

Kỹ Thuật Lập Trình

Lab 3 – Nhập xuất và sử dụng thư viện Các cấu trúc điều khiển

1 MỤC TIÊU CỦA BÀI THỰC HÀNH

- Nắm được một số lệnh nhập xuất và biết cách sử dụng một vài thư viện cơ bản.
- Hiểu và vận dụng các cấu trúc điều khiển rẽ nhánh để giải quyết trọn vẹn 1 bài toán hay một phần các bài toán cần đến dạng cấu trúc này
- Luyện kỹ năng giải quyết vấn đề thông qua quá trình phân rã bài toán, vét cạn các trường hợp.

2 BÀI TẬP BẮT BUỘC

Câu 1:

Cho một đường tròn tâm $O(x_0, y_0)$ và bán kính R . Viết chương trình kiểm tra xem một điểm $A(x, y)$ là nằm trên, trong hay ngoài đường tròn đó. Cụ thể, chương trình cho phép người lập trình nhập vào:

- Tọa độ điểm O và bán kính R . Nếu người dùng nhập R là số âm \rightarrow chương trình thông báo lỗi và cho nhấn ENTER để kết thúc
- Nhập vào tọa độ điểm A muốn kiểm tra. Chương trình thực hiện kiểm tra và in kết quả

Hướng dẫn:

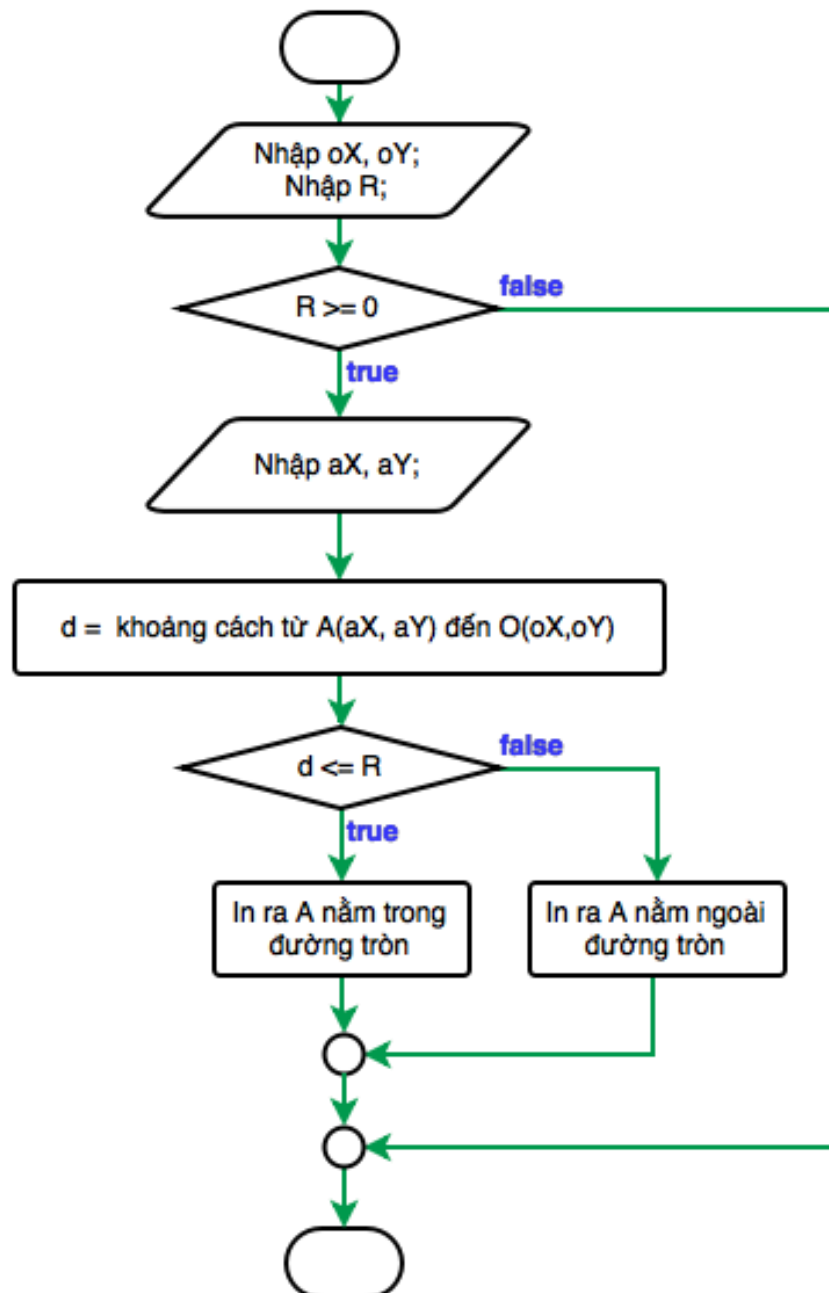
a) **Dữ liệu:** chương trình phải có các biến (ô nhớ) chứa các dữ liệu sau đây

