M7.C1: Exam: Midterm Exam

Due No due date **Points** 30 **Questions** 30 **Time Limit** 60 Minutes

Instructions

The Midterm Exam covers Modules 1–6. It has 30 multiple-choice questions. You have 60 minutes to take the exam. The exam is open book, open notes. Collaboration or communication with others is not permitted.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	35 minutes	29 out of 30

Score for this quiz: **29** out of 30
Submitted Mar 8 at 1:17pm
This attempt took 35 minutes.

Question 1

1 / 1 pts

Which is an abstract data type (ADT)?

Correct!

- (selected) A list
- A string
- A float
- A boolean

Question 2

1 / 1 pts

Which of the following statements is correct?

- The queue ADT supports the removing of items from one end and the adding of items to the other end but the list ADT does not.

Correct!

- The list ADT supports the printing of the list contents but the queue ADT does not.
- The queue ADT supports the insertion and deletion of items at the front and the back.
- The queue ADT supports the printing of the list contents but the list ADT does not.

Question 3

1 / 1 pts

Which abstract data type (ADT) is best suited to store the names of all currently available smartphone models?

Correct!

- A set

- An array
- A linked list
- A stack

Question 4

1 / 1 pts

Which of the following is correct for a list ADT?

- A list can be implemented in a programming language only using the LinkedList ADT.
- A list's behavior is similar to that of a queue.
- Correct! An element can be found and removed from the end of the list.
- A list can print or remove an element only from the beginning of the list.

Question 5

1 / 1 pts

Which is the correct code to prepend an item to a list?

Correct!

```
ListPrepend(list, newNode) {
   if (list⇢head == null) {
      list⇢head = newNode
      list⇢tail = newNode
   }
   else {
      newNode⇢next = list⇢head
      list⇢head = newNode
   }
}
```

```
ListPrepend(list, newNode) {
   if (list⇢head == null) {
      list⇢head = newNode
      list⇢tail = newNode
   }
   else {
      list⇢head⇢next = newNode
      list⇢head = newNode
   }
}
```

```
ListPrepend(list, newNode) {
   if (list⇢head == null) {
      list⇢head = newNode
      list⇢tail = newNode
   }
   else {
      newNode⇢next = list⇢tail
      list⇢tail = newNode
   }
}
```

```
ListPrepend(list, newNode) {
   if (list⇢head == null) {
      list⇢head = newNode
      list⇢tail = newNode
   }
   else {
      list⇢tail⇢next = newNode
      list⇢tail = newNode
   }
}
```

Question 6

1 / 1 pts

Which XXX will complete the algorithm to separate numberList into two lists (even and odd) using an array?

```
MathematicalFunction(numberList) {


   Create evenNumberList array
   Create oddNumberList array

   for (i = 0; i < numberList⇢length; ++i) {
      XXX {
         ArrayAppend(evenNumberList, numberList[i])
      }
      else {
         ArrayAppend(oddNumberList,numberList[i])
      }
      SortAscending(evenNumberList)
      SortAscending(oddNumberList)
   }


   for (i = 0; i < evenNumberList⇢length; ++i) {
      Display evenNumberList[i]
   }
   for (i = 0; i < oddNumberList⇢length; ++i) {
      Display oddNumberList[i]
   }
}
```

- if (numberList[i] % 1 == 0)
- Correct! if (numberList[i] % 2 == 0)
- if (numberList[i] % 2 == 1)
- if (numberList[i] % 1 == 1)

Question 7

1 / 1 pts

What is the space complexity of the algorithm?

```
ArithmeticSeries(list, listSize) {
   i = 0
   arithmeticSum = 0
   while (i < listSize) {
      arithmeticSum = arithmeticSum + list[i]
      i = i + 1
   }
   return arithmeticSum
}
```

Correct!

- S(N) = N + k
- S(N) = k
- S(N) = N
- S(N) = N * k

Question 8

1 / 1 pts

The upper bound of an algorithm with best case runtime $T(N) = 3N + 16$ and worst case runtime $T(N) = 4N^2 + 10N + 5$ is _____.

- $7N$
- $7N^2$
- $4N^2 + 10N$

Correct!

- $19N^2$

Question 9

1 / 1 pts

The Big O notation of the algorithm $7 + 12N + 3N^2$ is _____.

- $12N$
- $3N^2$
- 7
- Correct! N^2

Question 10

1 / 1 pts

The lower bound of an algorithm's runtime complexity is _____.

- ≥ the best case
- Correct! ≤ the best case
- ≤ the worst case
- ≥ the worst case

Question 11

1 / 1 pts

If a queue is implemented as a linked list, a dequeue removes _____ node.

Correct!

