Bài tập số 06

(Deadline: 02 tuần kể từ ngày giao – Ngày giao: Ngày 07 tháng 5 năm 2016)

Mục tiêu: Làm quen với một số cấu trúc dữ liệu cơ bản trong C

- *Làm việc với bản ghi liệt kê (enum)*
- *Làm việc với bản ghi (struct), con trỏ bản ghi, mảng bản ghi, một số biến thể của bản ghi*
- *Làm việc với ngăn xếp (stack), hàng đợi (queue), danh sách liên kết (link list), và cây nhị phân (binary tree)*
- *Rèn luyện kỹ năng áp dụng các kiến thức trên vào xây dựng chương trình giải quyết bài toán thực tế*

***Chú ý 1**: Với các yêu cầu, cần biết cách thiết kế và xây dựng hàm, cách sử dụng hàm đã xây dựng được*

***Chú ý 2**: SV là chữ số cuối cùng của thẻ sinh viên*

Chú ý 3**: Tên file mã nguồn chương trình vẫn theo quy định trong hướng dẫn. **Tuy nhiên, các bài được tạo thành 1 file hoàn chỉnh, các phần chỉ là các hàm con trong đó

Chú ý 4**: **Phần đầu mã mỗi chương trình, sử dụng cú pháp chú thích nhiều dòng /* */ thêm vào thông tin Họ và tên (tiếng Việt không dấu), Mã lớp, mã thẻ sinh viên

Phần I: Làm việc với typedef và enum

6.01. Định nghĩa một kiểu bản ghi liệt kê (enum) gồm các hằng ký hiệu, cho biết tên các hành tinh gần trái đất xung quanh mặt trời (Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto), sử dụng mảng con trỏ định nghĩa tên tiếng Việt của các hành tinh trên, mảng định nghĩa những miêu tả ngắn gọn về những hành tinh này (ví dụ: "duy nhất có sự sống", …). Yêu cầu người sử dụng nhập vào hành tinh mà người sử dụng quan tâm, in ra tên tiếng Việt mô tả đầy đủ.

6.02. Có một bản ghi liệt kê (enum) gồm các hằng ký hiệu, cho biết các mùa trong năm của miền Bắc Việt Nam được định nghĩa như sau:

```
enum season {SPRING, SUMMER, AUTUMN, WINTER};
```

Xây dựng chương trình sử dụng kiểu dữ liệu bản ghi liệt kê trên thực hiện công việc:

- Yêu cầu người sử dụng nhập vào một ngày tháng, trả về thông tin về mùa ở thời điểm ngày tháng đó (sử dụng kinh nghiệm bản thân để cho biết thời điểm của mùa bắt đầu, kết thúc, …)
- Sử dụng hàm thư viện lấy thời gian hệ thống trong thư <time.h>, lấy thời gian hệ thống hiện tại và trả về thông tin về mùa ở thời điểm thời gian lấy được trên (sử dụng kinh nghiệm bản thân để cho biết thời điểm của mùa bắt đầu, kết thúc, …)

Phần II : Làm việc với bản ghi (struct)

6.03. Bản ghi một sinh viên tham gia một môn học gồm các thông tin: Họ và tên (char:40); Mã SV (char: 20 - theo định dạng của PTIT); Mã lớp quản lý (char: 20 – theo định dạng của PTIT); Điểm thi HP (01 bài, dạng số thực, lẻ 0.5); Điểm kiểm tra giữa kỳ (01 bài, dạng số thực, lẻ 0.5); Điểm kiểm tra ngắn (>1 bài, <10 bài, dạng số thực, lẻ 0.5); Điểm điểm danh (>1 lần, < 5 lần, có giá trị 1 (có mặt), 0 (vắng mặt)); Điểm thưởng (>1 lần, có giá trị 1 (thưởng), 0 (không thưởng)); Điểm phạt (>=0 lần, có giá trị 1 (bị phạt), 0 (không bị phạt)); Điểm bài tập (>1 bài, <5 bài, là số thực, điểm lẻ 0.5); Điểm tổng kết môn học dạng số (số thực, lẻ 0.5); Điểm tổng kết môn học dạng chữ (là các chuỗi: A+, A0, A-, B+, B0, B-, …). Các điểm (ngoại trừ điểm thưởng, điểm phạt và điểm điểm danh) sử dụng thang hệ điểm 10.

