Tutorial 7: Stacks

Stacks

1. What is a stack?
2. What is the end of the stack from which data can be stored and removed called?
3. What are the two operations that can be performed on a stack? What do these operations do to the stack?
4. On what basis are element inserted and deleted in a stack?
5. List three real life examples that work like a stack?
6. Write a C function that will push and element onto a stack. What error does this function throw?
7. Write a C function that will pop an element from a stack. What error does this function throw?
8. What is pushed on the stack to implement a function call?
9. Given the function definitions: int f(in x), int g(int x), and void h(int x, int y). Trace the following function call on the stack h(f(1), g(2)).

Using Stacks

10. Do we need a stack if we are only matching ')' against '('?
11. Track a stack matching parentheses for {(a + b) * (c + d)].

Implementing stacks with linked lists

12. Write a C function to push an item onto a linked list.
13. Write a C function to pop an item from a linked list.
14. Trace the functions written in the previous questions to match parentheses for {(a + b) * (c + d)].