Name:_____ Class:_____ Date:_____

Chapter 04: Arrays and Linked Structures

Reference and copyright:

Author	Title	Publisher	Date	ISBN
Kenneth Lambert	Fundamentals of Python: Data Structures	Cengage Learning; 2nd Edition	October 11, 2018	ISBN-10 : 0357122755 ISBN-13 : 978-0357122754

1. The list is the primary implementing structure in Python collections.
   a. True
   b. False

2. When an array object is traversed in a *for* loop, the object's __*iter*__ method is called.
   a. True
   b. False

3. When an array is instantiated, it is filled with zeros by default.
   a. True
   b. False

4. If an array's logical size is greater than zero, the index of the last item in the array is the logical size plus 1.
   a. True
   b. False

5. If the logical size of an array equals the physical size of the array, a new array must be created to add elements to the array.
   a. True
   b. False

6. Time performance is improved if you double the size of the array when you need to increase its size.
   a. True
   b. False

7. When a list's *append* method results in memory wasted beyond a threshold, the size of the underlying array is decreased.

Chapter 04: Arrays and Linked Structures

a. True
b. False

8. When an item is inserted into an array, the logical size of the array increases.
a. True
b. False

9. To access a two-dimensional array, you use two subscripts.
a. True
b. False

10. A ragged grid has a fixed number of columns and a variable number of rows.
a. True
b. False

11. A linked structure can have items that have a maximum of one link to another item.
a. True
b. False

12. The first item in a singly linked structure is accessed using a head link.
a. True
b. False

13. It's easier to get to an item's predecessor in a singly linked structure than in a doubly linked structure.
a. True
b. False

14. In a doubly linked structure, the first and last item have an empty link.
a. True
b. False

15. A linked structure can be stored in noncontiguous memory.
a. True
b. False

Chapter 04: Arrays and Linked Structures

16. In Python, a node in a doubly linked structure contains two fields: a data item and a reference to the next node.
    a. True
    b. False

17. On a linked structure, index-based operations must be emulated by manipulating links within the structure.
    a. True
    b. False

18. To start a traversal of a linked structure, initialize a variable to the structure's head pointer.
    a. True
    b. False

19. A traversal of a singly linked structure terminates when the temporary variable is equal to the head pointer.
    a. True
    b. False

20. Similar to an array, linked structures support random access.
    a. True
    b. False

21. Inserting data at the beginning of a linked structure uses constant time and memory.
    a. True
    b. False

22. The operation of removing an item at the end of a linked structure is constant in time and memory.
    a. True
    b. False

23. The insertion and removal of the first node of a singly linked structure require that the head pointer be reset.
    a. True
    b. False

Chapter 04: Arrays and Linked Structures

24. A circular linked structure contains a link from the last node back to the first node in the structure.
    a. True
    b. False

25. The run-time complexities of the operations on a doubly linked structure are typically double compared to the corresponding operations on the singly linked structure.
    a. True
    b. False

26. Which of the following best describes an array?
    a. a collection of data points that represent an object
    b. a list of values that are indexes to a database
    c. a numeric value that points to a position in RAM where data can be found
    d. a sequence of items that can be accessed at given index positions

27. What is the primary implementing structure of Python collections?
    a. list
    b. array
    c. linked list
    d. dictionary

28. Which of the following is true about Python's array module?
    a. it is limited to storing numbers
    b. it behaves much like a dictionary
    c. it can only hold character values
    d. you can define its size at run time

29. In the Array class defined in Chapter 4, how do you instantiate an array object that can hold 10 values?
    a. myArray(10) = Array
    b. Array myArray, 10
    c. myArray = Array(10)
    d. Array(10) myArray

30. Older programming languages implement arrays as static data structures which are inefficient. What do modern languages use to remedy the problems of static arrays?

