

NỘI DUNG CÁC BUỔI THỰC HÀNH

Buổi 1

Bài 1: Nhập mảng 1 chiều có N phần tử, mỗi phần tử có kiểu dữ liệu là 1 số nguyên. Hãy thực hiện các yêu cầu sau bằng ngôn ngữ lập trình C.

- a. Xuất mảng 1 chiều nói trên ra màn hình.
- b. Xoá 1 phần tử tại chỉ số K trong mảng.
- c. Liệt kê những phần tử mang giá trị lớn nhất trong mảng
- d. Liệt kê những phần tử mang giá trị nhỏ nhất trong mảng
- e. Đếm số phần tử có giá trị âm trong mảng
- f. Cho biết giá trị âm lớn nhất trong mảng.
- g. Cho biết giá trị dương nhỏ nhất trong mảng
- h. Tìm một phần tử có khoá bằng X có hay không trong mảng (dùng thuật toán tìm kiếm tuyến tính)
- i. Đếm số phần tử có giá trị bằng X trong mảng. Nếu không có thì thông báo không có.
- j. Sắp xếp mảng trên có giá trị tăng dần bằng thuật toán sắp xếp mà các bạn đã học (Selection Sort, Interchange Sort, Bubble Sort, Insertion Sort, Quick Sort)
- k. Tìm phần tử có giá trị bằng X trong mảng bằng thuật toán tìm kiếm nhị phân từ mảng đã được sắp xếp ở câu j.

Bài 2: Dùng mảng một chiều để lưu trữ một lớp học có N sinh viên. Biết rằng mỗi sinh viên bao gồm các thông tin sau: Tên (chuỗi ký tự), Mã số sinh viên (chuỗi ký tự), Điểm trung bình.

Hãy viết hàm thực hiện các yêu cầu sau:

- a. In danh sách sinh viên ra màn hình
- b. Liệt kê những sinh viên có điểm trung bình cao nhất trong lớp học.
- c. Cho biết số sinh viên có điểm trung bình ≥ 5 . Nếu không có thì thông báo không có.
- d. Tìm một sinh viên có tên X trong lớp học (X nhập từ bàn phím)
- e. Xoá một sinh viên có mã số cho trước trong lớp học. Nếu không có thì thông báo không có.
- f. Sắp xếp danh sách sinh viên tăng theo điểm trung bình bằng thuật toán sắp xếp mà các bạn đã học (Selection Sort, Interchange Sort, Bubble Sort, Insertion Sort, Quick Sort)
- g. Chèn một sinh viên vào lớp học, biết rằng sau khi chèn danh sách sinh viên vẫn tăng dần theo điểm trung bình.

Chú ý: Các thao tác ở câu 1 và câu 2 cho phép người dùng chọn từ Menu.

Bài 3: Dùng mảng 1 chiều để lưu trữ khi đổi một số ở hệ 10 sang hệ nhị phân

Bài 4: Dùng mảng 1 chiều để lưu trữ khi đổi một số ở hệ 10 sang hệ 16

Buổi 2

Bài 1: Nhập mảng 1 chiều có N phần tử, hãy sắp xếp mảng tăng dần bằng các thuật toán **Binary Sort, Shaker Sort, Shell Sort, Heap Sort**.

Bài 2: Phát sinh ngẫu nhiên mảng 1 chiều có N phần tử, mỗi phần tử có kiểu dữ liệu là 1 số nguyên. Sắp xếp và đo thời gian sắp xếp của các thuật toán ở bài 1.

Bài 3: Nhập mảng 1 chiều có N phần tử, mỗi phần tử là 1 số nguyên. Hãy sắp xếp và thống kê số phép so sánh, số phép gán của những thuật toán ở bài 1.

Bài 4: Thực hiện bài quản lý sinh viên, như ở buổi 1. Tuy nhiên thông tin của các sinh viên được lưu trên file. Với dòng đầu tiên là số sinh viên, các dòng tiếp theo là thông tin của từng sinh viên

N

0001 Tuan 7.3

Buổi 3

Bài 1: Nhập mảng 1 chiều có N phần tử, hãy sắp xếp mảng tăng dần bằng các thuật toán **Quick Sort, MergeSort, Radix Sort**.

Bài 2: Phát sinh ngẫu nhiên mảng 1 chiều có N phần tử, mỗi phần tử có kiểu dữ liệu là 1 số nguyên. Sắp xếp và đo thời gian sắp xếp của các thuật toán ở bài 1.

Bài 3: Nhập mảng 1 chiều có N phần tử, mỗi phần tử là 1 số nguyên. Hãy sắp xếp và thống kê số phép so sánh, số phép gán của những thuật toán ở bài 1.

Buổi 4

Bài 1: Tạo một danh sách liên kết đơn, biết rằng thành phần dữ liệu của một nút là một số nguyên dương, việc nhập kết thúc khi ta nhập 1 số âm. Thực hiện các thao tác sau.

- Thêm một nút vào đầu danh sách, vào cuối danh sách liên kết đơn.
- Đếm số nút trong danh sách liên kết đơn
- Tìm một nút có thành phần dữ liệu bằng X
- Xoá 1 nút đầu, cuối danh sách liên kết đơn.
- Giả sử trường dữ liệu của các thành phần trong danh sách liên kết đơn khác nhau từng đôi một. Hãy viết hàm xoá một nút có trường dữ liệu bằng X, nếu không có thì thông báo không có.
- Hãy sắp xếp danh sách liên kết đơn tăng dần theo thành phần dữ liệu.
- Hãy chèn 1 nút có trường dữ liệu bằng X (X nhập từ bàn phím), sao cho sau khi chèn danh sách vẫn tăng dần theo trường dữ liệu.

