

# BÀI TẬP THỰC HÀNH 03

## STACK - QUEUE

Hạn chót: xem tại nơi nộp bài trên <https://courses.uit.edu.vn/>

### NỘI DUNG

#### Bài 1

Viết chương trình cài đặt **stack** bằng danh sách liên kết đơn (*hoặc bằng mảng*) để lưu trữ tạm các số **nguyên**. Thực hiện các yêu cầu sau:

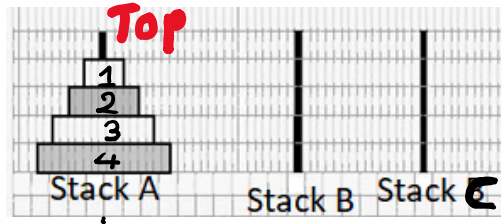
- Câu 1.** Viết hàm khởi tạo stack.
- Câu 2.** Viết hàm kiểm tra stack rỗng. Nếu rỗng trả về **true**, không rỗng trả về **false**.
- Câu 3.** Viết hàm kiểm tra stack có full hay không (nếu có)? Nếu full trả về **true**, không full trả về **false**.
- Câu 4.** Viết hàm đẩy một phần tử vào stack.
- Câu 5.** Viết hàm lấy một phần tử ra khỏi stack.
- Câu 6.** Xuất ra phần tử TOP của stack.
- Câu 7.** Trong hàm main, viết menu thể hiện các lựa chọn câu trên.

#### Bài 2

Viết chương trình cài đặt **queue** bằng danh sách liên kết đơn (*hoặc bằng mảng*) để lưu trữ tạm các số **nguyên**. Thực hiện các yêu cầu sau:

- Câu 1.** Viết hàm khởi tạo queue.
- Câu 2.** Viết hàm kiểm tra queue rỗng. Nếu rỗng trả về **true**, không rỗng trả về **false**.
- Câu 3.** Viết hàm kiểm tra queue có full hay không (nếu có)? Nếu full trả về **true**, không full trả về **false**.
- Câu 4.** Viết hàm đưa một phần tử vào queue.
- Câu 5.** Viết hàm lấy một phần tử ra khỏi queue.
- Câu 6.** Xuất ra phần tử REAR và FRONT của queue.
- Câu 7.** Trong hàm main, viết menu thể hiện các lựa chọn câu trên.

### Bài 3. Viết chương trình cài đặt **stack** minh họa bài toán Tháp Hà Nội:



Chú ý: SV theo dõi hướng dẫn cách cài đặt tháp của GV trong quá trình hướng dẫn thực hành.

Xem minh họa:

- [https://upload.wikimedia.org/wikipedia/commons/4/4f/Tower\\_of\\_Hanoi.gif](https://upload.wikimedia.org/wikipedia/commons/4/4f/Tower_of_Hanoi.gif)
- [https://upload.wikimedia.org/wikipedia/commons/6/60/Tower\\_of\\_Hanoi\\_4.gif](https://upload.wikimedia.org/wikipedia/commons/6/60/Tower_of_Hanoi_4.gif)

Mô tả & qui ước:

- Số lượng đĩa/tầng tháp tự qui định, nhưng tối thiểu 3 đĩa/tầng tháp.
- Ký hiệu đĩa/tầng tháp là “4”, “3”, “2”, “1”... (theo qui tắc Bottom => Top).
- Mỗi lần thực hiện, chỉ được chuyển 1 đĩa từ một cột này sang một trong hai cột khác.
- Đĩa to **KHÔNG** ở trên đĩa nhỏ.
- Cách thể hiện: GV hướng dẫn trên lớp.

Yêu cầu chung:

- Thiết kế theo hàm.
- Hàm main thể hiện menu lựa chọn các bước di chuyển tháp.

