Fisk University

Introduction to Computer Science II, Spring 2009

Quiz #10 (Data Structures) Friday, Apr 3rd, 2009

There are 11 questions to be answered in 20 minutes maximum. Please write legibly.

Q1: Linked lists allow:

a. Insertions and removals anywhere.
b. Insertions and removals only at one end.
c. Insertions at the back and removals from the front.
d. None of the above.

Q2: Which data structure represents a waiting line and limits insertions to be made at the back of the data structure and limits removals to be made from the front?

a. Stack.
b. Queue.
c. Binary tree.
d. Linked list.

Q3: What is the correct way of defining a self-referential "Node" ?

```
a. class Node {  Node n;  };
b. class Node {  Node *n;  };
```

c. All of the above.
d. None of the above.

Q4: The __________ operator takes as an argument the type of object being allocated and returns a __________.

a. `new`, pointer to an object of that type.
b. `delete`, reference to an object of that type.
c. `delete`, copy of the object of that type.
d. `sizeof`, reference to an object of that type.

Q5: Given that the line

```
delete newPtr;
```

just executed, what can you conclude?

a. The memory referenced by `newPtr` is released only if it is needed by the system.
b. The pointer `newPtr` is of type `void *`.
c. The pointer `newPtr` only exists if there was an error freeing the memory.
d. The pointer `newPtr` still exists but the memory pointed to is gone.

Q6: __________ is not an advantage of linked lists when compared to arrays.

a. Dynamic memory allocation.
b. Efficient insertion and deletion.
c. Direct access to any list element.
d. No need to allocate extra space, "just in case."

Q7: For a non-empty linked list, select the code that should appear in a function that adds a node to the end of the list. `newPtr` is a pointer to the new node to be added and `lastPtr` is a pointer to the current last node.   Each node contains a pointer `nextPtr`.

a. `lastPtr->nextPtr = newPtr;`
   `lastPtr = newPtr`.

b. `lastPtr = newPtr;`
   `lastPtr->nextPtr = newPtr`.

c. `newPtr->nextPtr = lastPtr;`
   `lastPtr = newPtr`.

d. `lastPtr = newPtr;`
   `newPtr->nextPtr = lastPtr`.

Q8: What kind of linked list begins with a pointer to the first node, and each node contains a pointer to the next node, and the pointer in the last node points back to the first node?

a. Circular, singly-linked list.
b. Circular, doubly-linked list.
c. Singly-linked list.
d. Doubly-linked list.

Q9: Which of the following statements about stacks is incorrect?

a. Stacks can be implemented using linked lists.
b. Stacks are first-in, first-out (FIFO) data structures.
c. New nodes can only be added to the top of the stack.
d. The last node (at the bottom) of a stack has a null (0) link.

Q10: A stack is initially empty, then the following commands are performed:

```
push 5
push 7
pop
push 10
push 5
pop
```

Which of the following is the correct stack after those commands (assume the top of the stack is on the left)?

a. `5 10 7 5`.
b. `5 10`.
c. `7 5`.
d. `10 5`.

Q11: A queue performs the following commands (in pseudo-code):

enqueue 4, 6, 8, 3, 1

dequeue three elements

enqueue 3, 1, 5, 6

dequeue two elements

What number is now at the front of the queue?

a. *3.*
b. *4.*
c. *5.*
d. *6.*