Review questions for Chapter 7.

Answer first, then check at the end.

True or False

1. When a named code finishes executing as part of a statement in a program and processing resumes with the statement just below where the name occurred, the place where the name of the code appears is known as *the receiving unit*.

2. The following is correct pseudocode for calculating the factorial of a number (n):

Factorial(n)

```
IF (n equals 0)
    RETURN 1
ELSE
    RETURN n*Factorial(n-1)
```

3. Quicksort is an effective sorting algorithm to use if the data to be sorted is in random order.

4. An **if** statement is an example of a control structure.

5. An **event-controlled loop** repeats until an event outside of the body of the loop occurs.

6. A recursive solution must have at least two cases, a base case and a general case.

7. Recursion is the ability of a subprogram to call itself.

8. Infinite recursion occurs because there is no general case.

9. A record is a named heterogeneous collection of items in which individual items are accessed by position.

10. A record is a named heterogeneous collection of items in which individual items are accessed by name.

11. An array is a named collection of homogeneous items in which individual items are accessed by name.

12. An array is a named collection of homogeneous items in which individual items are accessed by position.

13. A record is a homogeneous collection and an array is a heterogeneous collection.

14. The approach to problem solving developed by George Polya is valuable, but does not apply to computer-based solutions.

15. An array is a homogeneous collection and a record is a heterogeneous collection.

16. An algorithm is a calculation that determines how long it will take to solve a problem.

17. An algorithm is only useful if it uses a finite amount of time and data.

18. Pseudocode uses a mixture of English and indentation to express the processing steps of an algorithm.

19. A cooking recipe is an algorithm.

20. A selection construct allows a decision to be made in an algorithm that determines which action is performed next.

21. Top-down design is a problem-solving technique in which the problem is broken down into smaller pieces until each piece is basic enough to solve directly.

22. In top-down design, the main (top) module is the most abstract.

23. In top-down design, an abstract step is one for which some details remain unspecified.

24. In top-down design, a concrete step is one that does not need to be expanded further.

25. A module cannot be further expanded.

26. Algorithms cannot be tested until they are realized in a computer-based programming language.

27. Desk checking is the process of tracing the execution of a design on paper.

28. The Rosetta stone contains the same content in five different languages.

29. A selection sort compares adjacent elements and swaps them if they are in the wrong order.

30. The Quicksort algorithm is based on a "divide and conquer" strategy.

31. The Quicksort algorithm is based on recursion.

32. The Quicksort algorithm separates the items to be sorted into two sections based on a particular split value.

33. A sequential search begins the search process in the middle of the list.

34. A binary search uses a string of bits (binary digits) to filter the search items while looking for an item.

35. A binary search eliminates large portions of the data on each comparison.

36. A binary search requires that the data be sorted.

Multiple Choice

37. Who wrote the book *How to Solve It*, which outlines a general approach to problem solving?
A. Ada Lovelace
B. Steve Jobs
C. John Vincent Atanasoff
D. George Polya
E. George Boole

38. What is the first step in the How to Solve It list?
A. establish the players
B. understand the problem
C. refine the problem
D. gather resources
E. set a timeline

39. Which of the following best describes top-down design?
A. gathering small solutions to related subproblems into a complete solution
B. refining a particular solution into one that is more abstract
C. decomposing a general solution into more concrete solutions to subproblems
D. coverting modules into subproblems
E. converting classes into objects

40. What is the practice of hiding the details of a module with the goal of controlling access to the details of the module?
A. information hiding
B. abstraction
C. data abstraction
D. procedural abstraction
E. control abstraction

41. Which of the following represents a set of unambiguous instructions for solving a problem in a finite amount of time using a finite set of data?
A. pseudocode
B. algorithm
C. program construct
D. problem specification
E. problem analysis

42. When a pilot flies a commercial airliner, what property represents his or her view of the plane?

A. information hiding
B. abstraction
C. data abstraction
D. procedural abstraction
E. encapsulation

43. When a credit card reader inputs your number, what property is being exhibited?

A. encapsulation
B. abstraction
C. data abstraction
D. procedural abstraction
E. control abstraction

44. When a credit card purchase is made, what property is being exhibited?

A. information hiding
B. abstraction
C. data abstraction
D. procedural abstraction
E. control abstraction

45. What fundamental mathematical principle underlies the effectiveness of the Quicksort algorithm to sort certain lists of data elements rapidly?