- Tọa độ x và y của điểm O : kiểu double. Tên biến là oX, oY
- Tọa độ x và y của điểm A : kiểu double. Tên biến là aX, aY
- Bán kính R : kiểu double

b) **Giải thuật:**

Điểm A nằm trong đường tròn tâm O bán kính R nếu như khoảng cách từ A đến tâm O nhỏ hay bằng bán kính R (nằm trên cũng xem như trong). Ngược lại, điểm A nằm ngoài.

- ⇒ Cần phải tính khoảng cách giữa hai điểm A và O, gọi là d
- ⇒ Cần dùng hàm: sqrt
- ⇒ Dùng thư viện <math.h>
- ⇒ Dùng chỉ thị #include <math.h> ở đầu tập tin
- Các trường hợp ở đây là : $R \leq d$ và ngược lại



(Bản vẽ được phát thảo bởi công cụ online: <https://www.draw.io/>)



Câu 2:

Giả sử ngày hiện tại là Chủ nhật. Viết chương trình nhập vào một số X và cho biết xem sau X ngày từ ngày hiện tại là thứ mấy trong tuần.

Hướng dẫn:

a) **Dữ liệu:** chương trình phải có biến (ô nhớ) để chứa các dữ liệu sau đây:

- X : số không âm → kiểu unsigned int hay unsigned long long

b) **Giải thuật:**

Quan sát:

- | | |
|--|--------------------------|
| • Nếu x = 0 thì chính là ngày hiện tại | → Chủ nhật |
| • Nếu x = 1 là ngày kế tiếp | → Thứ 2 |
| • Nếu x = 2 | → Thứ 3 |
| • Nếu x = 3 | → Thứ 4 |
| • Nếu x = 4 | → Thứ 5 |
| • Nếu x = 5 | → Thứ 6 |
| • Nếu x = 6 | → Thứ 7 |
| • Nếu x = 7 | → Chủ nhật (lặp trở lại) |
| • Nếu x = 8 | → Thứ 2 |
| • ... | |

Kết luận:

Cứ sau 6 ngày thì lặp lại (Ngày thứ 7, tính từ 1, thì lặp lại)

→ Tính X chia cho 7 lấy phần dư ($X \% 7$). Phần dư 0 thì chính là Chủ nhật, 1 là Thứ 2, v.v.

→ Có thể dùng phát biểu if hay switch-case. Trường hợp này switch –case trong sáng hơn vì nó xét trường hợp hữu hạn

→ Học viên luyện tập với cả if-else và switch-case

Câu 3:

Viết chương trình cho độ dài 3 cạnh a, b, c, kiểm tra xem nó có phải là 3 cạnh của một tam giác cân, tam giác đều, tam giác vuông hay tam giác vuông cân không.

