1. What is one advantage of a linked list structure over array-based structures?

*It is easier to accommodate new memory allocation. Arrays are fixed in size once allocated. The allocation is dynamic in a linked list.*

2. What is one advantage of array-based structures over a linked list structure?

*An important advantage of array-based structures is the availability of random access to the array.*

3. Explain the term “fragmented memory.”

*Memory is generally allocated in blocks. Once parts of a block stop being used the memory locations are available but the block cannot be reallocated because it is partially in use.*

4. What is “dynamic” about dynamic data structures?

*Memory is allocated and deallocated dynamically as need arises.*

5. What is the only condition that would cause the Insert operation of a dynamic data structure to return a “data structure full” error?

*The only way to get a "data structure full" error would be if there is no memory available.*

6. There are many types of linked lists; however, the nodes in all of them have one thing in common. What is it?

*There is a field in the node that points to the next node in the list.*

7. The last node in a singly linked list, by definition, does not store the address of another node. Instead, what does it store?

*The last node has a null pointer. It literally points at nothing.*

8. All singly linked lists contain a reference variable called a “list header”. What is stored in it?

*The list header is the address of the first node.*

9. What is the advantage of implementing a stack using a singly linked list?

*There is no limit on the size of a Stack when it is implemented via a linked list. Note that in Java, the API Stack class is an extension of Vector, not Arrays. Vectors are dynamically resizable. (I think that it isn't possible to extend the Arrays class anyway. I read a discussion on Stack Exchange about this. Apparently Arrays is a final extension of Object. For security reasons, none of the primitives can be extended. So, no class like: "MyClass extends Integer[] ..." is allowed.)*

10. A data set consisting of four information nodes A, X, P, and C, is stored in a singly linked list. A field in each node named next is used to “link” them together. The memory locations of the nodes are: 200, 30, 500, and 60, respectively. The memory cell h is the list header. The nodes are to be stored in alphabetic order in the linked list.

a) Draw a picture, similar to [Figure 4.2a](http://proquest.safaribooksonline.com.libauth.tri-c.edu:2048/9780763757564/42_linked_lists" \l "figure_4.2a), showing the relative position of the nodes in main memory.

b) Add arrows to the picture drawn in part (a) to show the ordering of the nodes starting at the list header (see [Figure 4.3](http://proquest.safaribooksonline.com.libauth.tri-c.edu:2048/9780763757564/43_singly_linked_lists" \l "figure_4.3)).

c) Give the contents of the list header.

d) Give the contents of the next field of each node.

e) Draw the standard depiction (see [Figure 4.4](http://proquest.safaribooksonline.com.libauth.tri-c.edu:2048/9780763757564/43_singly_linked_lists" \l "figure_4.4)) of the four nodes in the linked list.

f) Add the node locations to your answer to part (e), as shown in [Figure 4.5](http://proquest.safaribooksonline.com.libauth.tri-c.edu:2048/9780763757564/43_singly_linked_lists" \l "figure_4.5).

g) Assuming a field named back, was added to each node to order them in reverse alphabetic order, give the contents of this field for each node.

11. Draw the implementation level depiction (see [Figure 4.9](http://proquest.safaribooksonline.com.libauth.tri-c.edu:2048/9780763757564/43_singly_linked_lists" \l "figure_4.9)) of the nodes described in the previous example. (Make up the memory location of the dummy node and the locations of the other linked nodes that are used to implement the structure.)



