

## Bài tập chương 2

### Phần stack, queue

**Bài 1.** Thực hiện liên tiếp các lệnh sau đối với một stack rỗng, kết quả thu được sẽ là gì? Vẽ hình minh họa.

```
Push(3);
Push(5);
Pop();
Push(7);
Pop();
Push(8);
Pop();
Pop();
```

**Bài 2.** Cho một danh sách móc nối với cấu trúc mỗi nút như sau:

```
typedef struct node
{
    float data;
    struct node *pNext;
} NODE;
```

Hãy viết hàm reverse để in ra nội dung danh sách móc nối theo thứ tự ngược

```
void reverse (NODE* pHead)
```

Gợi ý: sử dụng stack

**Bài 3.** Tính giá trị các biểu thức hậu tố sau, mô tả rõ các bước làm

- a)  $3\ 4\ +\ 5\ 6\ 7\ *\ 2\ -\ *\ -$
- b)  $3\ 4\ +\ 2\ ^\ 6\ 4\ *\ 2\ /\ -$
- c)  $3\ 4\ 7\ +\ \% \ 2\ 3\ /\ -$
- d)  $3\ 3\ 5\ \% \ ^\ 9\ 5\ 2\ -\ /\ 6\ *\ -$

**Bài 4.** Chuyển các biểu thức trung tố sau sang biểu thức hậu tố

- a)  $(a + b) * c^{(d + 4 - 5 * (9/3))}$
- b)  $3 + 4 / (2 + a - c * b) - 7 \% (5 + c)$
- c)  $2 + (3 - (4 * (5^{a \% b})))$
- d)  $3 - a * (3^2 - 6 \% (5 + 9))$

**Bài 5.** Định giá các biểu thức hậu tố sau (các toán hạng và toán tử cách nhau 1 dấu cách, các toán hạng có thể gồm nhiều chữ số)  $\sqrt{\quad}$  là toán tử 1 ngôi căn bậc hai, *AND*, *NOT*, *OR* là các toán tử logic,  $\geq$ ,  $>$ ,  $\leq$  là các toán tử quan hệ,  $!$  là toán tử 1 ngôi tính giai thừa, *abs* là toán tử 1 ngôi tính giá trị tuyệt đối

- a)  $15\ 2\ 3\ +\ /\ 2\ ^\ 4\ 12\ 2\ -\ \% \ +$
- b)  $1\ 3\ 5\ +\ +\ \sqrt{\quad}\ 2\ ^\ 4\ 12\ 6\ /\ -\ +$
- c)  $2\ 1\ 27\ 3\ 2\ ^\ /\ +\ 3\ \% \ -\ 4\ +$
- d)  $3\ 0\ \geq\ 45\ 7\ >\ 5\ 6\ \leq\ AND\ NOT\ OR$
- e)  $2\ 3\ 1\ 12\ 3\ 4\ *\ /\ +\ ^\ \%$

f)  $3 \ 4! \ 4 \ 5 + \% \ 12 + 15 - +$

g)  $1 \ 16 - abs \ 5 + 4 \ 6 + /$

**Bài 6.** Chuyển các biểu thức sau từ dạng trung tố về dạng hậu tố (các toán hạng và toán tử cách nhau 1 dấu cách, các toán hạng có thể gồm nhiều chữ số)

a)  $3 + 5 ^ (12 / 6 + 1) - 7 * 15 / 3 + 6$

b)  $\sin (5 * a + 7) + 2 - 6 * 5 * b + 3 ^ 2 ^ c / 2$

c)  $23 + a - \sqrt{(6 * b + 5)} / 9 ^ 2 * \cos(3 * c)$  (chú ý  $\sqrt{(6 * b + 5)}$  là tương đương với  $\sqrt{6 * b + 5}$ )

d)  $(3 \geq a) \text{ AND } ((c \geq d) \text{ OR } (NOT (4 < d)))$

e)  $5 ^ 2 ^ 3 ^ (4 * 9 - 5 / b + 6) \% (7 + a + c)$

f)  $abs (6 - b * 2 + 7 * a) ^ 2 / c + 5$

**Bài 7.** Cài đặt thuật toán định giá biểu thức trung tố tổng quát (thực hiện trên tập các toán tử trong slide)

**Bài 8.** Cài đặt thuật toán chuyển biểu thức ở dạng trung tố sang dạng hậu tố

**Bài 9.** Bài toán kiểm tra cặp ngoặc trong biểu thức có hợp lệ

Một biểu thức có dấu ngoặc là hợp lệ nếu số dấu ngoặc mở phải bằng số dấu ngoặc đóng cùng loại.

Ví dụ:

Biểu thức hợp lệ

- $\{2 + 3 * (5/3^2)\} / \{7 * b\}$
- $2 + \{3 * [4 + 6 - (2/3^2)]\}$

Biểu thức không hợp lệ

- $\{3 + (6 - [5 * 2])\}$
- $4 + (6 * (5 - a)$

Hãy viết chương trình nhận đầu vào là 1 biểu thức, kiểm tra xem biểu thức đó có hợp lệ hay không.

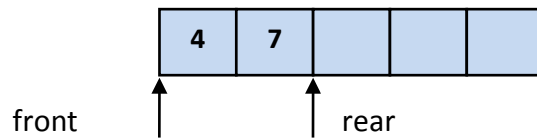
**Bài 10.** Bài toán đối sánh thẻ HTML

Trong một văn bản HTML thì mỗi thẻ mở phải đi kèm với 1 thẻ đóng

```
<div class="header-right">
  <form method="get" id="searchform" action="http://tinhocdaicuong.wordpress.com/" >
    <div><label class="hidden" for="s">Tìm kiếm:</label>
    <input type="text" value="" name="s" id="s" />
    <input type="submit" id="searchsubmit" value="Go" /></div>
  </form>
</div>
```

Hãy viết chương trình đọc vào 1 file .html và kiểm tra xem trong file đó có thẻ nào bị lỗi hay không

**Bài 11.** Cho một Queue được lưu trữ bởi một mảng có 5 ô nhớ



Ban đầu **front** = -1 và **rear** = 1, thực hiện liên tiếp các lệnh sau

1. **enqueue(5)**
2. **dequeue()**
3. **enqueue(3)**
4. **dequeue()**
5. **dequeue()**
6. **enqueue(9)**

Hãy vẽ trạng thái của queue sau mỗi bước

Nếu Queue được tổ chức dạng vòng thì kết quả có gì khác sau khi ta cũng thực hiện liên tiếp 6 lệnh trên, đồng thời thực hiện tiếp các lệnh

7. **enqueue(1)**
8. **dequeue()**
9. **enqueue(10)**

**Bài 12.** Cài đặt minh họa queue vòng dùng mảng

**Bài 13.** So sánh ưu nhược điểm khi lưu trữ queue bằng cấu trúc liên tiếp (mảng) và cấu trúc liên kết (danh sách liên kết đơn)

**Bài 14.** Cài đặt ứng dụng của queue để mô phỏng hàng đợi khám bệnh tại bệnh viện như mô tả trong slide