A. recursion
B. n factorial
C. information hiding
D. procedural abstraction
E. artificial intelligence

46. Which of the following produces in a search a true or false result?

A. control structure
B. data structure
C. data type
D. Boolean expression
E. information hiding

47. . Which of the following include selection statements and repetition statements?

A. control structures
B. operand specifier
C. data type
D. Boolean expression
E. information hiding

48. Which of the following is the ability for a subprogram to call itself?

A. argument
B. parameter

C. recursion
D. nested logic
E information hiding

49. Which of the following allows a WHILE loop to be contained within the body of another WHILE loop?

A. subprogram
B clear-box testing
C. recursion
D. nested logic
E. identifiers

50. Which of the following allows information to be passed into a subprogram?

A. record
B. argument
C. recursion
D. nested logic
E black-box testing

51. Which of the following requires the use of a "splitting value"?

A. selection sort
B. Quicksort
C. bubble sort
D. binary search
E. sequential search

52. Which of the following uses a "divide and conquer" approach?

A. selection sort
B. insertion sort
C. bubble sort
D. binary search
E. sequential search

53. Which of the following eliminates a large portion of the data with each comparison?

A. selection sort
B. Quicksort
C. bubble sort
D. binary search
E. sequential search

54. How many comparisons will be needed using a sequential search to find the value 69 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
14	27	29	38	42	55	57	61	64	69	77	79	84

A. 1
B. 2
C. 9
D. 10

E. 12

55. How many comparisons will be needed using a binary search to find the value 69 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
14	27	29	38	42	55	57	61	64	69	77	79	84

A. 1
B. 2
C. 9
D. 10
E. 12

56. Which of the following represents a set of unambiguous instructions for solving a problem in a finite amount of time using a finite set of data?
    A. algorithm
    B. pseudocode
    C. program construct
    D. problem specification
    E. problem analysis

Fill in the Blanks

57. _____ wrote a book called *How to Solve It*, which provides insight into the process of problem solving.

58. An _____ is a set of instructions for solving a problem in a finite amount of time using a finite amount of data.

59. A problem _____ is produced during the first phase of computer-based solutions that fully describes the problem to be solved.

60. During the _____ phase, an algorithm is translated into a programming language.

61. After a program is in use, changes made to it to enhance functionality or fix errors is called _____.

62. A(n) _____ expression is used as conditions in selection and repetition statements

.

63. During top-down design, a(n) _____ step is one that needs to be expanded further.

64. During top-down design, a(n) _____ step is one that is sufficiently detailed and needs no further expansion.

65. A(n) _____ is an instruction that determines the order in which other instructions in a program are executed.

66. _____ is the process of working through a design with a pencil and paper to test an algorithm.

67. . _____ makes the details at a lower level of design inaccessible during the design of the higher levels.

68. . _____ is the separation of the logical view of data from its implementation.

69. . _____ is the separation of the logical view of an action from its implementation.

70. A(n) _____ is the mechanism by which a value is passed into a subprogram.

71. _____ is the ability of a subprogram to call itself.

72. . The _____ of a recursive solution determines when the recursion will stop.

73. . A(n) _____ is a named heterogeneous collection of items in which individual items are accessed by name.

74. . A(n) _____ is a named homogeneous collection of items in which individual items are accessed by an index.

75. While at each iteration of a(n) _____sort, one more item is put into its permanent place.

76. A _____ search looks for an item sorted list by eliminating large portions of the data on each comparison.

77. The two basic forms of subprograms that exist are: _____ subprograms, named code that that does a particular task, and _____ subprograms, named code which also does a task but returns a single value to the calling unit.

Answer void ; value-returning (in that order)

78. The situation, known as a(n)_____, occurs when a subprogram keeps calling itself until the run-time support system runs out of memory.

Answer: infinite recursion

79. In a recursive binary search algorithm, the subprogram needs to know the _____ and the _____ indices within which it is searching.

80. The first pass of the _____ sort algorithm scans the list for the item that should be first in the list, then puts it there.

81. The _____ sort algorithm mirrors how we would sort a list of values if we were doing it by hand.

82. . The Quicksort algorithm was developed by _____.

Answer: C.A.R. (Tony) Hoare

83. The _____ sorting algorithm using a "splitting value" to divide a list of elements into two groups, then sorts those.