- Thông tin về trọng số của các điểm và đánh giá các điểm như sau: Điểm thi HP chiếm 70% điểm tổng kết môn học; Điểm kiểm tra giữa kỳ chiếm 10%, Điểm chuyên cần chiếm 10%, Điểm bài tập chiếm 10%;
- Điểm kiểm tra giữa kỳ được tính: Điểm bài kiểm tra giữa kỳ 50%, điểm trung bình các bài kiểm tra ngắn chiếm 50%
- Điểm chuyên cần: Điểm điểm danh chiếm 40%, điểm thưởng (= điểm thưởng – điểm phạt) chiếm 60%
- Điểm bài tập: Điểm trung bình của các bài tập được giao
- Điểm tổng kết: 2% sinh viên của lớp được A+, 10% được A0, 10% được A-, 10% được B+, 10% được B0, 10% được B-; 20% được C+, 20% được C0, 6% được C-, 2% được D0
- Điểm tổng kết dạng số theo đúng quy ước điểm tín chỉ, lấy mức giữa của khoảng điểm (ví dụ 9.5 – 10 là A+ thì khi được A+)

Viết chương trình thực hiện những công việc sau:

1) Định nghĩa bản ghi cho sinh viên với thông tin trên
2) Xây dựng hàm nhập thông tin sinh viên, thông tin được nhập liên tục cho đến hết danh sách (chọn tín hiệu điều khiển kết thúc nhập thích hợp), thông tin nhập được ghi vào file __**kiểu text/kiểu nhị phân**_____ với tên

file là MaSVIn.* (phần mở rộng * được chọn thích hợp, ví dụ file kiểu text sẽ là txt, dat, …). Thông tin giả lập của sinh viên có thể là danh sách sinh viên trong lớp đang tham gia, điểm có thể là các điểm giả lập (ngẫu nhiên trong các khoảng cho phép). Riêng điểm tổng kết chưa có. Việc ghi file được tiến hành tuần tự.

3) Xây dựng hàm đọc thông tin sinh viên từ file trong câu 2, in thông tin sinh viên ra màn hình
4) Xây dựng hàm đọc thông tin sinh viên từ file trong câu 2, thực hiện việc tính điểm tổng kết cho sinh viên theo cách tính ở trên. Kết quả tính toán cùng với thông tin các sinh viên được ghi vào một file ___**kiểu text/kiểu nhị phân** __ với tên file là MaSVOut.* Việc ghi file được tiến hành tuần tự.
5) Xây dựng hàm tìm kiếm: Với mã một sinh viên được cung cấp, hàm cho biết có hay không có sinh viên có mã đó trong lớp. Nếu có, in ra các thông tin sinh viên cùng điểm của sinh viên đó
6) Xây dựng hàm tìm điểm tổng kết môn học lớn nhất và nhỏ nhất. Cho biết sinh viên nào có điểm lớn nhất, điểm nhỏ nhất. Nếu có nhiều hơn một sinh viên, in tất cả các sinh viên đó. Hàm có hai lựa chọn, in ra màn hình hoặc in vào một file **___kiểu text / kiểu nhị phân ___** với tên lần lượt là MaSVMaxOut.* và MaSVMinOut.*
7) Xây dựng hàm tính điểm tổng kết trung bình của các sinh viên
8) Xây dựng hàm tìm điểm trung vị của các sinh viên
9) Xây dựng hàm tính phần trăm sinh viên được điểm trên một mức nào đó: Với 1 mức điểm nào đó, tính phần trăm số sinh viên có điểm trên mức điểm đó. Việc tính toán này dựa trên điểm tổng kết hệ cơ số 10, trước khi chuyển điểm tín chỉ
10) Xây dựng hàm sắp xếp danh sách sinh viên theo điểm tổng kết môn học theo thứ tự giảm dần / tăng dần. In kết quả sắp xếp ra thiết bị mong muốn (màn hình hoặc file)
11) Xây dựng hàm sắp xếp danh sách sinh viên theo điểm bài kiểm tra kết thúc môn học (còn gọi là bài kiểm tra giữa kỳ). In kết quả sắp xếp ra thiết bị mong muốn (màn hình hoặc file)

