

## ► Review Questions

---

### Concepts

1. What term describes values that can only be either true or false?  
(A) Decidable  
(B) Algorithmic  
(C) Boolean  
(D) Sequential
2. What is/are used to determine whether code should be run for both "IF" statements and "REPEAT UNTIL" loops?  
(A) Pseudocode  
(B) Iterations  
(C) Conditions  
(D) Events
3. Which combination of statements can be used to express algorithms?  
(A) Iterative, sequential, and selection  
(B) Correctness, efficiency, and clarity  
(C) Readable, iterative, and efficient  
(D) Selection, conditional, and Boolean
4. What type of problem cannot currently be determined or explained by an algorithm?  
(A) Indefinite problem  
(B) Undecidable problem  
(C) Tractable problem  
(D) Unreasonable problem
5. When is a compound condition using the logical operator AND true?  
(A) When either of the conditions are true  
(B) When both conditions are false  
(C) When both conditions are true  
(D) When the NOT operator is also used
6. Which type of loop runs a set number of times?  
(A) Indefinite  
(B) REPEAT UNTIL  
(C) Infinite  
(D) REPEAT n TIMES
7. Why is a "divide and conquer" search more efficient than a linear search?  
(A) You only look at half the dataset.  
(B) You eliminate half the dataset with each iteration.  
(C) You have to search all the values.  
(D) It is less efficient with large datasets.
8. Sequential statements  
(A) run one after the other in the order given  
(B) run only when the condition is true  
(C) run until a loop finishes  
(D) run until the user enters "done"

9. Else statements
- (A) run each time an "if" condition is true
  - (B) run when an "if" condition is false
  - (C) run every time an "if" statement runs
  - (D) do not need an "if" statement to run
10. With a problem that cannot be solved for all cases, what can sometimes be used as a close approximation?
- (A) A travelling solution
  - (B) A solvable solution
  - (C) A heuristic solution
  - (D) A tractable solution
11. What are variables used for in programs?
- (A) They hold values, numbers, or strings.
  - (B) They link libraries of programs to the current program.
  - (C) They indicate how long the fraction part of a real number is.
  - (D) They hold the indices for a list name.
12. How can an individual element in a list be identified?
- (A) Use the index or number of the element's position in the list.
  - (B) Use the built-in procedures for lists.
  - (C) Use the full list name.
  - (D) Use the list name plus the value in the list at the needed position.
13. How do parameters and arguments differ?
- (A) The words can be used interchangeably.
  - (B) Parameters are sent to procedures where they are then used as arguments.
  - (C) Arguments are sent to procedures where they are then used as parameters.
  - (D) Arguments are the intermediate values in a calculation until the calculation is complete and then stored in the parameter.
14. What is a reason to use a procedure?
- (A) When you need a section of code once in a program
  - (B) To avoid duplicating code
  - (C) To avoid a loop
  - (D) To use with a condition
15. How are assignment statements processed?
- (A) The left side of the  $\leftarrow$  is processed and then assigned to the variable on the right.
  - (B) The right side of the  $\leftarrow$  is processed and then assigned to the variable on the left.
  - (C) Strings are processed first and then numbers.
  - (D) Numbers are processed first and then strings.
16. What helps manage complexity in a program by abstracting out the details and allowing programmers to use variable names?
- (A) Data abstraction
  - (B) Element abstraction
  - (C) Memory abstraction
  - (D) Procedural abstraction
17. What is string1 concatenated with string2 when:
- ```
string1  $\leftarrow$  "Tik"
string2  $\leftarrow$  "Tok"
```
- (A) Tik Tok
  - (B) "Tik" "Tok"
  - (C) Tik Tok
  - (D) "TikTok"
18. If the variable *name* has the value "Hannah", what is the value if we take the substring of *name* starting at position 4 and going to the end?
- (A) "ah"
  - (B) "Hann"
  - (C) "nana"
  - (D) "nah"
19. With Boolean, what does "A OR B" mean?
- (A) Neither A or B can be true for the condition to be true.
  - (B) If A is true, then the condition is true.
  - (C) If B is not true, then the condition is true.
  - (D) Both A and B must be true for the condition to be true.
20. Which are more abstract: high-level or low-level programming languages?
- (A) High-level
  - (B) Low-level
  - (C) Both have the same level of abstraction
  - (D) Neither are abstract

