Challenge 3

By Tjeerd Bakker s2097966 and Soenke van Loh s2270862

questions

a) (single) Linked list can go from beginning to end and access, add, remove any element. There are different subtypes differing in functionality.

   a Stack is a FIFO data struct, only the top element is accessible, and elements are put on top of the stack.

   a queue is a LIFO data struct, only the top element is accessible, and elements are put at the end of the queue.

   A tree is a branched data struct, in order to be useful to the user it should be ordered which allows for easy and efficient searching.

   There are multiple types of trees.

b) Linked List for most dynamic cases, can be used when values must be accessed but there is no certain order.

   stack can be used like a stack of cards where I always want to work with the most recent object.

   Queue can be used for queuing of stuff like threads or users.

   trees (mostly binary trees) are good for storing a lot of sorted data which has to be accessed frequently.

c) Singly Linked List -> Doubly Linked List: add reference to prior object.

   Linked List -> circular linked list: lastptr->nextptr = firstptr

d) nullptr as this way when a function calls that pointer there is an easy check to see if the call is valid.

e) The destructors are called in opposite constructor call order after the last line of code in the scope is executed.

Design decisions List:

References used in order to modify accessed elements and not their copies and/or save memory by not copying the lists.

The string in the fill function is not passed by ref as it should be able for the user to just insert a new string without prior definition there. If the list that is to be filled is not empty false will be returned. Otherwise true is returned after filling.

In the concatenate the toAppend list is taken apart from beginning to end where the first element will be appended to the end of the current list and the last ptr is moved to that object. Afterwards the first object of toAppend is deleted and the first ptr goes to the next obj. Once first ptr is nullptr list is empty and true is returned. False is returned if the list is empty from the beginning on.

Design decisions tree:

Everything is clearly explained in the comments of the code but these are the main takeaways:

- You can use an arrow operator to quickly dereference a pointer and access its data
- Template classes are classes that handle parameters of a type that can be defined at compile time. This allows multiple class instantiation that can handle different types without having to write a new class for each new type.

The implemented search function works as follows:

- If the value of the current node is the desired value, return this node
- If the desired value is larger than the desired value, move to the right node
- If this node does not exist, the number is not in the tree
- If the desired value is smaller than the desired value, move to the left node
- If this node does not exist, the number is not in the tree

The implemented tree print function works as follows:

- If the right subnode is not a nullpointer, move to the right node
- Print the current node's value
- If the left subnode is not a nullpointer, move to the left node