Kỹ Thuật Lập Trình

Lab 4 – Cấu trúc điều khiển lặp

1 MỤC TIÊU CỦA BÀI THỰC HÀNH

- Hiểu và vận dụng các cấu trúc điều khiển lặp để giải quyết trọn vẹn 1 bài toán hay một phần các bài toán có cần đến dạng cấu trúc này
- Luyện kỹ năng giải quyết vấn đề thông qua quá trình phân rã bài toán thành các công việc con được lặp đi lặp lại.

2 BÀI TẬP BẮT BUỘC

Câu 1:

Viết chương trình in ra bảng mã ASCII của các ký tự chữ và số.

Hướng dẫn:

a) **Dữ liệu:** Chương trình phải có biến ô nhớ để chứa ký tự cần in ra giá trị trong bảng mã ASCII:
   - **ch**: kiểu char

b) **Giải thuật:**

Sử dụng vòng lặp để lặp đầy đủ các trường hợp ký tự sau:
   - Ký tự số : '0' – '9'
   - Ký tự chữ cái thường: 'a' – 'z'
   - Ký tự chữ cái in hoa: 'A' – 'Z'

Tìm hiểu cách ép kiểu để xuất giá trị trong bảng mã ASCII của các ký tự

Câu 2:

Viết chương trình cho phép:

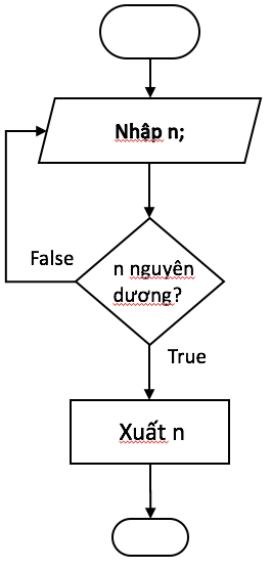
a) Nhập số N (nguyên dương), nhập cho đến khi đúng thì thôi.

b) In ra số N

Hướng dẫn:

a) Dữ liệu: Số N nguyên dương: (có thể) sử dụng kiểu unsigned long

b) Giải thuật:



- Vì việc nhập n thực hiện trước quá trình lặp → gợi ý sử dụng cấu trúc do…while
- Việc kiểm tra n nguyên dương là kiểu kiện thoát khỏi vòng lặp.

Câu 3:

Viết chương trình cho phép:

a) Nhập số N (nguyên dương), nhập cho đến khi đúng thì thôi.
b) Tính tổng bình phương của các số lẻ từ 1 đến N bằng cách lặp

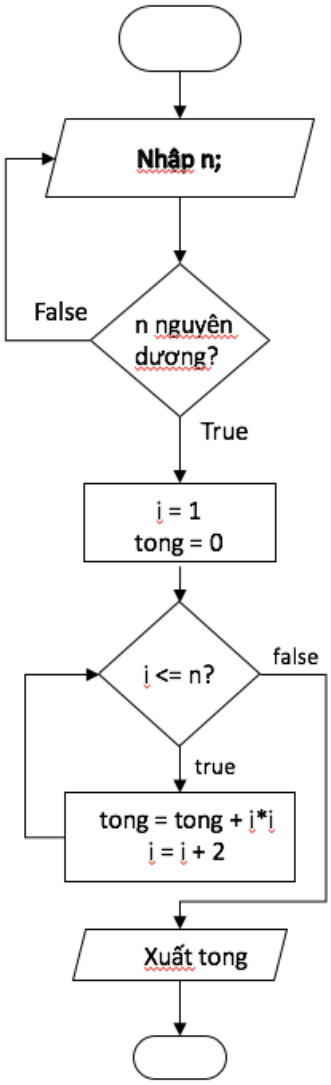
In ra N và kết quả - chọn cách in cho thẩm mỹ

Hướng dẫn:

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

- Biến lưu số n nguyên dương: (có thể) sử dụng unsigned long
- Biến lưu tổng bình phương: (có thể) sử dụng unsigned long long

b) Giải thuật:



- Vì số lần lặp đã cho trước → sử dụng cấu trúc lặp for cho việc tính tổng bình phương.