6.04. Việc thanh toán tiền điện sinh hoạt của các hộ dân được thực hiện theo tháng. Danh sách thanh toán tiền điện của một Sở điện A là thông tin: tháng thu, mã khách hàng, tên khách hàng, địa chỉ, chỉ số điện năng tiêu thụ. Xây dựng chương trình:

1) Nhập thông tin của các khách hàng mà Sở điện A quản lý và phụ trách. Thông tin nhập được được lưu vào file DSTDienIn.* (theo kiểu ___ tuần tự / ngẫu nhiên__)
2) In thông tin của các khách hàng đã nhập được từ file tạo bởi câu 1) ra màn hình
3) Thực hiện tín tiền điện mà khách hàng phải trả của tháng, tiền điện được tính theo cách tính trong bài 2.14. In hóa đơn cho mỗi khách hàng ra (___ màn hình / file ___)
4) Tính tổng số tiền mà Sở điện A thu được trong tháng
5) Tìm khách hàng có mức tiền tiêu thụ điện lớn nhất; nhỏ nhất trong các khách hàng mà sở phụ trách
6) Nhập vào một mức tiền tiêu thụ điện nào đó, cho biết tỷ số phần trăm khách hàng có số tiền điện tiêu thụ phải trả lớn hơn mức đó

<u>Phần III</u>: Làm việc với biến thể của bản ghi – union và bản ghi trường bít (bit fields)

6.05. Người ta mong muốn có một biến vừa có thể sử dụng như một số nguyên, vừa có thể sử dụng như một số thực (dạng float), vừa có thể sử dụng như một mảng ký tự có độ dài 60 ký tự. Hãy định nghĩa một kiểu dữ liệu cấu trúc union thỏa mãn điều đó, sử dụng minh họa việc làm việc với kiểu dữ liệu định nghĩa trên

6.06. Một màn hình LCD 16x2 có thể được điều khiển bởi các chân RS, EN, RW (mỗi chân tương ứng dữ liệu độ rộng 1 bit) và dữ liệu được chuyển DB0, DB1, DB2, DB3, DB4, DB5, DB6, và DB7 (mỗi đường dữ liệu ứng với độ rộng 1 bit).

- Hãy định nghĩa bản ghi trường ghi bít (bit fields) mô tả thông tin các chân điều khiển màn hình LCD trên
- Viết chương trình thực hiện các công việc điều khiển LDC với kiểu dữ liệu trường bản ghi ở trên và cấu trúc thực hiện các lệnh điều khiển LCD (tham khảo tại https://en.wikipedia.org/wiki/Hitachi_HD44780_LCD_controller) – Tham khảo cách thiết lập các đường dữ liệu như hình vẽ.