84. A(n) _____ search starts at one end of a list and examines each item in turn to find the item being sought.

85. A(n) _____ search starts by examining the middle element of a sorted list.

86. A(n) _____ search eliminates approximately half of the remaining search data with each comparison.

87. A(n) _____ search does not require that the list be sorted.

Short Answers

88. Name the four steps in Polya's "How to Solve It" list.

89. What are some typical questions that you should ask when faced with a problem?

90. In terms of problem solving, what are three questions that you might ask if you task is to send flowers to your mother?

91. Why is it appropriate to consider how a problem is solved by hand prior to considering how it might be solved by a computer?

92. What is an algorithm?

93. What is a repetition construct?

94. What is a selection construct?

95. Are the following directions for shampooing hair a good example of an algorithm? Explain.
    Rinse hair
    Put shampoo on hair
    Lather
    Repeat

96. Why is the following not a good example of an algorithm?
    Open car door
    Get into car
    Insert key in ignition
    Turn key
    While engine not running
        Start car
    Go to work

97. Modify the algorithm presented in the previous question to improve it.

98. What is data abstraction?

99. What is procedural abstraction?

100. What is a control structure?

101. What is an argument?

102. Distinguish between an atomic data type and a composite data type.

103. What is a homogeneous structure?

104. What is a heterogeneous structure?

105. Describe two composite data types and describe how their accessing mechanisms differ.

106. Show the state of the following array after the outer loop of the selection sort algorithm has executed two times.

[0]	[1]	[2]	[3]	[4]	[5]
19	23	2	4	99	1

107. Show the state of the following array after the outer loop of the bubble sort algorithm has executed one time.

[0]	[1]	[2]	[3]	[4]	[5]
19	23	2	4	99	1

108. Show the state of the following array after the outer loop of the selection sort algorithm has executed two times.

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

109. Show the state of the following array after the outer loop of the bubble sort algorithm has executed three times.

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

110. Show the state of the following array after the outer loop of the selection sort algorithm has executed three times.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
33	22	1	8	29	7	5	77	19	40	99

111. Show the state of the following array after the outer loop of the selection sort algorithm has executed four times.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
33	22	1	8	29	7	5	77	19	40	99

112. Show the state of the following array after the outer loop of the bubble sort algorithm has executed two times.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
33	22	1	8	29	7	5	77	19	40	99

113. Show the state of the following array after the outer loop of the bubble sort algorithm has executed three times.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
33	22	1	8	29	7	5	77	19	40	99

114. Show the state of the following array after the first recursive call is made in the Quicksort algorithm, using the value at index 0 as the split value.

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

115. Show the state of the following array after the first recursive call is made in the Quicksort algorithm, using the value at index 0 as the split value.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
33	22	1	8	99	7	5	77

116. How many comparisons does it take using a sequential search to determine that the value 8 is not in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

117. How many comparisons does it take using a sequential search to find the value 23 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

118. How many comparisons does it take using a sequential search to find the value 23 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
1	9	11	12	17	23

119. How many comparisons does it take using a sequential search to find the value 11 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
1	9	11	12	17	23

120. How many comparisons does it take using a binary search to determine that the value 8 is not in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
1	9	11	12	17	23

121. How many comparisons does it take using a binary search to find the value 9 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
1	9	11	12	17	23

Essay

122. Explain the "Look for Familiar Things" problem-solving strategy.

123. Explain the "Divide and Conquer" problem-solving strategy.

124. Relate the last step in Polya's “How To Solve It” list to the "Look for Familiar Things" strategy.

125. Polya How To Solve It list doesn't include a maintenance phase. Can you explain why?

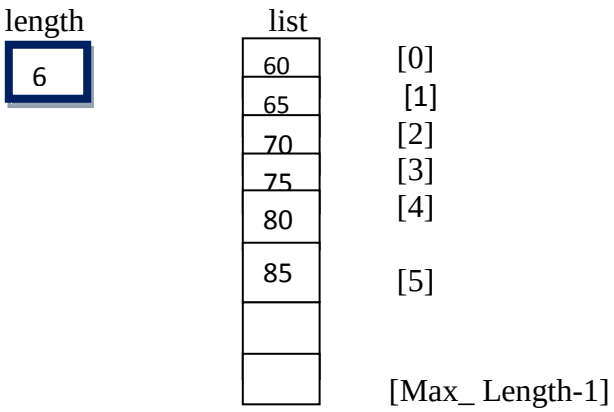
126. Why are people generally more adept at following an algorithm than developing one?

