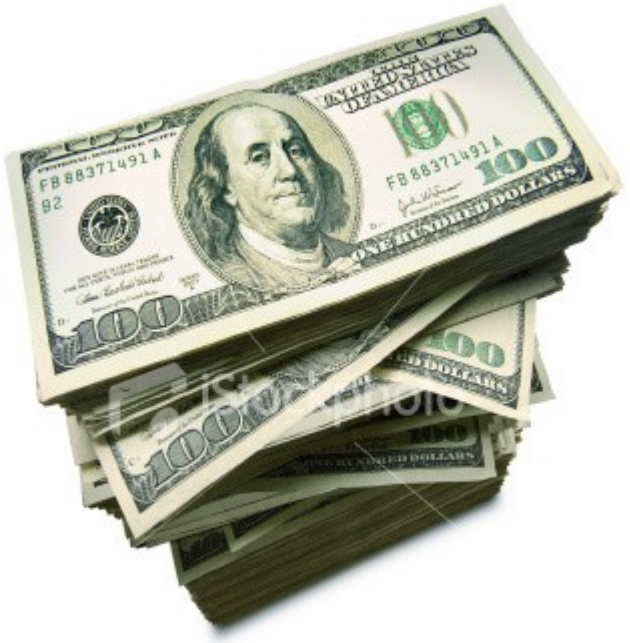
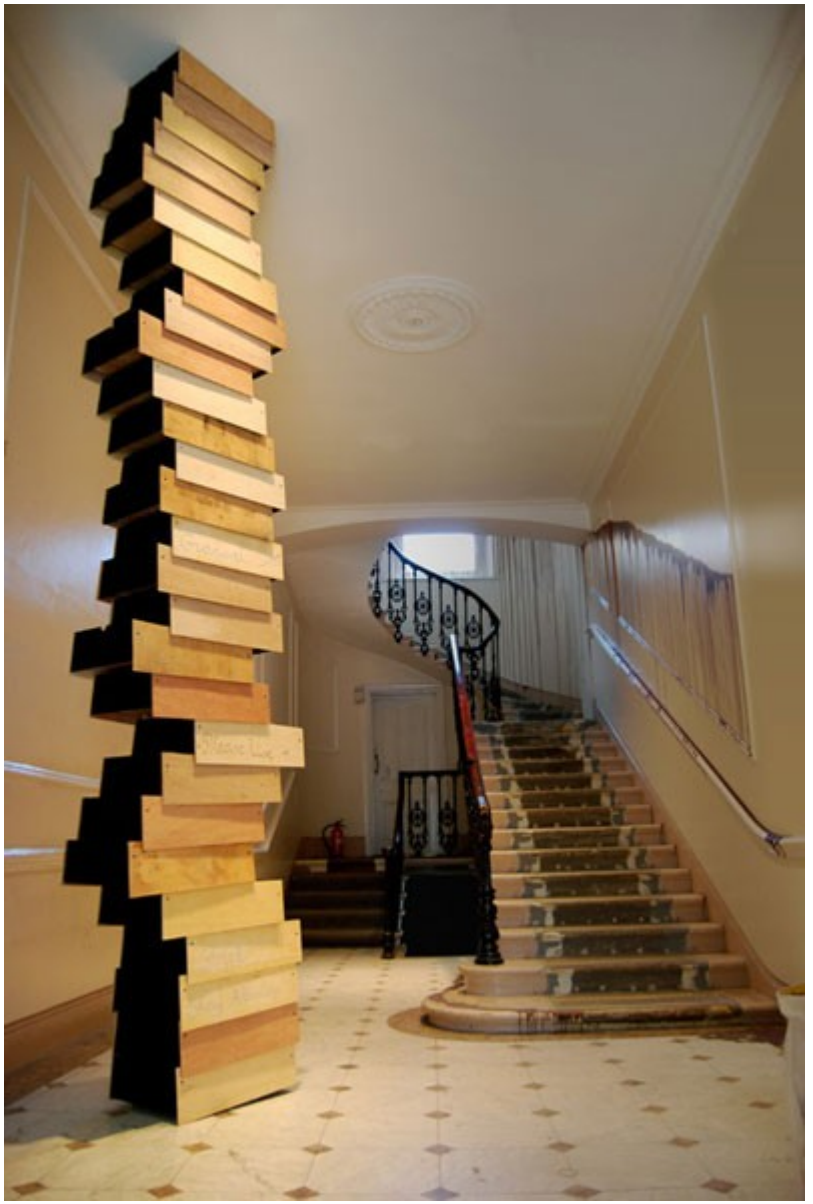
Stosy i kolejki

Stos





Cechy stosu

- Można położyć element na wierzch stosu
- Elementów na stosie może być ile chcemy
- Można zdjąć element z wierzchołka stosu, chyba że stos jest pusty.
- Stos pusty to oczywiście też stos
- Mając funkcję operującą na stosie, przez stos rozumiemy jego wierzchołek (wskazanie na pierwszy element)
- LIFO

Kolejka

Cechy kolejki

- Można dołożyć element na koniec kolejki
- Można pobrać element z początku kolejki
- Kolejka pusta to oczywiście też kolejka
- Przekazując kolejkę do funkcji mamy na myśli jej ostatni element
- FIFO

Zadania

- Zaimplementować:
  - push(S, x), pop(S), empty(S)
  - enqueue(Q, x), dequeue(Q)

Zadania cd.

- Sprawdzić poprawność zagnieżdżenia nawiasów
  - bool sprawdzNawiasy(string s, int dlugosc);
  - 'dlugosc' oznacza długość napisu
  - s[i] zwraca znak na i-tej pozycji
  - rozwiązać zadanie wykorzystując stos

Zadania cd.

- Odwrócić stos
  - void odwroc(elementStosu stos)
  - Wykorzystać dwa stosy pomocnicze

Odwrotna notacja polska

- (3+5)*4 == 3 5 + 4 *
- Wskazówki:
  - Dane (3 5 + 4 *) są przechowywane na stosie
  - Jeśli stos danych nie jest pusty, pobieramy kolejną daną:
    - Jeśli to liczba, to odkładamy ją na stos pomocniczy
    - Jeśli to operator, to zdejmujemy dwie liczby ze stosu pomocniczego, wykonujemy działanie i odkładamy wynik na stos pomocniczy
  - Jeśli stos danych jest pusty, zwracamy jako wynik liczbę ze stosu pomocniczym
- Operacje: czyOperator(dana), push(S, dana), pop(S, dana), empty(S)