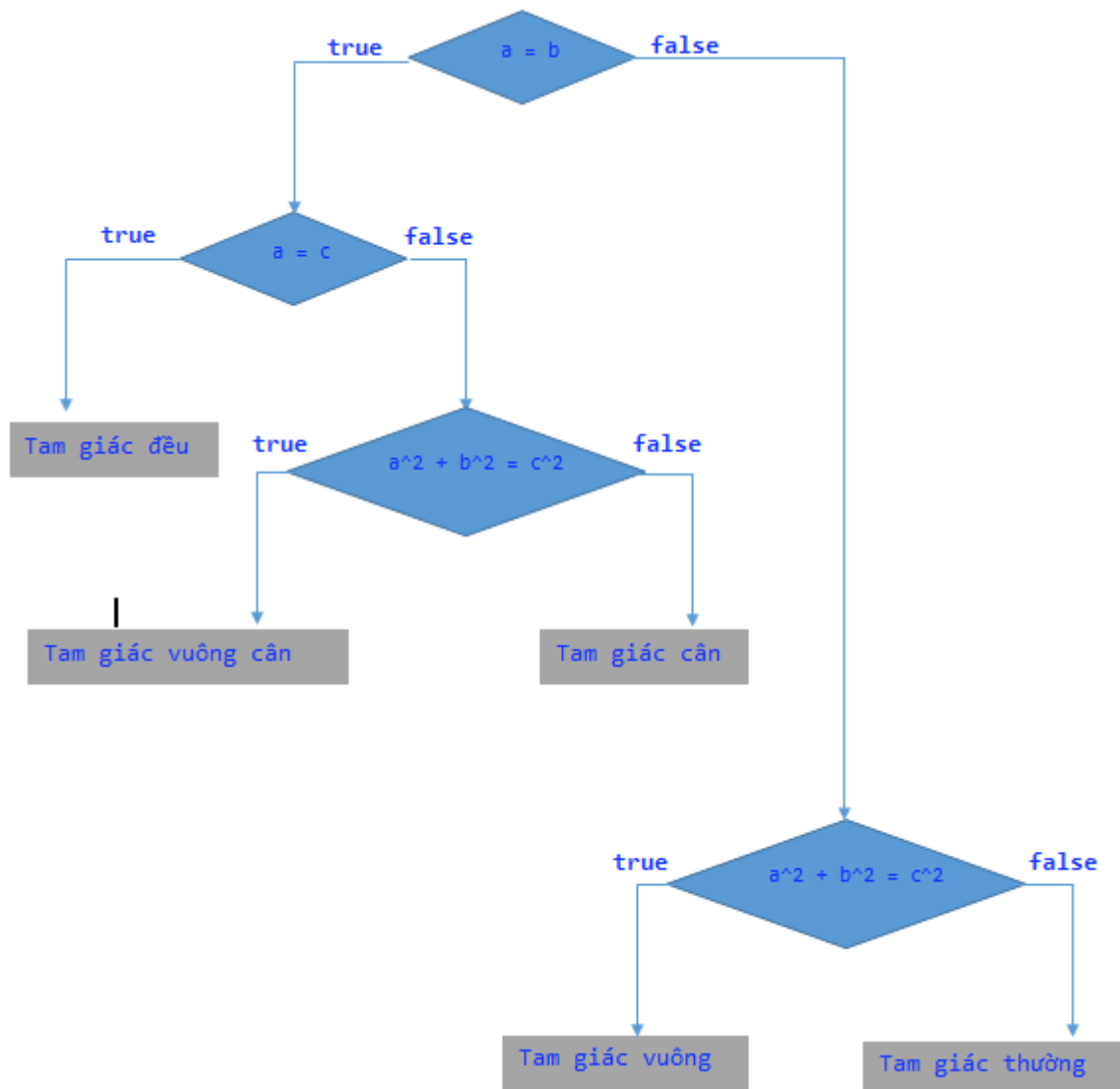
Hướng dẫn:

a) **Dữ liệu:** chương trình phải có biến (ô nhớ) để chứa các dữ liệu sau đây:

- a, b, và c: kiểu double hoặc float

b) Giải thuật:

- Kiểm tra xem a, b, c có thành lập tam giác hay không?
- Liệt kê đầy đủ các cặp cạnh: có 3 cặp cạnh: cặp (a, b), (a, c) và (b, c)
- Với mỗi cặp cạnh, ví dụ cặp (a, b) ta xét các trường hợp có thể có. Các trường hợp có thể có của cặp (a, b) được liệt kê như hình sau đây.
- Mỗi khi tam giác đã được xếp vào một trong 4 loại ĐỀU, CÂN, VUÔNG CÂN, VUÔNG thì không cần xét các cặp cạnh còn lại (a, c) và (b, c).
- Chỉ xét tiếp các cặp cạnh chưa xét, (a, c) và (b, c), khi loại của tam giác là THƯỜNG. Tuy nhiên, khi xét các cặp còn lại này thì không cần phải kiểm tra trường hợp ĐỀU NỮA, vì nó đã kiểm tra qua rồi khi xét cặp cạnh (a, b).



c) So sánh bằng với số thực:

Khi làm việc với các kiểu số thực (float, double hay long double), KHÔNG nên so sánh bằng (==) giữa các trị có kiểu số thực. Lý do, nhiều trường hợp máy tính không thể biểu diễn đúng con số theo yêu cầu, vì độ chính xác biểu diễn của máy tính có giới hạn. Ví dụ, máy tính không thể biểu diễn con số vô tỉ như π , $\sqrt{3}$, v.v.

Do đó, để so sánh hai số thực a và b cần tìm khoảng cách giữa chúng ($\text{abs}(a-b)$) và so sánh với một hằng số đủ nhỏ nào đó. Nếu khoảng cách nhỏ hơn hằng số này thì xem như a bằng b và ngược lại.

