Chapter 8 Exercises and Answers

Answers are in blue.

For Exercises 1-10, indicate which structure would be a more suitable choice for each of the following applications by marking them as follows:

A. Stack
B. Queue
C. Tree
D. Binary search tree
E. Graph

1. A bank simulation of its teller operation to see how waiting times would be affected by adding another teller.
   B
2. A program to receive data that is to be saved and processed in the reverse order.
   A
3. An electronic address book ordered by name.
   D
4. A word processor to have a PF key that causes the preceding command to be redisplayed. Every time the PF key is pressed, the program is to show the command that preceded the one currently displayed.
   A
5. A dictionary of words used by a spelling checker to be built and maintained.
   D
6. A program to keep track of patients as they check into a medical clinic, assigning patients to doctors on a first-come, first-served basis.
   B
7. A program keeping track of where canned goods are located on a shelf.
   A
8. A program to keep track of the soccer teams in a city tournament.
   C
9. A program to keep track of family relationships.
   C or E
10. A program to maintain the routes in an airline.
    E

For Exercises 11 - 30, mark the answers true or false as follows:

A. True
B. False

11. A binary search cannot be applied to a tree.
    B

12. A stack and a queue are different names for the same ADT.
    B
13. A stack displays FIFO behavior.
    B
14. A queue displays LIFO behavior.
    B
15. A leaf in a tree is a node with no children.
    A
16. A binary tree is a tree in which each node can have zero, one, or two children.
    A
17. A binary search tree is another name for a binary tree.
    B
18. The value in the right child of a node (if it exists) in a binary search tree will be greater than the value in the node itself.
    A
19. The value in the left child of a node (if it exists) in a binary search tree will be greater than the value in the node itself.
    B
20. In a graph, the vertices represent the items being modeled.
    A
21. Algorithms that use a list must know whether the list is array based or linked.
    B
22. A list may be linear or nonlinear, depending on its implementation.
    B
23. The root of a tree is the node that has no ancestors.
    A
24. Binary search trees are ordered.
    A
25. On average, searching in a binary search tree is faster than searching in a list.
    B
26. On average, searching in a binary search tree is faster than searching in a linked list.
    A
27. A binary search tree is always balanced.
    B
28. Given the number of nodes and the number of levels in a binary search tree, you can determine the relative efficiency of a search in the tree.
    A
29. Insertion in a binary search tree is always into a leaf node.
    B
30. A binary search tree is another implementation of a sorted list.

A

The following algorithm (used for Exercises 31 - 33) is a count-controlled loop going from 1 through 5. At each iteration, the loop counter is either printed or put on a stack depending on the result of Boolean function **RanFun()**. (The behavior of **RanFun()** is immaterial.) At the end of the loop, the items on the stack are popped and printed. Because of the logical properties of a stack, this algorithm cannot print certain sequences of the values of the loop counter. You are given an output and asked if the algorithm could generate the output. **Respond as follows:**

A. True
B. False
C. Not enough information

```
    Set count to 0
WHILE (count < 5)
    Set count to count + 1
    IF (RanFun())
        Write count, ' '
    ELSE
        Push(myStack, count)
WHILE (NOT IsEmpty(myStack))
    Pop(myStack, number)
    Write number, ' '
```

31. The following output is possible using a stack: 1 3 5 2 4.
    B
32. The following output is possible using a stack: 1 3 5 4 2.
    A
33. The following output is possible using a stack: 1 3 5 1 3.
    B

47. Draw the binary search tree whose elements are inserted in the following order:
    50 72 96 94 107 26 12 11 9 2 10 25 51 16 17 95

insert tree like one on page 690 (11) of C++ Plus, 3rd edition. remove box and arrow as answer.

48. If Print is applied to the tree formed in Exercise 47, in which order would the elements be printed?
    2 9 10 11 12 16 17 25 26 50 51 72 94 95 96 107
49. Examine the following algorithm and apply it to the tree formed in Exercise 47. In which order would the elements be printed?

```
Print2 (tree)
IF (tree is NOT null)
    Print (right(tree))      // Recursive call R1
    Write info(tree)
    Print(left(tree))      // Recursive call R2
```

107 96 95 94 72 51 50 26 25 17 16 12 11 10 9 2

50. Examine the following algorithm and apply it to the tree formed in Exercise 47. In what order would the items be printed?