Câu 4:

Viết chương trình cho phép:

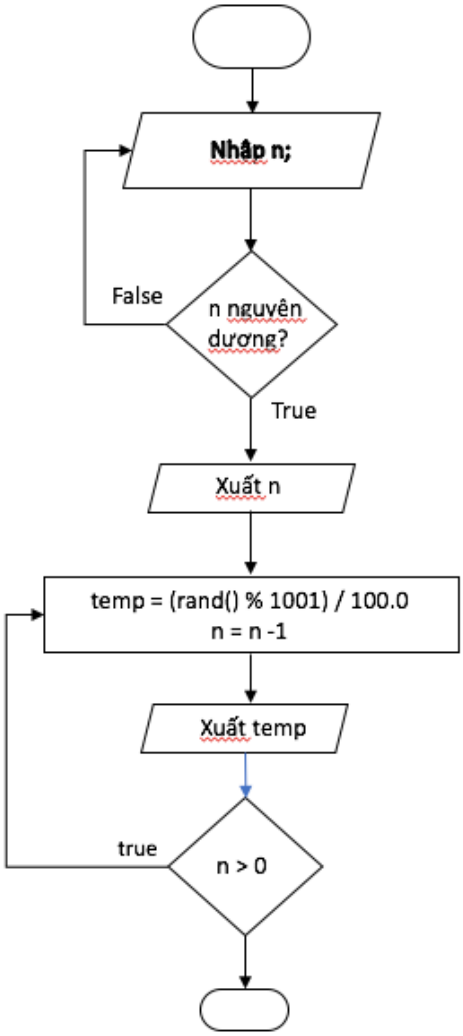
a) Nhập số N (nguyên dương), nhập cho đến khi đúng thì thôi.
b) Sinh ra N con số ngẫu nhiên các con số thực biểu diễn điểm hệ 10
c) In ra N và kết quả - chọn cách in cho thẩm mỹ

Hướng dẫn:

a) Dữ liệu:

- Biến n nguyên dương: có thể xử dụng unsigned long
- Biến lưu tạm điểm hệ 10: có thể sử dụng float

b) Giải thuật:



- Tham khảo them hàm rand() để lấy giá trị số nguyên dương ngẫu nhiên từ 0 đến RAND_MAX

Câu 5:

Viết chương trình cho phép:

a) Nhập số N (nguyên dương), nhập cho đến khi đúng thì thôi.
b) Sinh ra N con số ngẫu nhiên các con số thực biểu diễn điểm hệ 10
c) Tính trung bình của các điểm sinh ra ở trên
d) In ra N và kết quả - chọn cách in cho thẩm mỹ

Hướng dẫn:

a) Dữ liệu: chương trình nên có biến (ô nhớ) chứa các kiểu dữ liệu sau đây:
- Biến chứa số nguyên dương bất kì cho người dùng nhập vào: kiểu dữ liệu

long long
- Biến chứa số thực được sinh ra ngẫu nhiên từ chương trình: kiểu dữ liệu float
- Biến chứa giá trị trung bình của các số thực ngẫu nhiên: kiểu dữ liệu double

b) Giải thuật:
- Dùng vòng lặp while để kiểm tra số nhập vào có đúng là số nguyên dương không. Nếu không đúng, yêu cầu nhập lại.
- Dùng hàm thư viện srand kết hợp với time để khởi tạo hàm tạo số ngẫu nhiên.
- Dùng hàm thư viện rand để tạo số thực ngẫu nhiên kết hợp với vòng lặp for để tính giá trị trung bình của các số thực đã sinh ra.

Câu 6:

Viết chương trình cho phép:

a) Nhập số N (nguyên dương), nhập cho đến khi đúng thì thôi.
b) Tính N!
c) In ra N và kết quả - chọn cách in cho thẩm mỹ

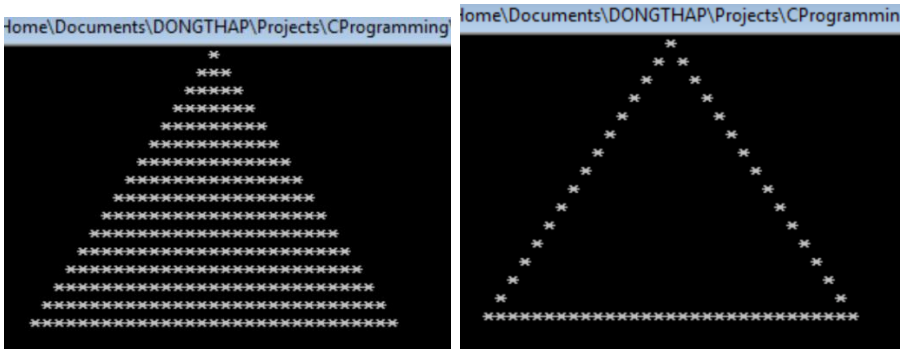
Hướng dẫn:

a) Dữ liệu: chương trình nên có biến (ô nhớ) chứa các kiểu dữ liệu sau đây:
- Biến chứa số nguyên dương bất kì cho người dùng nhập vào: kiểu dữ liệu long long
- Biến chứa giá trị trả về khi tính n!: kiểu dữ liệu long long

b) Giải thuật:
- Dùng vòng lặp while để kiểm tra số nhập vào có đúng là số nguyên dương không. Nếu không đúng, yêu cầu nhập lại.
- Dùng vòng lặp for để tính giá trị n!.
- Xuất kết quả ra màn hình