Với số float: hằng này có thể là

```
#define EPSILON 1.0E-6
```

Với số double: hằng này có thể là

```
#define EPSILON 1.0E-13
```

Lưu ý:

- (1) Sinh viên nên định nghĩa macro-function để kiểm tra tính bằng giữa hai tham số a và b , và sử dụng macro này để so sánh hai số hay hai biểu thức bất kỳ khi cần thiết (xem các phép so sánh bằng trong sơ đồ).
- (2) Sinh viên nên sử dụng kiểu enum để định nghĩa (quy ước) các trường hợp của tam giác. Ví dụ, $\text{TAM_GIAC_DEU} = 0$, TAM_GIAC_VUONG , v.v.

Câu 4:

Viết chương trình cho phép người dùng nhập vào các hệ số a , b , và c của một phương trình có bậc tối đa là 2 (a và b có thể bằng 0)

Giải và in ra kết quả.

Hướng dẫn:

Xem slide bài giảng

Câu 5:

Viết chương trình nhập vào:

- a. Ngày ghi chỉ số điện tháng trước (dd:mm:yyyy) + Chỉ số điện tháng trước (số kWh)
- b. Ngày ghi chỉ số điện tháng hiện tại + Chỉ số điện kỳ hiện tại.

c. Số hộ dùng chung trong một diện (1, 2, ...)

Chương trình tính và in ra hoá đơn như hình sau:

Phương pháp tính giá tiền trong hoá đơn được giải thích sau đây:

- Cột “Đơn giá” là giá tiền cho mỗi KWh theo từng mức bậc thang (có 6 mức)
- Cột “Định mức cơ bản” (DMCB) là số KWh theo quy định được hưởng giá theo mức bậc thang tương ứng.
- Cột “Định mức của khách hàng” (DMKH) là định mức cho từng khách hàng (đồng hồ). Cột này được tính từ cột “Định mức cơ bản” theo công thức sau:

$$DMKHi = DMCB * (N/T) * h$$

- Ở đó, N: số ngày tính tiền = số ngày từ ngày ghi hoá đơn kỳ trước đến ngày ghi hiện tại (không kể ngày hiện tại)
- T: Số ngày theo lịch của tháng trước liền kề. Ví dụ, ngày ghi hiện tại thuộc tháng 3 => số ngày tháng trước là số ngày tháng 2 => 28 hay 29 ngày.
- h: Số hộ dùng chung đồng hồ.
- Cột “ Sản lượng ...” là số KWh được tính theo mức bậc thang tương ứng (không vượt định mức của cột định mức của khách hàng)
- Cột số tiền là số tiền tính theo từng mức bậc thang

Ví dụ trong hình:

- Ngày ghi kỳ trước: 11/01/2016, chỉ số: 1000 KWh
- Ngày ghi kỳ hiện tại: 03/02/2016, chỉ số: 1236 KWh
- Số hộ dùng chung: 1
 - ⇒ N = 24 ngày
 - ⇒ T = 31 ngày (Tháng 01)

STT	Đơn giá (đồng)	Định mức cơ bản (kWh)	Định mức của khách hàng	Sản lượng của khách hàng	Số tiền (đồng)
1	1.484	50	39	39	57.876
2	1.533	50	39	39	59.787
3	1.786	100	77	77	137.522
4	2.242	100	77	77	172.634
5	2.503	100	77	4	10.012
6	2.587	Còn lại	Còn lại	0	0
				236 kWh	437.831
Thuế GTGT (10%):					43.783
Tổng cộng:					481.614

Hướng dẫn:

a) **Dữ liệu:** chương trình phải có biến (ô nhớ) để chứa các dữ liệu sau đây:

- Ngày ghi chỉ số điện tháng trước: kiểu dữ liệu int (tốt nhất là unsigned char)
- Tháng ghi chỉ số điện tháng trước: kiểu dữ liệu int (tốt nhất là unsigned char)
- Năm ghi chỉ số điện tháng trước: kiểu dữ liệu int (tốt nhất là unsigned int)
- Ngày ghi chỉ số điện tháng hiện tại: kiểu dữ liệu int
- Tháng ghi chỉ số điện tháng hiện tại: kiểu dữ liệu int
- Năm ghi chỉ số điện tháng hiện tại: kiểu dữ liệu int
- Chỉ số điện tháng trước: kiểu dữ liệu long long
- Chỉ số điện tháng hiện tại: kiểu dữ liệu long long
- Số ngày tính tiền: kiểu dữ liệu int
- Số ngày theo lịch của tháng trước liền kề: kiểu dữ liệu int
- DG (đơn giá) và các DMCB có thể dùng câu lệnh define để định nghĩa

b) **Giải thuật:**

Đây là bảng trường hợp diễn hình dùng cấu trúc if-else nối nhau (lồng nhau)