127. Compare and contrast the repetition and the selection algorithm constructs.

128. Describe the three types of abstraction that can be applied to software design.

129. Describe the strengths and weaknesses of open-source software development and discuss its viability.

130. Write an algorithm for searching a possible value in the following sorted array:



131. Describe the binary search algorithm in your own words.

Answers and solutions

True or False

1. When a named code finishes executing as part of a statement in a program and processing resumes with the statement just below where the name occurred, the place where the name of the code appears is known as *the receiving unit*.

Answer: False; it's called the *calling unit*.

2. The following is correct pseudocode for calculating the factorial of a number (n):

Factorial(n)

```
IF (n equals 0)
   RETURN 1
ELSE
   RETURN n*Factorial(n-1)
```

Answer: True

3. Quicksort is an effective sorting algorithm to use if the data to be sorted is in random order.

Answer; True

4. An **if** statement is an example of a control structure.

Answer: True

5. An event-controlled loop repeats until an event outside of the body of the loop occurs.

Answer: False

6. A recursive solution must have at least two cases, a base case and a general case.

Answer: True

7. Recursion is the ability of a subprogram to call itself.

Answer: True

8. Infinite recursion occurs because there is no general case.

Answer: False

9. A record is a named heterogeneous collection of items in which individual items are accessed by position.

Answer: False

10. A record is a named heterogeneous collection of items in which individual items are accessed by name.

Answer: True

11. An array is a named collection of homogeneous items in which individual items are accessed by name.
Answer: False

12. An array is a named collection of homogeneous items in which individual items are accessed by position.
Answer: True

13. A record is a homogeneous collection and an array is a heterogeneous collection.
Answer: False

14. The approach to problem solving developed by George Polya is valuable, but does not apply to computer-based solutions.
Answer: False

15. An array is a homogeneous collection and a record is a heterogeneous collection.
Answer: True

16. An algorithm is a calculation that determines how long it will take to solve a problem.
Answer: False; it actually solves the problem!

17. An algorithm is only useful if it uses a finite amount of time and data.
Answer: True

18. Pseudocode uses a mixture of English and indentation to express the processing steps of an algorithm.
Answer: True

19. A cooking recipe is an algorithm.
Answer: True

20. A selection construct allows a decision to be made in an algorithm that determines which action is performed next.
Answer: True

21. Top-down design is a problem-solving technique in which the problem is broken down into smaller pieces until each piece is basic enough to solve directly.
Answer: True

22. In top-down design, the main module is the most abstract.
Answer: True

23. In top-down design, an abstract step is one for which some details remain unspecified.
Answer: True

24. In top-down design, a concrete step is one that does not need to be expanded further.
Answer: True

25. A module cannot be further expanded.
Answer: False

26. Algorithms cannot be tested until they are realized in a computer-based programming language.
Answer: False. We can check algorithms with pencil and paper – it's called **desk checking**. Alan Turing ran chess-playing algorithms this way before the first computer was built!

27. Desk checking is the process of tracing the execution of a design on paper.
Answer: True

28. The Rosetta stone contains the same content in five different languages.
Answer: False. Only three scripts are used: the upper text is Ancient Egyptian hieroglyphs, the middle portion Demotic script, and the lowest Ancient Greek.

29. A selection sort compares adjacent elements and swaps them if they are in the wrong order.
Answer: False

30. The Quicksort algorithm is based on a "divide and conquer" strategy.
Answer: True

31. The Quicksort algorithm is based on recursion.
Answer: True

32. The Quicksort algorithm separates the items to be sorted into two sections based on a particular split value.
Answer: True

33. A sequential search begins the search process in the middle of the list.
Answer: False

34. A binary search uses a string of bits (binary digits) to filter the search items while looking for an item.
Answer: False

35. . A binary search eliminates large portions of the data on each comparison.
Answer: True

36. A binary search requires that the data be sorted.
Answer: True

Multiple Choice

37. Who wrote the book *How to Solve It*, which outlines a general approach to problem solving?

A. Ada Lovelace
B. Steve Jobs
C. John Vincent Atanasoff
D. George Polya
E. George Boole

Answer: D

38. What is the first step in the How to Solve It list?

A. establish the players
B. understand the problem
C. refine the problem
D. gather resources
E. set a timeline

Answer: B

39. Which of the following best describes top-down design?

A. gathering small solutions to related subproblems into a complete solution
B. refining a particular solution into one that is more abstract
C. decomposing a general solution into more concrete solutions to subproblems
D. coverting modules into subproblems
E. converting classes into objects

Answer: C

40. What is the practice of hiding the details of a module with the goal of controlling access to the details of the module?

