

BÀI THỰC HÀNH SỐ 4 :: CẤU TRÚC Rẽ NHÁNH - IF

MỤC TIÊU

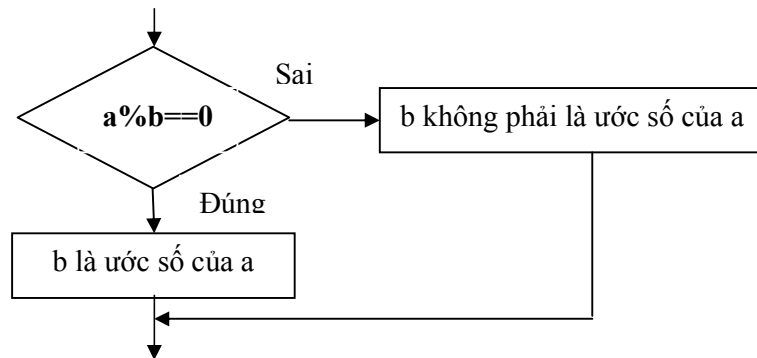
1. Diễn đạt được ngữ cảnh rẽ nhánh bằng ngôn ngữ tự nhiên và sơ đồ khối.
2. Vận dụng cấu trúc `if()` hoặc `if() else` để viết câu lệnh thực hiện ngữ cảnh rẽ nhánh.
3. Phân biệt câu lệnh và khối lệnh (lệnh ghép)

I. NỘI DUNG THỰC HÀNH

Vấn đề 1: *Diễn đạt rẽ nhánh: nếu ... thì...ngược lại...*

1.1. Viết chương trình nhập vào 2 số nguyên a và b ($b \neq 0$). Thông báo lên màn hình b có phải là ước số của a ? Lưu vào thư mục **LAB04** với tên file ***Divisor.CPP***

Hướng dẫn:



1.2. Biểu thức $a \in [A, B]$ được diễn đạt bằng ngôn ngữ tự nhiên như sau: *Nếu a lớn hơn hoặc bằng A và a bé hơn hoặc bằng B thì ta nói rằng a thuộc đoạn A và B , ngược lại ta nói a không thuộc đoạn A và B .*

- a) Viết chương trình nhập lần lượt các giá trị a , A và B . Thông báo lên màn a thuộc/không thuộc đoạn A và B .
- b) Viết chương trình nhập vào 1 ký tự **ch** từ bàn phím. Thông báo lên màn hình **ch** là chữ HOA hoặc **ch** là chữ thường hoặc **ch** chữ số?

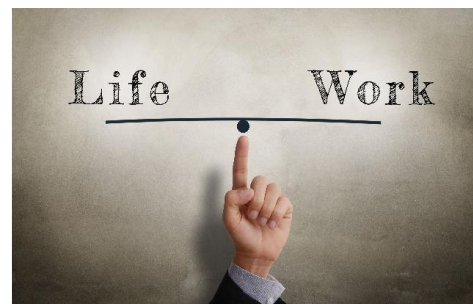
Yêu cầu: Trình bày bảng *test_case*.

Vấn đề 2: *Vận dụng if()...else...*

Viết chương trình nhập vào **số_ngày_làm_việc** trong tháng của một nhân viên sau đó thông báo lên màn hình số tiền lương mà họ nhận được (Lưu vào thư mục **LAB04** với tên file ***Salary.CPP***). Biết rằng công thức tính lương của công ty như sau:

$$[\text{Lương}] = \text{số_ngày_công} \times 300.000$$

Trong đó: Nếu **số_ngày_làm_việc** lớn hơn 22 thì những ngày dư ra (gọi là ngày làm việc ngoài giờ) được tính bằng 2 ngày công. Ví dụ; nhân viên A làm việc 30 ngày trong tháng 7 thì số ngày công của A là: $22 + (30-22) \times 2 = 38$

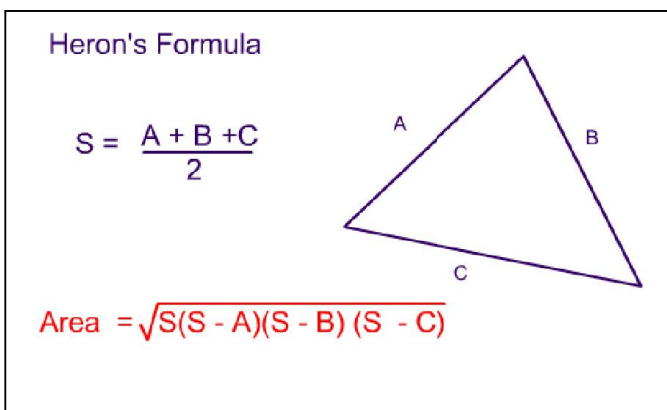


Gợi ý thuật toán:

1. Nhập số_ngày_làm_việc.
2. Tính số_ngày_công:
 Nếu (số_ngày_làm_việc > 22) thì số_ngày_công = 22 + (số_ngày_làm_việc - 22)*2
 Ngược lại, số_ngày_công = số_ngày_làm_việc;
3. Tính và thông báo Tiền_lương theo công thức [Lương]