**Hình DEMO:**

```
"E:\GoogleDrive ptheson\IT003..."
===== Demo Stack =====
0. Creating Tower.
1. A -> B
2. B -> A
3. B -> C
4. C -> B
5. C -> A
6. A -> C
=====
A Tower
STACK = < 3 2 1 >
=====
B Tower
STACK = < >
=====
C Tower
STACK = < >
=====
Please input:
```

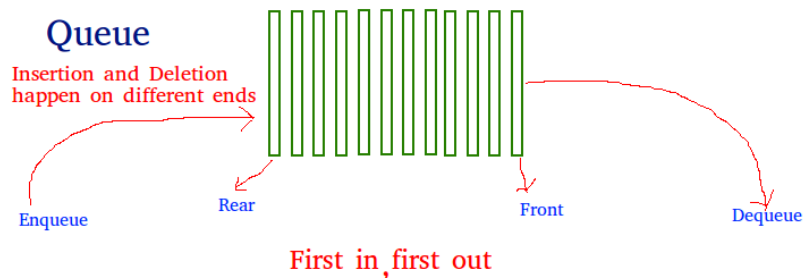
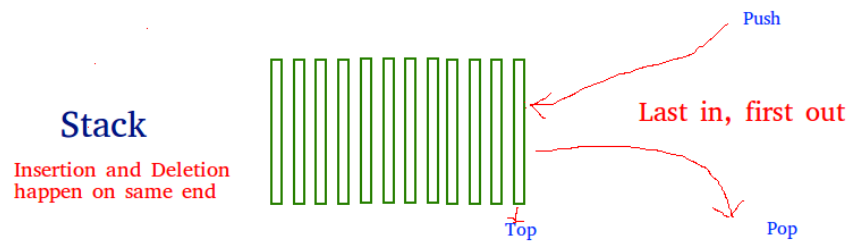
Trạng thái ban đầu

```
"E:\GoogleDrive ptheson\IT003..."
===== Demo Stack =====
0. Creating Tower.
1. A -> B
2. B -> A
3. B -> C
4. C -> B
5. C -> A
6. A -> C
=====
A Tower
STACK = < >
=====
B Tower
STACK = < >
=====
C Tower
STACK = < 3 2 1 >
=====
Please input:
```

Đã di chuyển xong tháp

--- HẾT ---

## HÌNH MINH HỌA CƠ CHẾ HOẠT ĐỘNG CỦA STACK - QUEUE



## QUY ĐỊNH CHUNG

1. Sử dụng C/C++ để thực hành. Trường hợp sinh viên sử dụng ngôn ngữ khác (như Python...) thì sẽ cộng thêm điểm, và phải báo trước giảng viên thực hành về sử dụng ngôn ngữ khác để thực hành.

2. Phải đặt tên file theo cách sau: **MSSV\_HoVaTen\_BuoiXX\_BaiYY.cpp**

Trong đó:

- **MSSV**: là mã số sinh viên.
- **HoVaTen**: Họ và tên của SV, viết liền không dấu. VD: NguyenVanAn
- **BuoiXX**: Thứ tự buổi thực hành. VD: Buoi01, Buoi02, Buoi03...
- **BaiYY**: Thứ tự bài. VD: Bai01, Bai02, Bai03...
- Mỗi thành phần cách nhau dấu “\_”
- Một ví dụ hoàn chỉnh: **39\_18520779\_NguyenVanAn\_Buoi01\_Bai01.cpp**
- Nếu có nhiều bài (nhiều file lập trình) thì SV nén lại với cấu trúc sau:

3. Trong file dùng để lập trình, SV ghi thêm đầy đủ thông tin như sau:

```
//STT: 39
//Họ và tên: Nguyễn Văn A
//Buổi 01 - Bài 01
//Ghi chú, hoặc Lưu ý: .....
```

4. Đối với các bài thực hành có nhiều ý nhỏ (nếu có), SV phải viết các menu lựa chọn cho từng tính năng. GV thực hành sẽ hướng dẫn chi tiết trong buổi học.
5. SV không nộp bài qua mail.