A. information hiding
B. abstraction
C. data abstraction
D. procedural abstraction
E. control abstraction

Answer: A

41. Which of the following represents a set of unambiguous instructions for solving a problem in a finite amount of time using a finite set of data?

A. pseudocode
B. algorithm
C. program construct
D. problem specification
E. problem analysis

Answer: B

42. When a pilot flies a commercial airliner, what property represents his or her view of the plane?

A. information hiding
B. abstraction
C. data abstraction
D. procedural abstraction
E. encapsulation

Answer: D

43. When a credit card reader inputs your number, what property is being exhibited?

A. encapsulation
B. abstraction
C. data abstraction
D. procedural abstraction
E. control abstraction

Answer: C

44. When a credit card purchase is made, what property is being exhibited?

A. information hiding
B. abstraction
C. data abstraction
D. procedural abstraction
E. control abstraction

Answer: D

45. What fundamental mathematical principle underlies the effectiveness of the Quicksort algorithm to sort certain lists of data elements rapidly?

A. recursion
B. n factorial
C. information hiding
D. procedural abstraction
E. artificial intelligence

Answer: A

46. Which of the following produces in a search a true or false result?
A. control structure
B. data structure
C. data type
D. Boolean expression
E. information hiding

Answer: D

47. . Which of the following include selection statements and repetition statements?
A. control structures
B. operand specifier
C. data type
D. Boolean expression
E. information hiding

Answer: A

48. Which of the following is the ability for a subprogram to call itself?
A. argument
B. parameter
C. recursion
D. nested logic
E information hiding

Answer: C

49. Which of the following allows a WHILE loop to be contained within the body of another WHILE loop?
A. subprogram
B clear-box testing
C. recursion
D. nested logic
E. identifiers

Answer: D

50. Which of the following allows information to be passed into a subprogram?

A. record
B. argument
C. recursion
D. nested logic
E black-box testing

Answer: B

51. Which of the following requires the use of a "splitting value"?

A. selection sort
B. Quicksort
C. bubble sort
D. binary search
E. sequential search

Answer: B

52. Which of the following uses a "divide and conquer" approach?

A. selection sort
B. insertion sort
C. bubble sort
D. binary search
E. sequential search

Answer: D

53. Which of the following eliminates a large portion of the data with each comparison?

A. selection sort
B. Quicksort
C. bubble sort
D. binary search
E. sequential search

Answer: D

54. How many comparisons will be needed using a sequential search to find the value 69 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
14	27	29	38	42	55	57	61	64	69	77	79	84

A. 1
B. 2

C. 9
D. 10
E. 12

Answer: D

55. How many comparisons will be needed using a binary search to find the value 69 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
14	27	29	38	42	55	57	61	64	69	77	79	84

A. 1
B. 2
C. 9
D. 10
E. 12

Answer: B

56. Which of the following represents a set of unambiguous instructions for solving a problem in a finite amount of time using a finite set of data?
    A. algorithm
    B. pseudocode
    C. program construct
    D. problem specification
    E. problem analysis

Answer: A

Fill in the Blank

57. _____ wrote a book called *How to Solve It*, which provides insight into the process of problem solving.

Answer: George Polya

58. An _____ is a set of instructions for solving a problem in a finite amount of time using a finite amount of data.

Answer: algorithm

59. A problem _____ is produced during the first phase of computer-based solutions that fully describes the problem to be solved.

Answer: specification

60. During the _____ phase, an algorithm is translated into a programming language.

Answer: implementation

61. After a program is in use, changes made to it to enhance functionality or fix errors is called _____.

Answer: maintenance

62. A(n) _____ expression is used as conditions in selection and repetition statements

    Answer: Boolean

.

63. During top-down design, a(n) _____ step is one that needs to be expanded further.

Answer: abstract

64. During top-down design, a(n) _____ step is one that is sufficiently detailed and needs no further expansion.

Answer: concrete

65. A(n) _____ is an instruction that determines the order in which other instructions in a program are executed.

Answer: control structure

66. _____ is the process of working through a design with a pencil and paper to test an algorithm.

