**Question 1**

**0.1 / 0.1 pts**

A doubly linked list keeps track of the next node in the list as well as

itself

the head node

the tail node

the previous node

None of these

**Question 2**

**0.1 / 0.1 pts**

How many steps are involved in the process of deleting a node?

one: delete the node from memory

two: remove the node without breaking links, then delete it from memory

three: create a blank node, remove the node being deleted, insert the blank node

four: create a blank node, insert the blank node before the node being deleted, remove the node being deleted, delete the blank node

None of these

**Question 3**

**0.1 / 0.1 pts**

To append a node to a list means to

delete a node from the beginning of the list

delete a node from the end of the list

add a node to the beginning of the list

add a node to the end of the list

None of these

**Question 4**

**0.1 / 0.1 pts**

If the head pointer points to **nullptr**, this indicates

the list has been previously created and then destroyed

the list needs to be destroyed

there are no nodes in the list

the list is full and cannot accept any new nodes

None of these

**Question 5**

**0.1 / 0.1 pts**

When you delete a node from a list, you must ensure that the links in the list are not permanently broken.

True

False

**Question 6**

**0.1 / 0.1 pts**

To insert a new node in ascending order into a list, the list must be

arranged in descending order

randomly ordered

empty

arranged in ascending order

None of these

**Question 7**

**0.1 / 0.1 pts**

A linked list can consist of structs, objects, and other abstract data types.

True

False

**Question 8**

**0.1 / 0.1 pts**

To create a linked list you must first create a(n)

header file

function template

exception

struct

None of these

**Question 9**

**0.1 / 0.1 pts**

The advantage a linked list has over a **vector** is that

a linked list can dynamically shrink or grow and a **vector** cannot

a linked list is smaller than a **vector**

a node can be inserted or removed faster from a linked list than from a **vector**

data removal and insertion are more accurate with a linked list than with a **vector**

None of these

**Question 10**

**0.1 / 0.1 pts**

The process of moving through a linked list is referred to as \_\_\_\_\_\_\_\_ the list.

cruising

traversing

hopping

revising

None of these

**Question 11**

**0.1 / 0.1 pts**

Linked lists are less complex to code and manage than arrays.

True

False

**Question 12**

**0.07 / 0.1 pts**

Select all that apply. Variations of the linked list are

doubly linked list

circular linked list

frontward linked list

backward linked list

**Question 13**

**0.1 / 0.1 pts**

Select all that apply. Which of the following containers can add elements at their back without requiring traversal?

the **list** container

the **circular\_list** container

the **forward\_list** container

the **reversal\_list** container

**Question 14**

**0.1 / 0.1 pts**

A linked list can grow and shrink as a program runs.

True

False

**Question 15**

**0.1 / 0.1 pts**

The last node in a linked list points to

a null pointer

the previous node

the first node in the list

nothing; the last node does not contain a pointer

None of these

**Question 16**

**0.1 / 0.1 pts**

Nodes in a linked list are stored in contiguous memory.

True

False

**Question 17**

**0.1 / 0.1 pts**

The \_\_\_\_\_\_\_\_ of a linked list points to the first node in the list.

starter

head

tail

declaration

None of these

**Question 18**

**0.1 / 0.1 pts**

A new node must always be made the last node in the list.

True

False

**Question 19**

**0.1 / 0.1 pts**

In an insertion or deletion routine: how many pointers are you required to create for use during the traversal process?

two: one for the node under inspection and one for the previous node

two: one for the node under inspection and one for the next node

one: for the node being inserted or deleted

three: one for the node under inspection, one for the next node, and one for the following node

**Question 20**

**0.1 / 0.1 pts**

A list that contains pointers to the previous node, the next node, and a node in the third dimension is known as a triple linked list.

True

False

**Question 21**

**0.1 / 0.1 pts**

While traversing a list, a node pointer knows when it has reached the end of the list if

it encounters the newline character

it encounters a null pointer

it finds itself back at the beginning of the list

it encounters a sentinel such as 9999

**Question 22**

**0.1 / 0.1 pts**

A linked list is called "linked" because each node in the series has a pointer that points to the next node in the list.

True

False

**Question 23**

**0.1 / 0.1 pts**

When working with a linked list one of the basic operations you can perform is to destroy the list.

True

False

**Question 24**

**0.1 / 0.1 pts**

ADT stands for

Algorithm Dependent Template

Algorithm Driven Template

Abstract Data Type

Automatic Data Type

None of these

**Question 25**

**0.1 / 0.1 pts**

If there are no nodes in a linked list, you cannot append a node to the list.

True

False

**Question 26**

**0.1 / 0.1 pts**

A linked list is a series of connected

ADTs

vectors

algorithms

nodes

None of these

**Question 27**

**0.1 / 0.1 pts**

Deleting an entire list simply requires the use of the **delete** operator.

True

False

**Question 28**

**0.1 / 0.1 pts**

When you create a linked list you must know in advance how many nodes the list will contain.

True

False

**Question 29**

**0.1 / 0.1 pts**

A new node cannot become the first node in the list.

True

False

**Question 30**

**0.1 / 0.1 pts**

Appending a node means adding it to the end of a list, and \_\_\_\_\_\_\_\_ a node means putting a new node in the list, but not necessarily at the end.

concatenating

popping

clamping

inserting

None of these

**Question 31**

**0.1 / 0.1 pts**

If new data needs to be added to a linked list, the program simply \_\_\_\_\_\_\_\_ and inserts it into the series.

allocates another node

removes a node

borrows a node from the compiler

Either removes a node or borrows a node from the compiler

None of these

**Question 32**

**0.1 / 0.1 pts**

The list container provided by the Standard Template Library is a template version of a

singly linked list

doubly linked list

circular linked list

backward linked list

None of these

**Question 33**

**0.1 / 0.1 pts**

The Standard Template Library (STL) provides a linked list container.

True

False

**Question 34**

**0.1 / 0.1 pts**

In a circular linked list, the last node points to the

head pointer

tail pointer

first node

None of these

**Question 35**

**0.03 / 0.1 pts**

Select all that apply. A \_\_\_\_\_\_\_\_ list contains pointers to the nodes before it and after it.

multi linked

circular linked

singly linked

doubly linked

**Question 36**

**0.1 / 0.1 pts**

Select all that apply. Which of the following is a basic list operation?

appending a node

traversing the list

inserting a node

deleting a node

None of these

**Question 37**

**0.1 / 0.1 pts**

A \_\_\_\_\_\_\_\_ is used to travel through a linked list and search for data.

node

pointer

null pointer

traversal operator

None of these

**Question 38**

**0.1 / 0.1 pts**

To delete an entire list, normally you must traverse the list, deleting each node, one by one.

True

False

**Question 39**

**0.1 / 0.1 pts**

Which type of list does NOT contain a null pointer at the end of the list?

backwards linked

doubly linked

circular linked

null linked

None of these

**Question 40**

**0.1 / 0.1 pts**

A linked list class must take care of removing the dynamically allocated nodes and this is done by

the constructor function

the destructor function

overriding the removal function

overloading the memory persistence operator

None of these

**Question 41**

**0.1 / 0.1 pts**

To build a list initially, you can use a(n) \_\_\_\_\_\_\_\_ routine.

build

append

constructor

initialization

None of these