Trước tiên phải tính DMKH1 từ công thức: $DMKH1 = DMCB * (N/T) * h$

Bảng trên được hiểu là:

Nếu SLKH (sản lượng khách hàng) \leq DMKH1 thì số tiền được tính theo công thức : $DG1 * SLKH$

Ngược lại: *nghĩa là $SLKH > DMKH1$. Do đó, cần kiểm tra xem nó có nằm trong khoảng kế tiếp hay không, nghĩa là khoảng $(DMKH1, DMKH1+DMKH2]$*

Nếu $SLKH \leq DMKH2$ thì số tiền được tính theo công thức: $DG1 * DMKH1 + DG2 * (SLKH - DMKH1)$

Ngược lại: *nghĩa là $SLKH > DMKH2$. Do đó, cần kiểm tra xem nó có nằm trong khoảng kế tiếp hay không, nghĩa là khoảng $(DMKH1+DMKH2, DMKH1+DMKH2+DMKH3]$*

(làm tương tự)

Cuối cùng lấy kết quả sinh ra cộng thêm số tiền phải đóng thuế rồi in kết quả tiền điện phải đóng của hộ gia đình.

Câu 6:

Viết chương trình có chức năng

(a) In ra các menu sau và dòng hướng dẫn lựa chọn như sau:

1. Nhập hàng hoá
2. Tìm hàng hoá
3. In ra danh sách hàng hoá
4. Xoá hàng hoá
5. Cập nhật hàng hoá
6. Lưu dữ liệu
7. Tải dữ liệu
8. Thoát

Hãy chọn chức năng:

(b) Đọc menu được chọn từ người dùng. In ra tên chức năng tương ứng theo bảng trên và chờ nhấn ENTER để kết thúc. Nếu nhập không đúng thì in ra dòng thông báo lỗi và chờ nhấn ENTER để kết thúc.

Hướng dẫn:

Đây là bài toán điển hình của cấu trúc switch-case. Với cấu trúc này, nên dùng typedef để tạo kiểu dữ liệu mới là enum chứa danh sách mã các trường hợp.

c) Kiểu dữ liệu:

```
typedef enum menu {  
    NHAP_HANG_HOA = 0,  
    TIM_HANG_HOA,  
    IN_DS_HANG_HOA,  
    v.v.  
}
```

d) **Dữ liệu:** chương trình phải có biến (ô nhớ) để chứa các dữ liệu sau đây:

- Lựa chọn của người dùng (biến : choice): kiểu dữ liệu unsigned char

e) **Giải thuật:**

- In danh sách menu → dùng hàm printf. Mỗi lệnh tương ứng 01 menu. Nên canh lề trái
- In ra dòng hướng dẫn chọn, không xuống hàng
- Đọc giá trị từ bàn phím gán cho biến choice
- Dùng lệnh switch case theo mã trường hợp để in ra tên tính năng tương ứng. Dùng **break;** để thoát khỏi switch

Câu 7: (dùng switch-case)

Viết chương trình nhập vào một chuỗi có định dạng là một biểu thức có dạng: (value1 op value2). Ở đó, value1 và value2 là hai số, op là một trong các phép toán sau: +, -, * và /

Chương trình tính và in ra kết quả của biểu thức.

Ví dụ:

Nhập: 13 + 45

In ra: 58

Hướng dẫn:

a) **Dữ liệu:**

Sử dụng các thư viện `<string>` và `<sstream>` để thao tác nhập xuất trên chuỗi ký tự

Sử dụng hàm `getline()` để lấy chuỗi ký tự từ bàn phím vào

Dùng `stringstream` để tiến hành phân tách các thành phần dữ liệu sau:

- operand_1: hạng tử đầu tiên → kiểu số nguyên
- operand_2: hạng tử thứ hai → kiểu số nguyên
- operator: phép toán → kiểu ký tự
- result: kết quả → kiểu nguyên

b) Giải thuật:

- Sử dụng cấu trúc switch-case để xét các trường hợp của phép tính được nhập vào.
- Xử lý phép tính trong các case của switch, gán kết quả vào biến result
- Xuất kết quả ra màn hình