Answer: Desk checking

67. _____ makes the details at a lower level of design inaccessible during the design of the higher levels.
Answer: Information hiding

68. _____ is the separation of the logical view of data from its implementation.
Answer: Data abstraction

69. . _____ is the separation of the logical view of an action from its implementation.
Answer: Procedural abstraction

70. A(n) _____ is the mechanism by which a value is passed into a subprogram.
Answer: argument or parameter

71. _____ is the ability of a subprogram to call itself.
Answer: Recursion

72. . The _____ of a recursive solution determines when the recursion will stop.
Answer: base case

73. . A(n) _____ is a named heterogeneous collection of items in which individual items are accessed by name.
Answer: Record

74. . A(n) _____ is a named homogeneous collection of items in which individual items are accessed by an index.
Answer: array

75. While at each iteration of a(n) _____sort, one more item is put into its permanent place.
Answer: selection

76. A _____ search looks for an item sorted list by eliminating large portions of the data on each comparison.
Answer: binary

77. The two basic forms of subprograms that exist are: _____ subprograms, named code that that does a particular task, and _____ subprograms, named code which also does a task but returns a single value to the calling unit.
Answer void ; value-returning (in that order)

78. The situation, known as a(n)_____, occurs when a subprogram keeps calling itself until the run-time support system runs out of memory.
Answer: infinite recursion

79. In a recursive binary search algorithm, the subprogram needs to know the _____

and the _____ indices within which it is searching.

Answer: first; last

80. The first pass of the _____ sort algorithm scans the list for the item that should be first in the list, then puts it there.

Answer: selection

81. The _____ sort algorithm mirrors how we would sort a list of values if we were doing it by hand.

Answer: selection

82. . The Quicksort algorithm was developed by _____.

Answer: C.A.R. (Tony) Hoare

83. The _____ sorting algorithm using a "splitting value" to divide a list of elements into two groups, then sorts those.

Answer: Quicksort

84. A(n) _____ search starts at one end of a list and examines each item in turn to find the item being sought.

Answer: sequential

85. A(n) _____ search starts by examining the middle element of a sorted list.

Answer: binary

86. A(n) _____ search eliminates approximately half of the remaining search data with each comparison.

Answer: binary

87. A(n) _____ search does not require that the list be sorted.

Answer: sequential

Short Answer

88. Name the four steps in Polya's “How to Solve It” list.

Answer: Understand the problem, devise a plan, carry out the plan, and examine the solution.

89. What are some typical questions that you should ask when faced with a problem?

Answer: What do I know about the problem? What is the information that I have to process in order to find the solution? What does the solution look like? What sort of special cases exist? How will I recognize that I have found the solution?

90. In terms of problem solving, what are three questions that you might ask if you task is to send flowers to your mother?

Answer: What flowers does my Mother like? What flowers are in bloom? Where is the nearest florist?

91. Why is it appropriate to consider how a problem is solved by hand prior to considering how it might be solved by a computer?

Answer: If we do not understand how a task could be done by hand, we do not understand it well enough to begin designing a computer-based solution.

92. What is an algorithm?

Answer: A set of unambiguous instructions for solving a problem in a finite amount of time using a finite amount of data.

93. What is a repetition construct?

Answer: It allows an action or set of actions to be repeated multiple times as long as some condition is true.

94. What is a selection construct?

Answer: It allows the algorithm to determine whether an action (or set of actions) is executed or not, or to choose between two actions.

95. Are the following directions for shampooing hair a good example of an algorithm? Explain.
    Rinse hair
    Put shampoo on hair
    Lather
    Repeat

Answer: No. There is no limit on the number of repetitions.

96. Why is the following not a good example of an algorithm?
    Open car door
    Get into car
    Insert key in ignition
    Turn key
    While engine not running
        Start car
    Go to work

Answer: It contains an infinite loop if the car won't start.

97. Modify the algorithm presented in the previous question (#99) to improve it.

Answer:

```
Open car door
Get into car
Insert key in ignition
Turn key
Set running to false
Set tried to 0
While NOT running AND tryAgain
        Start car
        If engine running
                Set running to true
        Else
                Set tried to tried + 1
                Set tryAgain to tried < 3
If NOT running
        Call garage
Else
        Go to work
```

98. What is data abstraction?

Answer: The separation of the logical view of data from its implementation details

99. What is procedural abstraction?

Answer: The separation of the logical view of an action from its implementation

100. What is a control structure?

Answer: An instruction that determines the order in which other instructions in a program are executed.

101. What is an argument?

Answer: A value passed into a subprogram when the subprogram is invoked.

