

Question Summary

(a) Identify two boundaries of the problem.
(b) The error is in function validNum (lines 34–42).
    (i) Identify the error.
    (ii) Describe the effect when lines 63–70 are executed.
(c) Describe how to make validNum more efficient.
(d) Identify a formal parameter for validNum and its associated actual parameter.

Worked Solution

(a) Boundaries
1) The requested number of values must not exceed the size of the integer range: count ≤ (maximum − minimum + 1) (otherwise uniqueness is impossible).
2) The range limits are inclusive and ordered: minimum ≤ maximum, and any generated value may be exactly the minimum or the maximum.
Other acceptable boundaries (any two): count ≥ 0; inputs must be integers; handle the case count = 0; reject negative range sizes.

(b)(i) Identify the error
The function initialises valid to FALSE and never sets it to TRUE. It should return TRUE when randomNum is not found

in values, but the current code only ever assigns FALSE again on a match and otherwise leaves it FALSE.

(b)(ii) Effect when lines 63–70 run
Because validNum(...) always returns FALSE, the loop WHILE validNum(randomVal, randomList) = FALSE never terminates (infinite loop) — new random numbers are generated forever and no values are written to randomList.

(c) Making validNum more efficient
• Early exit: as soon as a match is found, return FALSE immediately; if the loop finishes with no match, return TRUE.
• Better data structure: store already-used numbers in a set (hash-based) for O(1) average membership checks, or keep a Boolean ‘used’ array indexed by value offset.
• Alternative overall design: generate the full range [minimum..maximum], shuffle, and take the first count items (guaranteed unique).

Formal vs Actual Parameters

Formal parameter	Actual parameter (in call)
randomNum	randomVal (in line 66)
values	randomList (in line 66)

Final Answer

Final Answer
(a) e.g., count ≤ (max − min + 1) and minimum ≤ maximum (inclusive bounds).
(b)(i) valid never becomes TRUE; logic should indicate ‘not

found’.
(b)(ii) While-loop never ends → program hangs generating randoms.
(c) Early return on match / use a set or Boolean lookup / or shuffle the range.
(d) Formal: randomNum ↔ Actual: randomVal; Formal: values ↔ Actual: randomList.

Revision Tips

- For uniqueness, always check that requested count fits the available range.
- Boolean search pattern: start TRUE and flip to FALSE on match, or start FALSE and set TRUE when not found — but be consistent.
- Break/return as soon as the answer is known (early exit).
- Sets give O(1) membership on average — ideal for ‘already used?’ checks.

Exam Alignment

Exam Alignment
Matches 2024 MI: two sensible boundaries; identifies the Boolean logic error and infinite-loop effect; suggests an efficiency improvement (early exit or data structure); and correctly pairs formal with actual parameters.