

## BÀI TẬP THỰC HÀNH 05

### CÂY NHỊ PHÂN

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

#### NỘI DUNG

Viết chương trình cài đặt cây tìm kiếm nhị phân lưu trữ các **số nguyên** (hoặc **số thực**). Sau đó, viết hàm thực hiện các yêu cầu sau:

- Câu 1.** Viết hàm chèn node chứa giá trị (dữ liệu) vào cây.
- Câu 2.** Viết thêm hàm nhập giá trị cho cây theo cách thủ công nhập từ bàn phím (*điều kiện kết thúc tự quy định, không dùng phương pháp nhập số lượng phần tử cần nhập*).
- Câu 3.** Viết hàm tạo cây tự động. Các giá trị được chọn ngẫu nhiên trong đoạn [-358; 1358]. Số lượng cho một lần tạo tự động [10; 20].
- Câu 4.** Viết hàm duyệt cây theo NLR, LNR, LRN. Chú ý: In kèm thêm 3 địa chỉ của NODE giá trị, địa chỉ Left và địa chỉ Right cho cách duyệt LNR.

Ví dụ:

LNR :			
5	0xcba4b0	pLeft: 0	pRight: 0
10	0xcb6c88	pLeft: 0xcba4b0	pRight: 0xcba4c8
12	0xcba4e0	pLeft: 0	pRight: 0
13	0xcba4c8	pLeft: 0xcba4e0	pRight: 0xcba4f8
14	0xcba4f8	pLeft: 0	pRight: 0
15	0xcb6c70	pLeft: 0xcb6c88	pRight: 0xcba498
17	0xcba510	pLeft: 0	pRight: 0
20	0xcba498	pLeft: 0xcba510	pRight: 0xcba528
25	0xcba528	pLeft: 0	pRight: 0

- Câu 5.** Viết hàm tìm kiếm từ giá trị X. X do người dùng nhập vào hoặc được chọn ngẫu nhiên trong [300; 800]. Nếu tìm thấy giá trị X thì trả về địa chỉ của node hoặc True. Nếu không tìm thấy trả về False.
- Câu 6.** Viết hàm xóa một giá trị X trên cây.
- Câu 7.** Viết hàm xóa toàn bộ các node trên cây.
- Câu 8.** Viết hàm đếm toàn bộ số node của cây.
- Câu 9.** Viết hàm đếm số node lá của cây.
- Câu 10.** Viết hàm in ra các node nhánh (NLR) từ một node nhập từ bàn phím.
- Câu 11.** Viết hàm đếm số node có giá trị nhỏ hơn X.
- Câu 12.** Viết hàm đếm số node có giá trị lớn hơn X.
- Câu 13.** Viết hàm đếm số node có giá trị lớn hơn X và nhỏ hơn Y ( $X < \text{node} < Y$ ).

**Câu 14.** Viết hàm so sánh hai giá trị đếm node của câu #11 và #12.

**Câu 15.** Viết hàm đếm các node có giá trị chẵn và lẻ. So sánh hai giá trị đếm được.

**Câu 16.** Viết hàm tính tổng các node trong cây.

**Câu 17.** Viết hàm tính tổng các node chẵn trong cây.

**Câu 18.** Viết hàm tính tổng node lẻ có trong cây.

**Câu 19.** Viết hàm so sánh kết quả câu #17 và câu #18.

**Câu 20.** Viết hàm tìm ra node có giá trị lớn nhất và nhỏ nhất trong cây.

**Câu 21.** Trong hàm main thể hiện các menu lựa chọn các hàm thuộc các câu trên.

--- HẾT ---

Thống nhất cách khai báo cấu trúc NODE

```
struct NODE
{
    int data;
    NODE* pLeft;
    NODE* pRight;
};
```

## 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.