Câu 7:

Viết chương trình in ra tam giác cân rỗng và đặc



Hướng dẫn:

a) **Dữ liệu:** Ta cần một biến N để biểu diễn độ dài một cạnh của tam giác cần in và chiều cao H của tam giác.

b) **Giải thuật:**

Nhận thấy số dòng phải in chính là chiều cao của tam giác, dựa vào N chúng ta có thể tính được độ dài đường cao H. Điều này cho ta biết rằng ta có thể sử dụng vòng lặp để in ra từng dòng cho tam giác. Vì chúng ta đã biết được số lần lặp cố định dựa vào kích thước cạnh tam giác, vòng lặp for sẽ thích hợp trong trường hợp này.

Quan sát tam giác cân đặc, dòng đầu tiên, kí tự chính giữa sẽ là dấu '*' cách một khoảng so với lề là N/2-1 khoảng trắng, các kí tự '*' dòng 2 sẽ cách lề N/2-2 khoảng trắng, tương tự cho các dòng tiếp theo. Và độ dài một dòng được biểu diễn bằng số dấu '*' sẽ tăng đều từ 1, 3, 5, đến N. Sinh viên có thể sử dụng hàm setw kết hợp với 2 vòng lặp lồng nhau để in ra tam giác đặc.

Với trường hợp tam giác rỗng. Mỗi hàng trừ hàng đầu tiên và hàng cuối cùng thì chỉ có 2 ký tự '*'. Hàng đầu tiên chỉ có một ký tự '*' và hàng cuối cùng sẽ có N ký tự '*'. Trong trường hợp này ta chỉ cần sử dụng một vòng lặp để in ra phần thân tam giác, mỗi hàng in 2 kí tự '*' sao cho đúng vị trí, hàng trên cùng và hàng dưới cùng sẽ xử lý riêng. Sử dụng quan sát đối với tam giác đặc phía trên và hàm setw để hoàn chỉnh bài in tam giác rỗng.

Câu 8:

Viết chương trình tính số hạng thứ n của dãy Fibonaci

F(0) = 1

F(1) = 1

F(n) = F(n-1) + F(n-2), n>=2

Hướng dẫn:

a) **Dữ liệu:** Chương trình phải có các biến (ô nhớ) để chứa các dữ liệu:
   - **n**: kiểu float (để kiểm tra yêu cầu nhập vào chính xác n phải là một số nguyên không âm)
   - **previous, current, next**: kiển int

b) **Giải thuật:**
   - Kiểm tra điều kiện nhập vào của giá trị n nhập vào. Yêu cầu phải kiểm tra được n là số nguyên không âm. Mọi giá trị nhập khác (chẳng hạn ký tự, số âm, ký tự,…) đều phải báo lỗi
   - Khởi tạo các biến previous, current, next với giá trị khởi tạo thích hợp
   - Lặp từ 0 đến n và cập nhật giá trị các biến nhớ previous, current, next thích hợp
   - Xuất kết quả giá trị của số Fibonacci thứ n. Lưu ý đảm bảo kiểm tra F(0) = F(1) = 1

Câu 9:

Viết chương trình tính sin(x)

Hướng dẫn:

Hàm sin(x) có thể được xấp xỉ theo Taylor:

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$$

a) **Dữ liệu:** x là số thực, vì vậy ta có thể chọn kiểu dữ liệu cho nó là: float hoặc double.

b) **Giải thuật:**

Xác định giá trị Epsilon (một số rất nhỏ), ví dụ Epsilon = 10^{-10}.

Nhận thấy phép tổng bên vế phải của công thức có thể phân tách thành n toán hạng / phần tử.

Để tính sin(x) ta cần cộng tích lũy các phần tử bằng cách dùng vòng lặp, mỗi lần lặp ta sẽ tính giá trị phần tử và cộng tiếp vào biến kết quả. Vòng lặp sẽ dừng khi giá trị của phần tử đủ nhỏ (so sánh với epsilon). Ta nhận thấy rằng giá trị của epsilon càng nhỏ thì độ chính xác tính toán xấp xỉ của ta càng cao.

Mỗi phần tử cần tính toán các thành phần sau:

- A = x^{2n+1}
- B = (2n + 1)!
- C = $(-1)^n$

Phần tử = C * A / B

Tổng tất cả các phần tử ở mỗi lần lặp chính là sin(x) cần tính.

Lưu ý: góc x trong công thức đơn vị là radian, vì vậy sinh viên cần đổi x từ radian sang đơn vị độ trước khi tiến hành các vòng lặp tính toán.