PIN No	Name	Function
1	VSS	Ground voltage
2	VCC	+5V
3	VEE	Contrast voltage
4	RS	Register Select 0 = Instruction Register 1 = Data Register
5	R/W	Read/ Write, to choose write or read mode 0 = write mode 1 = read mode
6	E	Enable 0 = start to latch data to LCD character 1= disable
7	DB0	Data bit 0 (LSB)
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7 (MSB)
15	BPL	Back Plane Light +5V or lower (Optional)
16	GND	Ground voltage (Optional)

JHD 162A Pin Configuration

Phần IV: Các cấu trúc dữ liệu khác

6.07. Thực hiện việc đọc tài liệu Deteil C How to program, phần Chương 12, mục 12.2 và 12.4. Xây dựng một danh sách liên kết – link list (tham khảo mã trong Fig 12.3). Làm việc và mở rộng công việc:

- Xây dựng 03 hàm chèn mới: một hàm cho phép luôn chèn lên đầu danh sách, một hàm cho phép luôn chèn xuống cuối danh sách, một hàm chèn vào danh sách để tạo thành một danh sách có các giá trị khóa giảm dần
- Xây dựng hàm tìm kiếm xem một giá trị nào đó có tồn tại trong danh sách liên kết không
    + Mở rộng: hàm tìm kiếm nếu thấy cho biết vị trí node tìm thấy cách node đầu tiên mấy node
- Tạo một chuỗi các số ngẫu nhiên và kiểm tra các hàm đã xây dựng ở trên

6.08. Thực hiện việc đọc tài liệu Deteil C How to program, phần Chương 12, mục 12.5. Xây dựng một ngăn xếp – stack (tham khảo mã trong Fig 12.8). Làm việc với ngăn xếp đó

6.09. Thực hiện việc đọc tài liệu Deteil C How to program, phần Chương 12, mục 12.6. Xây dựng một hàng đợi – queue (tham khảo mã trong Fig 12.13). Làm việc với hàng đợi đó

6.10. Thực hiện việc đọc tài liệu Deteil C How to program, phần Chương 12, mục 12.7. Xây dựng một cây nhị phân – binary tree (tham khảo mã trong Fig 12.19). Làm việc với cây nhị phân đó

Phần V: Một số bài tập mở rộng

(Dành riêng cho sinh viên ngành CN)

6.11. [*Xử lý dữ liệu chuỗi*] Để thuận lợi cho việc nhập dữ liệu, đặc biệt khi làm việc với mảng, nhiều ngôn ngữ lập trình hỗ trợ việc nhập linh động như sau (ví dụ với Matlab):

>>a=-1:0.5:1;

Cho ta một véc-tơ (có thể coi là mảng) gồm các giá trị từ -1 đến 1, mỗi phần tử khác nhau 0.5 đơn vị. Định dạng: Start:Step:Stop

>>a=0:9

Cho ta một véc-tơ (có thể coi là mảng) gồm các giá trị từ 0 đến 9, mỗi phần tử khác nhau 1 đơn vị. Định dạng: Start:Stop, mặc định Step = 1

>> a =[1 2 3 4 5];

Cho ta một véc tơ (có thể coi là ma trận 1x5, hay véc-tơ), gồm các giá trị 1, 2, 3, 4, và 5. Định dạng: các phần từ đặt trong dấu [và], cách nhau bởi ít nhất một dấu cách hay một dấu phẩy (,)

Xây dựng chương trình khi người sử dụng nhập dữ liệu vào với các kiểu định dạng trên sẽ trả lại cho chúng ta véc-tơ như mô tả.

6.12. Trong việc thực hiện duyệt chuỗi (khi xử lý ngôn ngữ, văn bản, … trong các ngôn ngữ như Prolog), sẽ rất thuận lợi nếu chúng ta tổ chức một biểu thức ở dạng postfix (còn gọi là Reverse Polish Notation), cùng với nó là biểu diễn ở dạng prefix (còn gọi là Polish Notation – kí hiệu Ba lan). Tham khảo tại:

http://www.cs.man.ac.uk/~pjj/cs212/fix.html

https://cs.nyu.edu/courses/Fall12/CSCI-GA.1133-002/notes/InfixToPostfixExamples.pdf

http://csis.pace.edu/~wolf/CS122/infix-postfix.htm

Xây dựng một chương trình gồm các hàm thực hiện việc chuyển đổi một biểu thức giữa các dạng prefix, infix, và postfix