Chapter 04: Arrays and Linked Structures

a. dynamic arrays
b. linked lists
c. data stacks
d. hash tables

31. What method does Python's list type use to increase the size of the underlying array?
a. size
b. append
c. increase
d. augment

32. The process for resizing an array named myArray is shown below. What is the missing code?

```
if logicalSize == len(myArray):
    temp = Array(len(myArray) + 1)
    for i in range(logicalSize):
        <missing code>
    a = temp
```

a. myArray[temp] = myArray[i]
b. temp [i] = myArray[i]
c. myArray[i] = temp[i]
d. temp = myArray(len(myArray))

33. What process is required to avoid wasting memory if successive calls to the pop method on a list are made?
a. delete the array
b. grow the array
c. reset the array
d. shrink the array

34. In the following code to insert an item in an array, what is the missing code?

```
for x in range(logicalSize, targetIndex, -1):
    myArray[x] = myArray[x - 1]
a[targetIndex] = newItem

<missing code>
```

a. targetIndex += 1

Chapter 04: Arrays and Linked Structures

b. targetIndex -= 1
c. logicalSize += 1
d. logicalSize -= 1

35. Which of the following statements accesses the second column in the third row of a two-dimensional array?
a. twoDim[2][1]
b. twoDim[4][3]
c. twoDim[1][2]
d. twoDim[2][3]

36. The following code sums all the values in the two-dimensional array. What is the missing code?

```
sum = 0
for row in range(grid.getHeight()):
    for column in range(grid.getWidth()):
        <missing code>
```

a. sum += grid[column][row]
b. sum += grid[row-1][column-1]
c. sum += grid[column+1][row+1]
d. sum += grid[row][column]

37. How does a programmer access the first item in a singly linked structure?
a. by using the 0 index
b. with the first() method
c. by following a head link
d. by a call to getLink(1)

38. Which of the following is an advantage of a doubly linked structure over a singly linked structure?
a. it is less complex to implement
b. you can easily access the predecessor of an item
c. you can easily access the successor of an item
d. it uses less memory

39. What does the last item in a singly linked structure contain?
a. an empty link

Chapter 04: Arrays and Linked Structures

b. a link to the first item
c. a link to the previous item
d. a method for appending an item

40. What type of memory scheme does a linked list use?
a. overlapping
b. noncontiguous
c. sequential
d. random

41. Which statement tests if a singly linked node variable named *myItem* has been initialized?
a. if myItem is Null:
b. if myItem != None:
c. if myItem = None:
d. if myItem is not Null:

42. What are almost all operations on arrays based upon?
a. hashes
b. keys
c. links
d. indexes

43. What is the operation on a linked structure called that visits each node without deleting it?
a. probe
b. insertion
c. removal
d. traversal

44. What type of linked structure operation is the following code performing?

```
z = 0
probe = head
while probe != None:
    z = probe.data + z
    probe = probe.next
```

a. traversal
b. initialization

Chapter 04: Arrays and Linked Structures

c. visit with removal
d. insertion

45. The following code searches a linked structure. What is the missing code?

```
probe = head
while probe != None and targetItem != probe.data:
    <missing code>
if probe != None:
    print("Target item found!")
else:
    print("Target item not found!")
```

a. probe.data = next.data
b. probe.next = targetItem.data
c. probe = probe.next
d. probe.head = probe.next

46. On average, what is the performance of a sequential search on a singly linked structure?
a. logarithmic
b. linear
c. exponential
d. random

47. What type of operation is the following code performing?

```
probe = head
while probe != None and targetItem != probe.data:
    probe = probe.next
if probe != None:
    probe.data = newItem
    return True
else:
    return False
```

a. sum of all items
b. replacement
c. insertion
d. deletion

48. What action does the following code perform assuming the Node class defined in Chapter 4?

Chapter 04: Arrays and Linked Structures

```
head = Node(newItem, head)
```

a. deletion of an item in a linked list
b. appending an item to the end of a linked list
c. replacing an item at the beginning of a linked list
d. insertion of an item at the beginning of a linked list

49. Why are the insertion and removal of the first node special cases of the insert and remove *i*th node on singly linked structures?

a. the tail pointer must be reset
b. the first item must be deleted
c. the head pointer must be reset
d. the last item must be deleted

50. How does the class definition of a doubly linked structure differ from a singly linked structure?

a. by adding a *previous* pointer
b. by adding another *head* pointer
c. by adding an extra data field
d. by removing the *tail* pointer