

2024 CpSc Q9

Section: Software Design and Development

Topic: Data Flow, Algorithm Design (filter & average), and Security (DoS — resource starvation)

Question Summary

- (a) Complete data flows for steps 2-4.
- (b) Explain how identifying data flow at design stage assists implementation.
- (c) Design an algorithm for Step 3 (average distance for journeys > 1 hour).
- (d) (i) Describe a resource starvation DoS attack, (ii) State one cost to the company.

Worked Solution

Step	IN/OUT	Data flow
1	IN	Text file of journey records (date, distance miles, driving time hours)
1	OUT	Arrays: date[], distance[], drivingTime[]
2	IN	distance[], drivingTime[]
2	OUT	avgSpeed[] — per-journey average speed = distance[i] / drivingTime[i]
3	IN	distance[], drivingTime[]

3	OUT	avgDistance — average distance of journeys where drivingTime > 1.0
4	IN	avgSpeed[], distance[], avgDistance
4	OUT	File of average speeds for journeys where distance > avgDistance

(b) Clear data flows act like ‘contracts’ between steps. They:

- specify the arrays/values each step must read and produce;
- guide variable declarations and parameter lists during coding;
- reduce bugs by making dependencies explicit;
- help testing by defining input and output for each step.

Algorithm for Step 3

```

ALGORITHM AverageDistanceOver1hr(distance[],
drivingTime[]) RETURNS REAL
total ← 0.0
count ← 0
FOR i FROM 0 TO length(distance[])-1 DO
IF drivingTime[i] > 1.0 THEN
total ← total + distance[i]
count ← count + 1
ENDIF
ENDFOR
IF count = 0 THEN RETURN 0.0 ELSE RETURN total / count
ENDIF
END ALGORITHM

```

(d)(i) A resource starvation DoS attack overwhelms CPU, memory, or bandwidth so legitimate requests cannot be served.

(d)(ii) One cost: lost revenue due to downtime (customers unable to buy or access services).

Final Answer

Final Answer

(a) See table above.

(b) Data flows define inputs/outputs, guiding modular code and testing.

(c) Pseudocode filters journeys with time > 1h and averages distances.

(d) Resource exhaustion prevents service; cost = lost revenue.

Revision Tips

- Name arrays consistently across steps to make data flow obvious.
- Use running totals and counts for averages.
- When filtering, define the condition clearly.
- Describe file outputs precisely (what data and which condition).

Exam Alignment

Exam Alignment

Matches 2024 MI: correct flows for steps 2–4, explanation of data flow purpose, valid algorithm, and accurate DoS description and company cost.