Câu 10:

Viết chương trình tính các hàm khác trong trong slide bài giảng: cos(x), tan(x), log(x)

Hướng dẫn:

Hàm cos(x) và ln(x) có thể được xấp xỉ theo Taylor:

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} \quad = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots$$

$$\ln x = 2\left[\left(\frac{x-1}{x+1}\right) + \frac{1}{3}\left(\frac{x-1}{x+1}\right)^3 + \frac{1}{5}\left(\frac{x-1}{x+1}\right)^5 + \cdots\right] \qquad x > 0$$

a) **Dữ liệu:** x là số thực, vì vậy ta có thể chọn kiểu dữ liệu cho nó là: float hoặc double.

b) **Giải thuật:**
Sinh viên tiến hành phân tích tương tự như câu 10 để tính toán các các toán hạng / phần tử ở mỗi lần lặp.

Việc nhận thấy quy luật biến đổi của mỗi phần tử trong các lần lặp là rất quan trọng.

Tan(x) ta có thể tính bằng cách lấy sin(x) / cos(x).

Lưu ý: góc x trong công thức cos(x) đơn vị là radian, vì vậy sinh viên cần đổi x từ radian sang đơn vị độ trước khi tiến hành các vòng lặp tính toán.

Câu 11:

Viết chương trình tính giá trị đa thức bậc N

Hướng dẫn:

Dạng đa thức tính toán có thể là: $A_0 + A_1X + A_2X^2 + A_3X^3 + A_4X^4 + \dots$

a) **Dữ liệu:** Người dùng cần nhập:
   - Bậc N của đa thức: có thể chọn kiểu dữ liệu là số nguyên dương.
   - Giá trị của biến X: có thể chọn kiểu dữ liệu là số thực.
   - N lần hệ số của mỗi đơn thức: có thể chọn kiểu dữ liệu là số thực.

a) **Giải thuật:**

Xét đa thức sau: $a_nx^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0$

Chương trình yêu cầu người dùng nhập các hệ số dưới dạng chuỗi "$a_n\ a_{n-1}\ a_{n-2}\ \dots\ a_1\ a_0$". Sinh viên dùng stringstream để đọc các số trong đó.

Tiếp theo chương trình yêu cầu nhập x, nhập đến khi nào đúng là số thực thì thôi.

Cách tính đa thức:

- Giải thuật dùng cách biểu diễn sau cho đa thức bậc n:
  $[[a_n*x + a_{n-1}]* x + a_{n-2}]*x + \dots$
- Do đó, chúng ta tích luỹ giá trị của biểu thức bởi biểu thức sau:
  $p_x = p_x*x$ + hệ_số
- Khởi động, $p_x = 0$ => Biểu thức trên lần được đánh giá cho ra "hệ số" bậc n
  - Lần kế tiếp, sẽ là $a_n*x + a_{n-1}$
  - Lần nữa là, $[a_n*x + a_{n-1}]*x = a_n*x^2 + a_{n-1}*x$
  - … cuối cùng sẽ có biểu thức mong muốn.

Ví dụ cụ thể cho giải thuật:

Đa thức cần tính: $3x^2 + 4x + 5$

Người dùng nhập: 3 4 5

Người dùng nhập x = 2

Khởi động p_x = 0

Lần lặp 1:

- Đọc hệ số // coeff = 3
- p_x = p_x*x + coeff // p_x = 0*2 + 3 = 3

Lần lặp 2:

- Đọc hệ số // coeff = 4
- p_x = p_x*x + coeff // p_x = 3*2 + 4 = 10

Lần lặp 3:

- Đọc hệ số // coeff = 5
- p_x = p_x*x + coeff // p_x = 10*2 + 5 = 25

Ta tính được giá trị cần tìm là p_x = 25

Câu 12:

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. Nếu người dùng nhập sai thì phải xoá màn hình và cho nhập lại, **đến khi nhập đúng**.

Hướng dẫn:

Dạng đa thức tính toán có thể là: $A_0 + A_1X + A_2X^2 + A_3X^3 + A_4X^4 + \ldots$

a) **Dữ liệu:** sinh viên tham khảo câu 6 lab 3

b) **Giải thuật:**

Vì chương trình sẽ in ra các sự lựa chọn cho người dùng trước một lần và chỉ lặp lại khi người dùng nhập sai. Đối với kiểu cấu trúc lặp ít nhất một lần thì vòng lặp do-while sẽ thích hợp hơn cả. Phần in menu sẽ tương tư câu 6 lab 3. Nhưng khi người dùng nhập sai thì menu sẽ được in lại. Sinh viên có thể xem xét việc dùng thêm một biến Boolean làm điều kiện cho vòng lặp tiếp tục.

Để xóa màn hình console trên Windows, sinh viên có thể dùng lệnh system(“cls”).