Vấn đề 3: Diễn đạt ngữ cảnh rẽ nhánh và vận dụng if() ...else... để viết chương trình

- 3.1. Cho 3 số dương a, b, và c. Với điều kiện nào thì:
 - a) a, b và c là các cạnh của một tam giác?
 - b) a, b và c là các cạnh của một tam giác đều?
 - c) a, b và c là các cạnh của một tam giác cân?
 - d) a, b và c là các cạnh của một tam giác vuông?
- 3.2. Viết chương trình nhập vào 3 số dương a, b và c từ bàn phím. Nếu chúng là các cạnh của một tam giác thì thông báo lên màn hình diện tích của tam giác đó, ngược lại thông báo không tồn tại tam giác với bộ cạnh a, b và c.
- 3.3. Viết chương trình nhập vào 3 số dương a, b và c. Thông báo lên màn hình tính chất “vuông”, “đều”, “cân” của tam giác nếu có.
Yêu cầu: Xây dựng bảng dữ liệu test_case

Vấn đề 4: Sử dụng if() else lồng nhau.

Viết chương trình nhận vào 3 số nguyên lần lượt tương ứng với h:m:s của một đồng hồ điện tử. Hỏi sau 1 giây, màn hình đồng hồ hiển thị như thế nào? Lưu vào thư mục **LAB04** với tên file **Clock.CPP**

Gợi ý dữ liệu test đúng như sau:

Test	input				output
	h	m	s	Đồng hồ ban đầu	Đồng hồ sau 1 giây
1	1	2	3	1:2:3	1:2:4
2	1	2	59	1:2:59	1:3:0
3	1	59	59	1:59:59	2:0:0
4	23	59	59	23:59:59	0:0:0

Ghi chú: Khi đủ 60 giây đồng hồ sẽ chuyển thành 1 phút, đủ 60 phút chuyển thành 1 giờ

Vấn đề 5: Sử dụng if() else lồng nhau để loại trừ điều kiện.

Viết chương trình (tên file **Credit.cpp** lưu vào **LAB04**) nhập vào điểm thi môn NHẬP MÔN TIN HỌC theo thang điểm 10 sau đó chuyển sang thang điểm 4 và điểm chữ (A, B+, B, C+, C, D+, D, F) tương ứng (xem bảng sau)

Điểm thang điểm 4	Điểm chữ
mark = 4.00	A
$3.50 \leq \text{mark} < 4.00$	B+
$3.00 \leq \text{mark} < 3.50$	B
$2.50 \leq \text{mark} < 3.00$	C+
$2.00 \leq \text{mark} < 2.50$	C
$1.50 \leq \text{mark} < 2.00$	D +
$1.00 \leq \text{mark} < 1.50$	D
mark < 1.00	F

Dữ liệu vào từ màn hình:

- Đọc vào điểm thi nằm trong khoảng 0..10 (số thực)

Kết quả ra màn hình:

- Tương ứng với mỗi điểm số (*mark*) nhập từ bàn phím nằm trong [0..10], chuyển sang điểm 4 và cho biết điểm chữ tương ứng.
- Trường hợp điểm số (*mark*) nhỏ hơn 0 hoặc lớn hơn 10 thì in ra **Khong hop le**

input	output
mark = 0	Diem theo thang 4 la 0.00 - Diem F
mark = 5	Diem theo thang 4 la 2.00 - Diem C
mark = 7.5	Diem theo thang 4 la 3.00 - Diem B
mark = 8	Diem theo thang 4 la 3.20 - Diem B+
mark = 10	Diem theo thang 4 la 4.00 - Diem A
mark = -1	Khong hop le
mark = 11	Khong hop le

II. BÀI TẬP

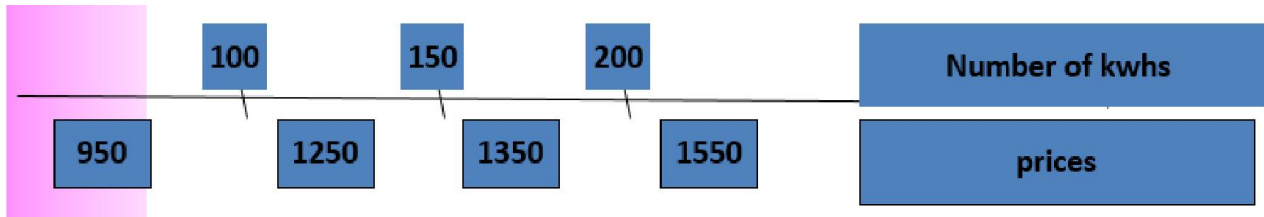
Bài 0021



Yêu cầu: Viết chương trình nhập vào số áo mua, tính tiền phải trả.

Bài 0022

Tính tiền phải trả khi sử dụng điện, biết rằng các giá điện (*prices*) được cho theo các cột mốc của số kw tiêu thụ (*Number of kwhs*) như sau:



Yêu cầu:

- Diễn đạt ngữ cảnh tính tiền điện bằng ngôn ngữ tự nhiên
- Viết chương trình nhập vào số kw điện tiêu thụ và hiển thị lên màn hình số tiền phải trả.
- Lập bảng dữ liệu kiểm thử

Bài 0023

Viết chương trình nhập tháng, năm. Sau đó tính ra số ngày trong tháng đó. Biết rằng năm nhuận là năm chia hết cho 4 nhưng không chia hết cho 100, hoặc chia hết cho 400.

