1. This would not work. By changing the previous pointer before the new pointer, you would insert it after 3. Also, the size is not increased, so it would not work at all.
2. 1. ADT - What a data type is and how it should function
   2. List - an ordered ADT with one size that can be duplicated
   3. Push - add something to the end of the structure
   4. Pop - remove the last element off the structure and normally return that value
   5. Stack - a data structure that acts as a First in Last out data structure, most common appearance of push and pop
   6. Arrays - ADT with a fixed size that is contiguous in memory
   7. Time Analysis - Amount of time as a function dependent on the amount of data
   8. Linked List - an ADT that has at least one piece of data and a node pointing to the next node in the list
   9. Friend - a class that can access private variables, do not need getters and setters
   10. Kluge - a haphazard, often temporary solution to a problem
   11. K-o-a-l-a
   12. Lists are 0-indexed, so going to position k would in actuality result in you inserting into the k+1th element.
   13. Unlike a doubly linked list, there is no pointer to the previous node, so it would have to be done manually.
   14. All you need to do is get to the address of the first element in the array, and then add, since arrays are contiguous.
   15. You must loop through the entire structure to be sure whether or not the value is present inside.