

CẤU TRÚC DỮ LIỆU & GIẢI THUẬT
HOMEWORK 5.2 – Ngăn xếp, Hàng đợi
---oOo---

1. Giả sử đã có class Stack được khai báo ở trang 27, chương 5

- a. Hãy viết các lệnh để thực hiện những yêu cầu sau:
- Khai báo stack S có N phần tử kiểu int. Khởi tạo S rỗng.
 - Đưa các giá trị sau vào S: 15, 8, 6, 21
 - Lấy 21 ra khỏi S
 - Lấy 8 ra khỏi S
 - Gán các giá trị 1 → 99 vào S
 - Lần lượt lấy các phần tử trong S ra và in lên màn hình
- b. Cho mảng a chứa N số nguyên (giá trị tuần tự từ 1 đến N). Hãy viết các lệnh để đảo ngược các phần tử của mảng a.

2. Cho một ngăn xếp s và 1 đoạn chương trình sau:

```
struct STACK s;  
InitStack(s, 10);    // khởi tạo stack s rỗng, maxsize=10  
int x, y = 5;  
Push(s, 8);          // push 8 vào s  
Push(s, y);  
Push(s, 9);  
Pop(s, x);            // lấy phần tử ở đỉnh stack và lưu vào x  
Push(s, 18);  
Pop(s, x);  
Push(s, 22);  
while (IsEmpty(s) == 0)  
{  
    Pop(s, x);  
    printf("%d ", x);  
}
```

Hãy cho biết kết quả khi thi hành đoạn code trên ?

3. Ta có 4 lựa chọn:

1. Stack; 2. Queue; 3. List; 4. Cả 3 CTDL này đều không thích hợp.
Trong mỗi tình huống sau, hãy cho biết áp dụng lựa chọn nào là thích hợp nhất:
- a. Các khách hàng tại quầy bán vé xe lửa lấy số thứ tự để mua vé
 - b. Một danh sách tên theo thứ tự ABC
 - c. Các số nguyên cần phải sắp thứ tự
 - d. Danh sách các món hàng đã bán trong ngày tại quầy thu ngân trong siêu thị
 - e. Chương trình có sử dụng kỹ thuật Back-tracking
 - f. Các máy bay đang chờ đáp xuống phi trường

4. Thuật toán Balan ngược

Cho các biểu thức số học ở dạng hậu tố (postfix). Hãy áp dụng thuật toán Ba Lan ngược để tính toán giá trị của các biểu thức đó. Ghi rõ trạng thái của stack qua mỗi bước tính toán

- a. $10\ 3 + 2 * 5 * 7\ 1 - 3 / +$
b. $3\ 5 + 2 * 7\ 3 + 5 / + 2\ 8 * -$

5. Hãy áp dụng thuật toán Balan ngược để biến đổi biểu thức sau thành dạng postfix:

$$(2+6/(8-5))*9$$

Trình bày trạng thái của stack sau mỗi lần có sự thay đổi. Tính giá trị của biểu thức postfix, ghi rõ trạng thái của stack qua mỗi bước tính toán.

6. Hãy áp dụng thuật toán Balan ngược để viết chương trình chuyển đổi biểu thức dạng trung tố (infix) sang dạng hậu tố (postfix) và tính giá trị biểu thức.

Chương trình cần báo lỗi khi biểu thức có lỗi dấu ngoặc đơn: thiếu hay thừa dấu “(“ hay dấu “)””. Chỉ báo lỗi xuất hiện đầu tiên.

Yêu cầu:

- Input: biểu thức trung tố được đọc từ file input.txt
- Output: kết quả được xuất ra file output.txt
- File input.txt và output.txt mặc định lưu cùng thư mục với chương trình.
- Không được phép sử dụng class stack có sẵn của C++
- Biểu thức infix chỉ gồm các số nguyên dương có 1 chữ số và các dấu $+ - * / ()$

Ví dụ:

input.txt	output.txt
$5+((1+2)*4)+3$	$512+4*+3+$
$5+((1+2)*4+3$	Lỗi thiếu dấu)
$5+(1+2)*4)+3$	Lỗi thừa dấu)

7. Cho hàng đợi ban đầu như sau: (hàng đợi có tối đa 6 phần tử)

0	1	2	3	4	5
	A	B	C		
f = 1		r = 3			

Vẽ tình trạng của hàng đợi, cho biết giá trị f, r tương ứng với mỗi lần thực hiện thao tác sau:

Thao tác	f	r	Queue					
			0	1	2	3	4	5
Bổ sung E vào hàng đợi								
Loại 2 phần tử khỏi hàng đợi								
Bổ sung I, J, K vào hàng đợi								
Loại 2 phần tử khỏi hàng đợi								
Bổ sung O vào hàng đợi								
Loại 2 phần tử khỏi hàng đợi								

8. Stack và Queue

Giả sử rằng cấu trúc dữ liệu được dùng để lưu trữ các phần tử của ngăn xếp và hàng đợi là một danh sách liên kết đơn (có thông tin về phần tử đầu pHead và phần tử cuối pTail). Hãy cho biết khi cài đặt các thao tác THÊM VÀO và LẤY RA của ngăn xếp và hàng đợi thì sử dụng các

thao tác nào của danh sách liên kết đơn (AddHead, AddTail, RemoveHead, RemoveTail) sẽ cho hiệu quả nhất? Điền câu trả lời vào chỗ trống trong bảng sau:

	Ngăn Xếp		Hàng Đợi	
	Thuật ngữ	Thao tác cài đặt	Thuật ngữ	Thao tác cài đặt
Thêm vào				
Lấy Ra				

--- Hết ---