102. Distinguish between an atomic data type and a composite data type.

Answer: An atomic data type cannot be divided into parts. A composite data type is a named collection of items in which the individual items can be accessed.

103. What is a homogeneous structure?

Answer: One in which all the individual parts must have the same data type.

104. What is a heterogeneous structure?

Answer: One in which all the individual parts do not have to be of the same data type.

105. Describe two composite data types and describe how their accessing mechanisms differ.

Answer: A record is a heterogeneous data type in which the individual items are accessed by name.

An array is a homogeneous data type in which the individual items are accessed by their place in the structure.

106. Show the state of the following array after the outer loop of the selection sort algorithm has executed two times.

[0]	[1]	[2]	[3]	[4]	[5]
19	23	2	4	99	1

Answer:

[0]	[1]	[2]	[3]	[4]	[5]
1	2	23	4	99	19

107. Show the state of the following array after the outer loop of the bubble sort algorithm has executed one time.

[0]	[1]	[2]	[3]	[4]	[5]
19	23	2	4	99	1

Answer:

[0]	[1]	[2]	[3]	[4]	[5]
1	19	23	2	4	99

108. Show the state of the following array after the outer loop of the selection sort algorithm has executed two times.

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

Answer:

[0]	[1]	[2]	[3]	[4]	[5]
1	9	12	23	17	11

109. Show the state of the following array after the outer loop of the bubble sort algorithm has executed three times.

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

Answer:

[0]	[1]	[2]	[3]	[4]	[5]
1	9	11	12	17	23

110. Show the state of the following array after the outer loop of the selection sort algorithm has executed three times.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
33	22	1	8	29	7	5	77	19	40	99

Answer:

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
1	5	7	8	29	33	22	77	19	40	99

111. Show the state of the following array after the outer loop of the selection sort algorithm has executed four times.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
33	22	1	8	29	7	5	77	19	40	99

Answer:

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
1	5	7	8	29	33	22	77	19	40	99

112. . Show the state of the following array after the outer loop of the bubble sort algorithm has executed two times.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
33	22	1	8	29	7	5	77	19	40	99

Answer:

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
1	5	33	22	7	8	29	19	40	77	99

113. Show the state of the following array after the outer loop of the bubble sort algorithm has executed three times.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
33	22	1	8	29	7	5	77	19	40	99

Answer:

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
1	5	7	33	22	8	29	77	19	40	99

114. Show the state of the following array after the first recursive call is made in the Quicksort algorithm, using the value at index 0 as the split value.

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

Answer:

[0]	[1]	[2]	[3]	[4]	[5]
11	1	9	12	17	23

115. Show the state of the following array after the first recursive call is made in the Quicksort algorithm, using the value at index 0 as the split value.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
33	22	1	8	99	7	5	77

Answer:

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
7	22	1	8	5	33	99	77

116. How many comparisons does it take using a sequential search to determine that the value 8 is not in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

Answer: 6

117. How many comparisons does it take using a sequential search to find the value 23 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
12	1	9	23	17	11

Answer: 4

118. How many comparisons does it take using a sequential search to find the value 23 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
1	9	11	12	17	23

Answer: 6

119. How many comparisons does it take using a sequential search to find the value 11 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
1	9	11	12	17	23

Answer: 3

120. How many comparisons does it take using a binary search to determine that the value 8 is not in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
1	9	11	12	17	23

Answer: 3

121. How many comparisons does it take using a binary search to find the value 9 in the following array?

[0]	[1]	[2]	[3]	[4]	[5]
1	9	11	12	17	23

Answer: 3

Essay

122. Explain the "Look for Familiar Things" problem-solving strategy.

Answer: Many different problems have similar parts and the same problem can often be expressed in different ways. Part of understanding the problem involves examining the problem for subproblems that you have seen before. If a solution exists for the problem or subproblems, use it. If a solution exists for a similar problem, adapt it. This strategy can be summed up in the expression "never reinvent the wheel."

123. Explain the "Divide and Conquer" problem-solving strategy.

Answer: Divide a large problem into a collection of smaller problems. The collection of solutions to

the smaller problems solves the large problem. If one of the smaller problems is still to large to solve easily, break it into a collection of smaller problems and repeat the process.

124. Relate the last step in Polya's “How To Solve It” list to the "Look for Familiar Things" strategy.

Answer: The last step is to examine the solution for correctness and possible reuse. When examining a new problem, look for parts of the problem that are familiar. If you find one, look to see how you solved it before and use the solution.