21. Procedures are abstract:
- (A) by the use of parameters
  - (B) by being easier to manage
  - (C) through reuse
  - (D) all of the above
22. How are simulations useful?
- (A) They allow the modification of multiple variables at a time to determine what changes make the most impact.
  - (B) They provide the freedom of testing all possibilities without interference.
  - (C) They can test the impact of a dangerous situation on a pilot test group of humans prior to a real-world event.
  - (D) They allow testing of hypotheses without impacting or being impacted by the real world.
23. What does an API provide?
- (A) The pseudocode for commonly used algorithms
  - (B) Documentation for how program modules in a library can be used
  - (C) The arguments and parameters for a procedure
  - (D) The program code for commonly used modules
26. The algorithm below is not working correctly. Which line of code will make it work as intended?
- (Compare cars to available parking spots)
- ```

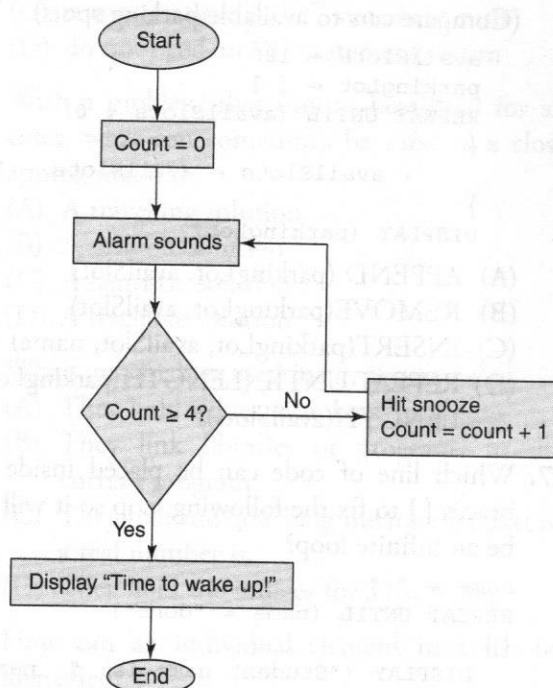
availSlots ← 180
parkingLot ← [ ]
REPEAT UNTIL (availSlots = 0)
{
    availSlots ← availSlots - 1
}
DISPLAY (parkingLot)
  
```
- (A) APPEND (parkingLot, availSlot)
  - (B) REMOVE(parkingLot, availSlot)
  - (C) INSERT(parkingLot, availSlot, name)
  - (D) REPEAT UNTIL (LENGTH(parkingLot) = LENGTH(availSlots))
27. Which line of code can be placed inside the braces { } to fix the following loop so it will not be an infinite loop?
- ```

name ← " "
REPEAT UNTIL (name = "done")
{
    DISPLAY ("Student name is: ", name)
}
  
```
- (A) IF name ≠ "done"
  - (B) count = count + 1
  - (C) name ← INPUT( )
  - (D) name ← studentRoster

### Application of Concepts

24. How can a smaller representation of something, such as an event or process, be used to determine what could happen in the real world?
- (A) Through simulations
  - (B) Through planning
  - (C) Through imaging
  - (D) Through the use of lab work
25. What is the output of the algorithm written in pseudocode below at 7:00 a.m. Friday?
- If Monday–Friday at 8:00 a.m.  
Set thermostat to 62
- If Saturday or Sunday  
Set thermostat to 70
- If time is 5:00 p.m.  
Set thermostat to 68
- (A) 62
  - (B) 68
  - (C) 70
  - (D) Unknown

28. Below is a flowchart of an alarm clock snooze process. What value does the variable "count" have after tracing through the flowchart?



- (A) 0  
(B) 4  
(C) 5  
(D) Count = count + 1
29. Are the two conditional statements equivalent?  
age > 42    NOT (age < 42)
- (A) Yes  
(B) No  
(C) Only when age is a positive number  
(D) Only when age is 0
30. What are the elements in the list "fruit" after the code below?

```

fruit ← ["grapes"]
APPEND (fruit, "bananas")
APPEND (fruit, "oranges")
APPEND (fruit, "apples")
REMOVE (fruit, 2)
INSERT (fruit, 3, "mango")
APPEND (fruit, "kiwi")
INSERT (fruit, 5, "blueberries")
REMOVE (fruit, 1)
  
```

- (A) grapes, mango, apples, blueberries, kiwi  
(B) oranges, mango, apples, blueberries, kiwi  
(C) apples, mango, kiwi, blueberries  
(D) bananas, mango, apples, kiwi, grapes

31. What is the result of executing the following code when called with LeapYear(3004)?

```

PROCEDURE LeapYear(year)
  IF (year MOD 400 = 0)
    DISPLAY (year, "is a leap year!")
  ELSE IF (year MOD 100 = 0)
    DISPLAY (year, "is not a leap year.")
  ELSE IF (year MOD 4 = 0)
    DISPLAY (year, "is a leap year!")
  ELSE
    DISPLAY ("Invalid year entered.")

  DISPLAY ("Enter a year:")
  yr ← INPUT ( )
  LeapYear(yr)
  
```

- (A) 3004 is not a leap year.  
(B) 204 is not a leap year.  
(C) 3004 is a leap year!  
(D) Invalid year entered.

32. Which block of code sets the alarm for each day of the week?

**Block 1:**

```

IF (day = "Wed")
  setAlarm ← 8
ELSE IF (day = "Sat" OR day = "Sun")
  setAlarm ← 11
ELSE
  setAlarm ← 9
  
```

**Block 2:**

```

IF (day NOT("Sat") OR day NOT("Sun"))
  setAlarm ← 9
IF (day = "Wed")
  setAlarm ← 8
  
```

- (A) Block 1  
(B) Block 2  
(C) Block 1 and Block 2  
(D) Neither Block 1 nor Block 2

33. What is returned from the procedure below after the call: weight (9, 10, 11)?

```

PROCEDURE weight(wt1, wt2, wt3)
{
  IF (wt1 ≥ wt2 AND wt1 ≥ wt3)
    RETURN wt1
  ELSE IF (wt2 > wt3 OR wt2 > wt1)
    RETURN wt2
  ELSE
    RETURN wt3
}
  
```

- (A) 9  
(B) 10  
(C) 11  
(D) 1011



