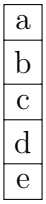
Tutorial 4

1 Stack

- A stack is an ordered container of data that enforces a LIFO policy on adds/removes
- Permitted operations: push, pop, peek, isEmpty
- e.g. back and forward putton on a browser. Any idea how to implement this?



2 References

- References are like pointers
- The reference parameter is just another name (i.e., an alias) for the variable in the calling context

Q: Is the valid?

```
int foo(int &i) {return i;}
int main() {foo(1);}
```

3 Exercises

1. Write a recursive function **bool isCircular(const List &l)** that returns true if a singly linked list is circular and false otherwise. You may use helper functions if you wish.
2. Write a recursive function **int findMin(const List &l)** that returns the minimum value stored in a singly linked list of integers. You may use helper functions if you wish.
3. Write a recursive function int recursivePower(int base, int exponent) that returns base-exponent. For example, **recursivePower(4, 2)** should return 16. You should not need to use helper functions. Try to solve it in $O(\log n)$ time. (Hint: $x^{10} = x^5 \times x^5$)