

AP Computer Science Principles | Unit 3

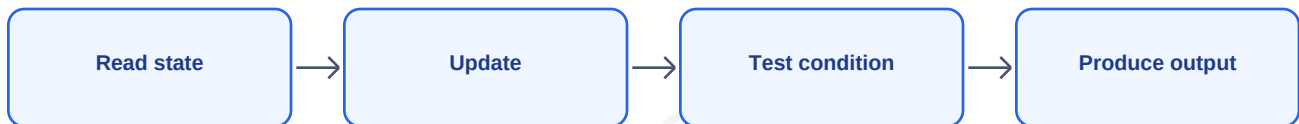
3.1 Sequencing, Selection, and Iteration

Solutions

Reference scenario

Trace pseudocode using AP CSP assignment and condition rules.

```
x <- 2
REPEAT 3 TIMES {
  x <- x + 2
  IF (x > 5) { DISPLAY(x) }
}
```



3.1.1

Trace: total <- 0; FOR EACH n IN [2, 5, 4] { IF (n > 3) { total <- total + n } }; DISPLAY(total).

2 is skipped. 5 changes total to 5, and 4 changes it to 9. The displayed value is 9.

3.1.2

Write pseudocode that displays the even integers from 2 through 8 inclusive.

```
n <- 2
REPEAT 4 TIMES {
  DISPLAY(n)
  n <- n + 2
}
```

The output is 2, 4, 6, 8.

AP Computer Science Principles | Unit 3

3.2 Variables and Lists

Solutions

Reference scenario

Track a list update using one-based AP CSP indexing.

```
items <- [4, 7, 9]
items[2] <- items[1] + 1
APPEND(items, 12)
DISPLAY(items[2])
```

3.2.1

Start with values = [3, 6, 8]. Remove the item at position 1, then insert 5 at position 2. What is the final list?

Removing position 1 gives [6, 8]. Inserting 5 at position 2 shifts 8 right, producing [6, 5, 8].

3.2.2

Write pseudocode that counts how many items in values are greater than 10, including an empty list.

```
count <- 0
FOR EACH value IN values {
  IF (value > 10) { count <- count + 1 }
}
DISPLAY(count)
An empty list leaves count at 0.
```

AP Computer Science Principles | Unit 3

3.3 Procedures, Parameters, and Return Values

Solutions

Reference scenario

Distinguish returning a value from displaying it.

```
PROCEDURE addFee(price, fee) {  
    RETURN(price + fee)  
}  
result <- addFee(12, 3)  
DISPLAY(result * 2)
```

3.3.1

If `addFee(8, 2)` is called without `DISPLAY`, does the procedure shown print 10? Explain.

No. It returns 10 to its caller; returning and displaying are distinct operations. A surrounding `DISPLAY` would be needed to show it.

3.3.2

Write a procedure `countAbove(values, cutoff)` that returns the number of list items strictly greater than `cutoff`.

```
PROCEDURE countAbove(values, cutoff) {  
    count <- 0  
    FOR EACH value IN values {  
        IF (value > cutoff) { count <- count + 1 }  
    }  
    RETURN(count)  
}  
For [2, 7, 7] and cutoff 2, the returned value is 2.
```

AP Computer Science Principles | Unit 3

3.4 Algorithm Efficiency

Solutions

Reference scenario

Compare growth in work rather than the speed of one small run.

Algorithm A: checks each of n items once

Algorithm B: checks each pair in two full n loops

When $n = 100$: A = 100 checks; B = 10000 checks

3.4.1

For two full nested loops each running n times, how many body executions occur at $n = 20$ and $n = 40$?

At $n = 20$ there are 400 executions; at $n = 40$ there are 1600, which is four times as many.

3.4.2

Why can binary search eliminate half a sorted search range after comparing the midpoint, and why does the same argument fail on an unsorted list?

Sorted order tells which side can contain the target. Without that order, a smaller midpoint value gives no reliable information about where other values lie.

AP Computer Science Principles | Unit 3

3.5 Simulation and Randomness

Solutions

Reference scenario

Model a fair die using the inclusive AP CSP RANDOM operation.

```
roll <- RANDOM(1, 6)
success defined as roll <= 2
Possible outcomes: 1, 2, 3, 4, 5, 6
```

3.5.1

Use RANDOM(1, 10) to model a 30% chance event. Give the success condition.

Use RANDOM(1, 10) <= 3. Exactly three of the ten equally likely integer outcomes count as success.

3.5.2

A traffic simulation assumes every vehicle moves at the same speed. Explain one limit on using it to predict real congestion.

Real drivers, intersections, incidents, and vehicle behavior vary. The simplified simulation may help explore its assumptions but cannot establish accurate real-world congestion predictions without validation.