- the head
- a random
- the middle
- the tail

Question 12

1 / 1 pts

Given a Queue implemented as a linked list, identify the correct dequeue algorithm.

```
QueueDequeue(queue) {
   tailData = queue⇢tail⇢data
   ListRemoveAfter(queue, queue⇢tail)
   return tailData
}
```

```
QueueDequeue(queue) {
   headData = queue⇢head⇢data
   ListRemoveAfter(queue, queue⇢head)
   return headData
}
```

Correct!

```
QueueDequeue(queue) {
   headData = queue⇢head⇢data
   ListRemoveAfter(queue, null)
   return headData
}
```

```
QueueDequeue(queue) {
   tailData = queue⇢tail⇢data
   ListRemoveAfter(queue, null)
   return tailData
}
```

Question 13

1 / 1 pts

The following algorithm is an example of _____.

```
def MyMath(num)
    if num <= 1
        return num
    return MyMath(num - 2) + MyMath(num - 1)
```

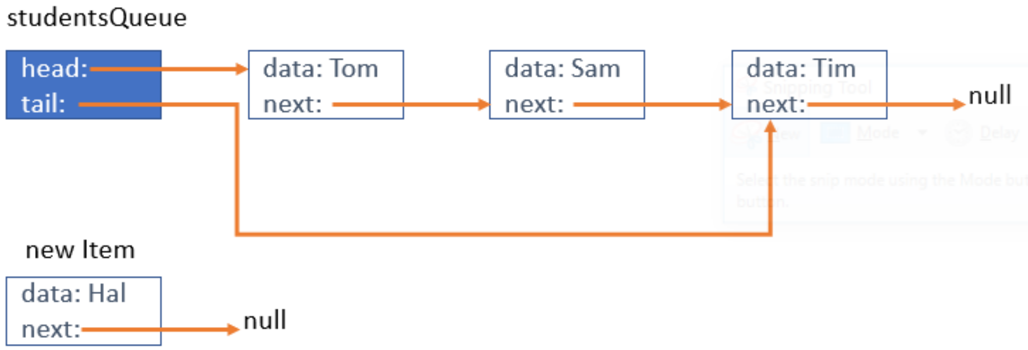
Correct!

- an exponential runtime complexity
- a constant runtime complexity
- a logarithmic runtime complexity
- a linear runtime complexity

Question 14

0 / 1 pts

Given the following queue, which pointer points to the new item when the operation QueueEnqueue(studentsQueue, "Hal") is executed?



ou Answered

- Node Tom’s next pointer
- The studentsQueue head pointer
- The null pointer

orrect Answer

- Node Tim’s next pointer

Question 15

1 / 1 pts

Given the deque 10, 12, 14, 16 (front is 10), what will the deque look like after the following operations? push-front 2,
push-back 13,
push-front 8,
pop-front,
pop-back,
push-back 18

- 8, 10, 12, 14, 16, 18
- 8, 10, 12, 14, 16, 13, 18
- 2, 10, 12, 14, 16, 13, 18

Correct!

- 2, 10, 12, 14, 16, 18

Question 16

1 / 1 pts

Given the deque 10, 12, 14, 16 (front is 10), what is the outcome of the following operations? print(IsEmpty(deque))
print(PeekFront(deque))
print(PeekBack(deque))
print(GetLength(deque))

- false, 10, 16, true
- false, 10, 16, 2

Correct!

- false, 10, 16, 4
- false, 10, 16, true

Question 17

1 / 1 pts

Given a stack myData: 34, 78 (top is 34), what is the output after the following operations? Peek(myData)
Push(myData, 2)
Push(myData, 15)
Pop(myData)
Pop(myData)
print(IsEmpty(myData))

- 34 true
- 78 false
- Correct! false
- true

Question 18

1 / 1 pts

If a stack is implemented as a linked list, then an empty stack will have _____.

- a null head and non-null tail pointer
- non-null head and tail pointers
- a non-null head and null tail pointer
- Correct! null head and tail pointers

Question 19

1 / 1 pts

If a stack is implemented as a linked list, then a push will be a(n) ______ operation.

- replace
- Correct! prepend
- insert
- append

Question 20

1 / 1 pts

Given an empty stack menuItems, what will be the result of the following operations? StackPush(menuItems , item Pizza)
StackPush(menuItems , item Chips)
StackPush(menuItems , item Nachos)
StackPush(menuItems , item Tea)
print(StackPop(menuItems))

- Correct! Tea
- Tea Nachos Chips Pizza
- Pizza
- Pizza Chips Nachos Tea

Question 21

1 / 1 pts

If a stack is implemented as a linked list, which XXX would replace the missing statement?

```
StackPop(stack) {
   XXX
   ListRemoveAfter(stack, null)
   return headData
}
```

Correct!