Bộ dữ liệu test:

Input		Output
Tháng	Năm	Số ngày trong tháng
2	2011	28
2	1900	28
2	2000	29
3	2011	31
13	2012	Nhập sai

Hướng dẫn:

- Viết biểu thức điều kiện để kiểm tra một năm có nhuận hay không: $nhuan = <BT>$;
- Nếu (nhuận) thì tháng 2 có 29 ngày, ngược lại tháng 2 có 28 ngày
- Các tháng 1,3,5,7,8,10,12 có 31 ngày, các tháng còn lại (trừ tháng 2) có 30 ngày.

Bài 0024

Viết chương trình nhập điểm trung bình và xếp loại học lực theo tiêu chí sau:

- $9 \leq \text{ĐTB}$: Xuất Sắc
- $8 \leq \text{ĐTB} < 9$: Giỏi
- $7 \leq \text{ĐTB} < 8$: Khá
- $6 \leq \text{ĐTB} < 7$: Trung Bình Khá
- $5 \leq \text{ĐTB} < 6$: Trung Bình
- $\text{ĐTB} < 5$: Yếu

Yêu cầu: Kiểm tra tính hợp lệ của dữ liệu điểm (dữ liệu sai nếu điểm âm hoặc lớn hơn 10)

Bộ dữ liệu test:

Input	Output
Điểm Trung Bình	Xếp loại
8.69	Giỏi
7.0	Khá
2.0	Yếu
11.5	Nhập sai

Bài 0025

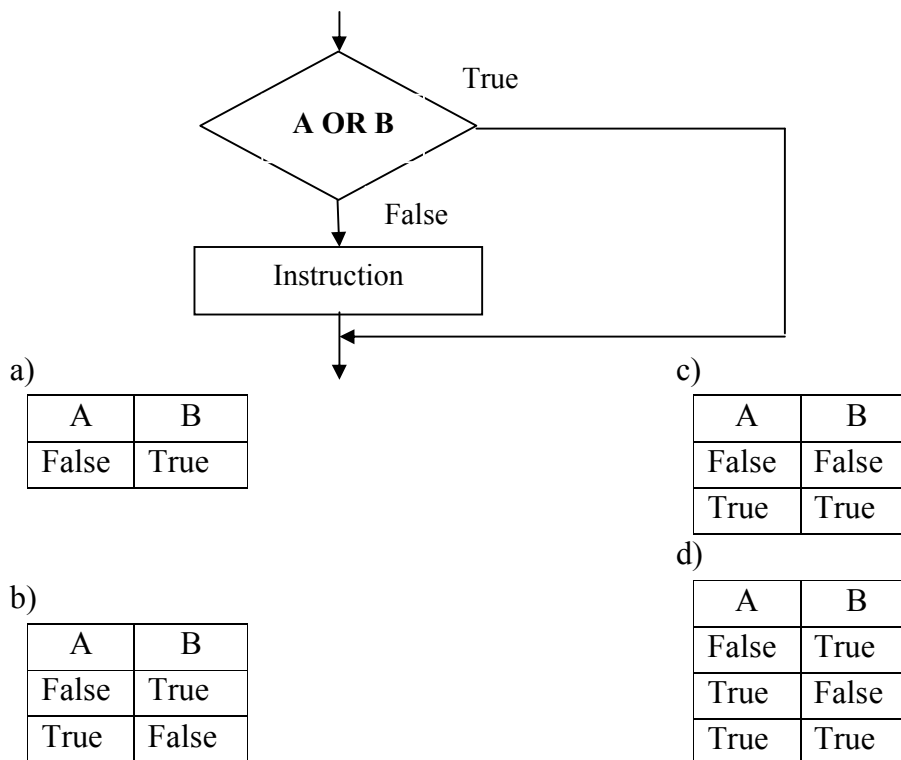
Viết chương trình nhập vào 4 số nguyên. Tìm và in ra số lớn nhất.

Hướng dẫn: Ta có 4 số nguyên a, b, c, d. Tìm 2 số nguyên lớn nhất x, y của 2 cặp (a, b) và (c,d) sau đó so sánh 2 số nguyên x, y để tìm ra số nguyên lớn nhất.

Ghi chú: Có thể so sánh tuần tự a với các số bên phải của dãy a, b, c, d để tìm số lớn nhất.

English

Q1. Which of the followig is suitable as a **test case** for testing the illustrated logic through decision condition coverage (branch coverage)?



Q2. Which of the following is logically equivalent to the expression below? Here A, B, C, and Dare integers, '<' is a less-than operator, and '≤' is a less-than-or-equal-tooperator.

not ((A ≤ B) or (C < D))

- a) (A < B) and (C ≤ D) b) (A < B) or (C ≤ D)
 c) (B < A) and (D ≤ C) d) (B < A) or (D ≤ C)