125. Polya How To Solve It list doesn't include a maintenance phase. Can you explain why?

Answer: Polya addressed general problem solving. The maintenance phase in computer problem solving involves maintaining an implementation of a solution over time and changing circumstances. Polya's process stopped when the solution was implemented once. Thus, Polya did not envision any maintenance of the solution.

126. Why are people generally more adept at following an algorithm than developing one?

Answer: We follow algorithms all the time in our daily lives. We follow recipes; we follow instruction while putting together anything from a toy to a doghouse; we follow directions to an address. Few of us develop recipes, or write instructions on how to put something together. We have all probably given verbal directions on how to find a specific location, but not often enough to be good at giving unambiguous ones. Following directions (algorithms) is an everyday occurrence; creating them is not.

127. Compare and contrast the repetition and the selection algorithm constructs.

Answer: Both constructs make a decision, based on a condition, which determines what happens next. In a repetition construct, the decision determines whether a particular set of actions is performed again, and allows the actions to be performed multiple times. The decision of a selection construct determines if an action is going to be executed or skipped. An alternative version allows a choice between one action (or set of actions) and another.

128. Describe the three types of abstraction that can be applied to software design.

Answer: Data abstraction, as the name implies, focuses on the data managed by a program. It

separates the logical view of the data from its implemention at the program level. Procedural abstraction creates a similar separation between the logical view and implementation details of the actions that occur in a program. Control abstraction separates the logical view of a control structure from its actual implementation in a particular programming language.

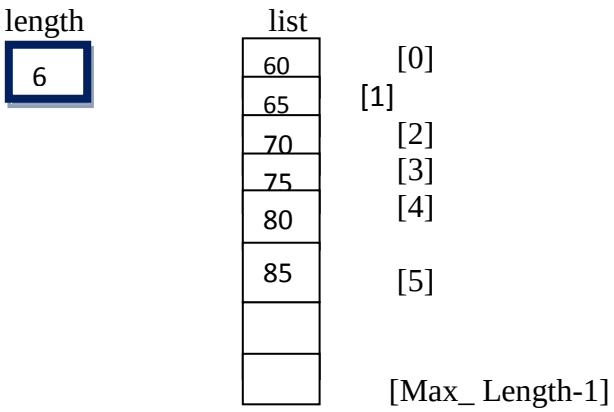
129. Describe the strengths and weaknesses of open-source software development and discuss its viability.

Answer: In an open-source software development project, the source code is freely available to anyone who wants to look at it, modify it, and even sell it. The only restriction is that any redistribution be under the same terms. Proponents of open-source software argue that such products benefit from the rigorous scrutiny anyone can apply and from the ability of anyone to make improvements to the product, thus, enhancing the reliability of such open-source software products. Open-source advocates also illustrate the cost-effective elements of the model by pointing out that open source licensing agreements do not force users to utilize indefinitely the open source software these user might initially purchase.

Open-source opponents argue that it destroys the ability of software companies to capitalize financially on the hard work, problem solving efforts, and creativity devoted to a project. Moreover, these opponents assert that the security vulnerabilities of open-source software would leave confidential and national security government data, medical information, and critical transportation systems much more vulnerable to security risks from hackers and terrorists.

The success and adoption of the Linux operating system by businesses, educational institutions, and even some government agencies strongly suggests that the open-source model may be much more commercially viable than proprietary software companies believe to be the case.

130. Write an algorithm for searching a possible value in the following sorted array:



Answer:

```
Set index to 0
Set found to FALSE
WHILE (index < length AND NOT found)
        IF (data[index] equals searchItem)
                Set found to TRUE
        ELSE IF (data[index] > searchItem)
                Set index to length
        ELSE
                Set index to index + 1
```

131. Describe the binary search algorithm in your own words.

Answer: It either locates the item or eliminates half of the sorted array with one comparison. Instead of searching for the item beginning of the array and progressing sequentially, the binary algorithm starts at the middle of the array. If the item for which one is searching is less than the item in the middle, one knows that the item will not be in the second half of the array. One then continues by searching the data in the first half of the array. If the item for which one is searching is greater than the middle, one continues searching between the middle and the end of the array. If the middle item is equal to the one for which you are searching, the search stops. The process continues, in this way, with each comparison dividing in half where the item might be. The binary search stops when the item is found or when the portion of the array where the item could be is empty.