

Solutions

Each answer shows the steps, then the **result**.

2.1

- a solution to a problem
- expressed as a sequence of defined steps

2.2

- (a) `Age >= 13 AND Age <= 19`
- (b) `UserName <> "" AND LENGTH(UserName) <= 12`
- (c) `Age < 5 OR Age >= 80`

2.3

- a sensible identifier and type for each row, for example:

identifier	data type	description
MemberName	STRING	the member's name
Visits	INTEGER	the number of visits this month
FeePaid	BOOLEAN	whether this month's fee has been paid
MonthlyFee	REAL	the monthly fee

2.4

- declarations and `Count ← 0`
- a post-condition loop that ends when the word is "STOP"
- input the word inside the loop
- `IF LENGTH(Word) > 5` adds 1 to `Count`
- output `Count` after the loop

```
DECLARE Word : STRING
DECLARE Count : INTEGER
Count ← 0
REPEAT
    INPUT Word
    IF LENGTH(Word) > 5 THEN
        Count ← Count + 1
    ENDIF
UNTIL Word = "STOP"
OUTPUT Count
```

2.5 any four of:

- set the fare to 3.00
- multiply the distance in km by 1.50
- add this amount to the fare
- check whether the time is from midnight to 6 a.m.

- if it is, multiply the fare by 2

3.1 any five of:

- set the total mileage, the number of cars and the number of high-mileage cars to zero
- input the mileage of the first car
- repeat until the mileage input is zero
- add each mileage to the total, and add 1 to the number of cars
- if a mileage is more than 100 000, add 1 to the number of high-mileage cars
- input the next mileage at the end of each pass
- after the loop, output the total divided by the number of cars, and the number of high-mileage cars

3.2

(a) set **Attempts** to 0; a loop containing **INPUT PIN** and adding 1 to **Attempts**

- "Try again" only while the PIN is wrong and fewer than 3 attempts have been made
- the loop ends when the PIN is correct **or** 3 attempts have been made
- an IF after the loop outputs "Access granted" or "Locked"

```

DECLARE PIN : STRING
DECLARE Attempts : INTEGER
Attempts ← 0
REPEAT
    INPUT PIN
    Attempts ← Attempts + 1
    IF PIN <> "4821" AND Attempts < 3 THEN
        OUTPUT "Try again"
    ENDIF
UNTIL PIN = "4821" OR Attempts = 3
IF PIN = "4821" THEN
    OUTPUT "Access granted"
ELSE
    OUTPUT "Locked"
ENDIF

```

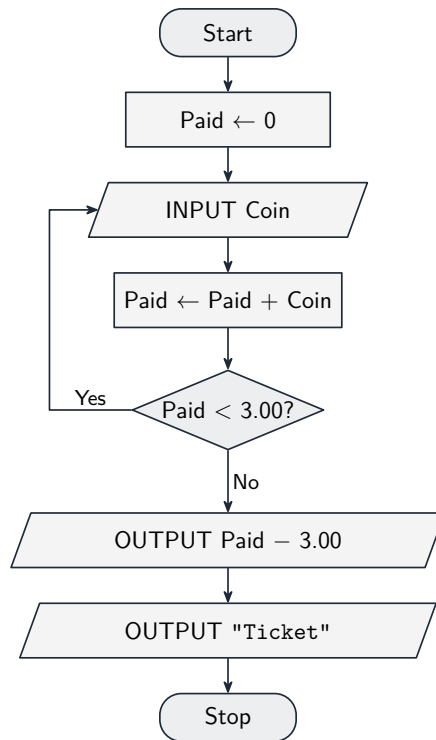
(b) a **post-condition** loop

- the PIN must be input at least once before it can be checked, and the decisions come after the input

3.3

(a) Start and Stop in rounded boxes; **Paid** ← 0 in a rectangle

- an input parallelogram, then **Paid** ← **Paid** + **Coin**
- a decision **Paid** < 3.00? whose **Yes** exit goes back to the input
- both outputs in parallelograms after the **No** exit



(b) a **post-condition** loop

- the decision comes **after** the input, so at least one coin is always input

3.4 any **five** of, in a sensible order:

- set the goods total to 0
- for each item, multiply the price by the quantity
- add each result to the goods total
- if the goods total is less than 50.00, add 5.00 for delivery
- calculate the tax as 20% of this amount
- add the tax to give the invoice total