Print3 (tree)

```
IF (tree is NOT null)
    Print (right(tree))      // Recursive call R1
    Print(left(tree))        // Recursive call R2
    Write info(tree)
```

107 95 94 96 51 72 17 16 25 10 2 9 11 12 26 50

Chapter 10 Exercises and Answers

Answers are in blue.

For Exercises 1- 18, mark the answers true and false as follows:

A. True

B. False

1. An operating system is an example of application software.
   B
2. An operating system provides a basic user interface that allows the user to use the computer.
   A
3. A computer can have more than one operating system, but only one OS is in control at any given time.
   A
4. Multiprogramming is the technique of using multiple CPUs to run programs.
   B
5. In the 1960s and 70s, a human operator would organize similar computer jobs into batches to be run.
   A
6. Batch processing implies a high level of interaction between the user and the program.
   B
7. A timesharing system allows multiple users to interact with a computer at the same time.
   A
8. A dumb terminal is an I/O device that connects to a mainframe computer.
   A
9. A logical address specifies an actual location in main memory.
   B
10. An address in a single contiguous memory management system is made up of a page and an offset.
    B
11. In a fixed partition system, main memory is divided into several partitions of the same size.
    B
12. The bounds register contains the last address of a partition.
    B It contains the length of the partition.
13. The first page in a paged memory system is page 0.
    A
14. A process in the running state is currently being executed by the CPU.
    A
15. The process control block (PCB) is a data structure that stores all information about a process.
    A
16. CPU scheduling determines which programs are in memory.
    B
17. The first-com, first-served scheduling algorithm is provably optimal.
    B

18. A time slice is the amount of time each process is given before being preempted in a round robin scheduler.

    A

48. When a program is compiled, where is it assumed that the program will be loaded into memory? That is, where are logical addresses assumed to begin?

    At location 0.

49. If, in a single contiguous memory management system, the program is loaded at address 30215, compute the physical addresses (in decimal) that correspond to the following logical addresses:

    A. 9223

    39438

    B. 2302

    32517

    C. 7044

    37259

50. In a single contiguous memory management approach, if the logical address of a variable is *L* and the beginning of the application program is *A*, what is the formula for binding the logical address to the physical address?

    L + A

51. If, in a fixed partition memory management system, the current value of the base register is 42993 and the current value of the bounds register is 2031, compute the physical addresses that correspond to the following logical addresses:

    A. 104

    43097

    B. 1755

    44748

    C. 3041

    Address out of bounds of partition.

52. If more than one partition is being used (either fixed or dynamic), what does the base register contain?

    The base register contains the beginning address of the current partition.

53. Why is the logical address compared to the bounds register before a physical address is calculated?

    The bounds register contains the length of the current partition. If the logical address is greater than the bounds register, then the physical address is not within the current partition.

54. If, in a dynamic partition memory management system, the current value of the base register is 42993 and the current value of the bounds register is 2031, compute the physical addresses that correspond to the following logical addresses:

    A. 104

    43097

    B. 1755

    44748

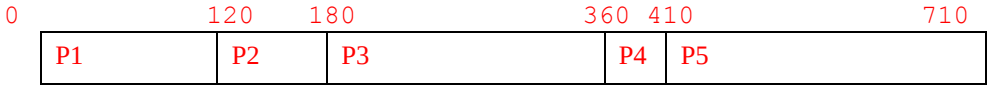
    C. 3041

    Address out of bounds of partition.

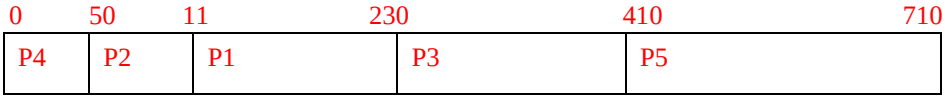
Use the following table of processes and service time for Exercises 69 through 71.

Process	P1	P2	P3	P4	P5
Service time	120	60	180	50	300

69. Draw a Gantt chart that shows the completion times for each process using first-come, first served CPU scheduling. V=356



70. Draw a Gantt chart that shows the completion times for each process using shortest-job-next CPU scheduling. V = 174



71. Draw a Gantt chart that shows the completion times for each process using round-robin CPU scheduling with a time slice of 60. V= 388

72. Distinguish between fixed partitions and dynamic partitions.

The sizes of the partitions are fixed in a fixed partition scheme, although they are not necessarily the same size. In a dynamic partition scheme, the partitions are allocated as needed.