Bài 2: Dùng danh sách liên kết đơn để lưu trữ và in ra khi đổi một số ở hệ 10 sang hệ 2.

Bài 3: Dùng danh sách liên kết đơn để lưu trữ và in ra khi đổi một số ở hệ 10 sang hệ 16.

Bài 4: Dùng danh sách liên kết đơn để tính tổng, hiệu hai ma trận thưa.

Hướng dẫn: Lưu giá trị các phần tử khác 0 và vị trí của nó, vị trí được tính bằng cách đưa về chỉ số

của các phần tử trong mảng một chiều. Ví dụ: 4 ở vị trí 0, 1 ở vị trí 3, 2 ở vị trí 7, 3 ở vị trí 9 đối với ma trận dưới đây:

| | | | |
|---|---|---|---|
| 4 | 0 | 0 | 1 |
| 0 | 0 | 0 | 2 |
| 0 | 3 | 0 | 0 |

Bài 5: Dùng danh sách liên kết đơn để quản lý một lớp học tương tự như **bài 2** ở **buổi 1**.

Bài 6: Dùng danh sách liên kết đơn để cộng, trừ, nhân chia hai số lớn.

Bài 7: Cài đặt các thao tác trên danh sách liên kết kép.

Buổi 5

Bài 1: Cài đặt Stack (ngăn xếp) bằng mảng một chiều

Bài 2: Cài đặt Stack bằng danh sách liên kết đơn.

Bài 3: Cài đặt Queue bằng mảng một chiều

Bài 4: Cài đặt Queue bằng danh sách liên kết đơn

Bài 5: Dùng Stack để tính biểu thức toán học được nhập dạng hậu tố như sau: Biểu thức hậu tố là biểu thức có dạng như sau: 532*+

Biểu thức hậu tố trên chính là $5+(3*2) = 11$ trong dạng biểu diễn trung tố.

Bài 6: Dùng Queue mô phỏng dòng chữ chạy quanh màn hình.

Buổi 6

Bài 1: Tạo một cây nhị phân tìm kiếm, trong đó trường Key tại mỗi nút là một số nguyên dương. Hãy viết hàm thực hiện các thao tác sau:

a. In cây nhị phân tìm kiếm nói trên theo các thứ tự : LNR, LRN, NLR, NRL, RNL, RLN.

b. Tìm một nút có khoá bằng X trên cây.

c. Xóa 1 nút có khoá bằng X trên cây, nếu không có thì thông báo không có.

d. Đếm số nút trong cây

e. Đếm số nút có đầy đủ 2 cây con

f. Đếm số nút lá.

g. Tính chiều cao của cây

h. Vẽ cây nhị phân tìm kiếm ra màn hình (khuyến khích).

Bài 2: Dùng cây nhị phân tìm kiếm để thống kê số lần xuất hiện của từng ký tự trong một chuỗi nhập từ bàn phím.

Bài 3: Tạo một cây nhị phân tìm kiếm bằng cách đọc dữ liệu từ file. File được tổ chức như sau:

```
N //số nút trong cây
8 //giá trị của các nút
```

Bài 4:

a. Dùng cây nhị phân để thành lập 1 từ điển Anh - Việt. Gợi ý: Cấu trúc dữ liệu của 1 nút trên cây là:

```
typedef struct tagNode
{
    char Eword[10];
    char Vmeaning[20];
    struct tagavNode* pleft;
    struct tagavNode* pright;
}Node;
typedef Node *TREE;
```

b. Cho phép người dùng tra cứu từ điển trên cây này.

Chú ý: Dữ liệu được tổ chức lưu trữ trong file

Buổi 7

Bài 1: Tạo một cây nhị phân tìm kiếm cân bằng, trong đó trường Key tại mỗi nút là một số nguyên dương. Hãy viết hàm thực hiện các thao tác sau:

- i. In cây nhị phân tìm kiếm nói trên theo các thứ tự : LNR, LRN, NLR, NRL, RNL, RLN.
- j. Tìm một nút có khoá bằng X trên cây.
- k. Xóa 1 nút có khoá bằng X trên cây, nếu không có thì thông báo không có.
- l. Đếm số nút trong cây
- m. Đếm số nút có đầy đủ 2 cây con
- n. Đếm số nút lá.
- o. Tính chiều cao của cây
- p. Vẽ cây nhị phân tìm kiếm ra màn hình (khuyến khích)

Bài 2:

a. Dùng cây nhị phân để thành lập 1 từ điển Anh - Việt. Gợi ý: Cấu trúc dữ liệu của 1 nút trên cây là:

```
typedef struct tagAVNode
{
    char Eword[10];
    char Vmeaning[20];
    struct tagavNode* pleft;  struct tagavNode* pright;
}AVNode;
typedef AVNode *AVTREE;
```

b. Cho phép người dùng tra cứu từ điển trên cây này.
Chú ý: Dữ liệu được tổ chức lưu trữ trong file

Yêu cầu về các bài cần nộp:

Buổi 1: (1 hoặc 2) và (3 hoặc 4)

Buổi 2: 1 và (2 hoặc 3) và 4

Buổi 3: 1 và (2 hoặc 3)

Buổi 4: 1 và (2 hoặc 3) và (4 hoặc 5 hoặc 6 hoặc 7)

Buổi 5: 1 và 2 và 3 và 4 và (5 hoặc 6) (bài 5, 6 là bài cộng điểm, không bắt buộc)

Buổi 6: 1 và 2 và (3 hoặc 4)

Buổi 7: 1 hoặc 2 (Buổi 7 là bài cộng điểm, không bắt buộc)