- headData = stack⇢head⇢data
- headData = stack⇢tail⇢data
- headData = null
- headData = stack

Question 22

1 / 1 pts

Which XXX would replace the missing statement in the following algorithm?

```
ListInsertAfter(students, curNode, newNode) {
   if (students→head == null) {
      students→head = newNode
      students→tail = newNode
   }
   XXX {
      students→tail→next = newNode
      newNode→prev = students→tail
      students→tail = newNode
   }
   else {
      sucNode = curNode→next
      newNode→next = sucNode
      newNode→prev = curNode
      curNode→next = newNode
      sucNode→prev = newNode
   }
}
```

- else if (curNode != students→head)

Correct!

- else if (curNode == students→tail)
- else if (sucNode == students→tail)
- else if (curNode == students→head)

Question 23

1 / 1 pts

Identify the error in the following algorithm for traversing a linked list.

```
ListTraverse(list) {
   curNode = list⇢head
   while (curNode is not null) {
      Print curNode's data
      curNode = list⇢head⇢next
   }
}
```

The statement while (curNode is not null) should be while (list⇢tail is not null).

Correct!

The statement curNode = list⇢head⇢next should be curNode = curNode⇢next.

The statement curNode = list⇢head should be curNode = list⇢tail.

The statement while (curNode is not null) should be while (curNode is null).

Question 24

1 / 1 pts

Identify the correct algorithm to use for finding the insertion position of a new item in the linked list studentList.

```
ListFindInsertionPosition(studentList, dataValue) {
   curNodeA = null
   curNodeB = studentList⇢tail
   while (curNodeB != null and dataValue > curNodeB⇢data) {
      curNodeA = curNodeB
      curNodeB = curNodeA
   }
   return curNodeA
}
```

```
ListFindInsertionPosition(studentList, dataValue) {
   curNodeA = null
   curNodeB = studentList⇢head
   while (curNodeB == null and dataValue > curNodeB⇢data) {
      curNodeA = curNodeB
      curNodeB = curNodeB⇢next
   }
   return curNodeA
}
```

```
ListFindInsertionPosition(studentList, dataValue) {
   curNodeB = null
   curNodeA = studentList⇢head
   while (curNodeB == null and dataValue < curNodeB⇢data) {
      curNodeA = curNodeB
      curNodeB = curNodeB⇢next
   }
   return curNodeB
}
```

Correct!

```
ListFindInsertionPosition(studentList, dataValue) {
   curNodeA = null
   curNodeB = studentList⇢head
   while (curNodeB != null and dataValue > curNodeB⇢data) {
      curNodeA = curNodeB
      curNodeB = curNodeB⇢next
   }
   return curNodeA
}
```

Question 25

1 / 1 pts

For an empty, singly-linked list with a dummy node, the list's _____.

- head is null
- Correct! head and tail point to the same, non-null node
- head and tail point to different, non-null nodes
- tail is null

Question 26

1 / 1 pts

Which XXX would replace the missing statement in the given algorithm for traversing a circular linked list?

```
CircularListTraverse(head) {
   if (head is not null) {
      current = head
      do {
         visit current
         current = current⇢next
      } while (XXX)
   }
}
```

- current == tail
- current != tail

Correct!

- ◉ current != head
- ○ current == head

Question 27

1 / 1 pts

In the ListInsertAfter function for singly-linked lists, the curNode parameter is ignored when _____.

- ○ the list's head and tail pointers point to different nodes
- Correct! ◉ the list's head pointer is null
- ○ the list has more than 2 items
- ○ the newNode argument is also null

Question 28

1 / 1 pts

Which XXX completes the following algorithm for inserting a new node into a singly-linked list?

```
ListInsertAfter(list, curNode, newNode) {
   if (list⇢head == null) {
      list⇢head = newNode
      list⇢tail = newNode
   }
   else if (curNode == list⇢tail) {
      list⇢tail⇢next = newNode
      list⇢tail = newNode
   }
   else {
      XXX
   }
}
```

- ```
  newNode⇢next = curNode⇢next
  curNode = newNode
  ```

Correct!

- ```
  newNode⇢next = curNode⇢next
  curNode⇢next = newNode
  ```

- ```
  newNode⇢next = null
  curNode⇢next = list⇢tail⇢next
  ```

- ```
  newNode⇢next = curNode⇢next
  curNode⇢next = null
  ```

Question 29

1 / 1 pts

Identify the error in the following algorithm to search for a node in the singly-linked list of students.

```
ListSearch(students, key) {
   curNode = students⇢head
   while (curNode is null) {
      if (curNode⇢data == key) {
         return curNode
      }
      curNode = curNode⇢next
   }
   return null
}
```

The statement curNode = curNode⇢next should be curNode = students⇢head.

Correct!

The while condition should be while (curNode is not null).

The if condition should be if (curNode⇢data != key).

The statement curNode = students⇢head should be curNode = students⇢tail.

Question 30

1 / 1 pts

When adding an item to an array-based list with an allocation size equal to the list length, a new array is generally allocated with ______ the current length.

Correct!

- twice
- the same size as
- one more than
- one less than

Quiz Score: **29** out of 30