

C++ EXERCISES 2

Ex 2.1: Nhập số tự nhiên n sau đó tìm và in ra:

- a) Các số chẵn nhỏ hơn n và
- b) Các số lẻ nhỏ hơn n .

Ex 2.2: Nhập số tự nhiên n sau đó tính tổng:

- a) $S = 1 + 2 + \dots + n$

Ex 2.3: Nhập số tự nhiên n rồi tính tổng sau:

- a) $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots + \frac{1}{n}$

Ex 2.4: Nhập số tự nhiên n rồi tính các tổng sau:

- a) S_n = Tổng các số tự nhiên không lớn hơn n
- b) S_1 = Tổng các số tự nhiên lẻ không lớn hơn n
- c) S_2 = Tổng các số tự nhiên chẵn không lớn hơn n .

Ex 2.5: Nhập số nguyên dương n và tính:

- a) $S = 1 + 1.2 + 1.2.3 + \dots + 1.2.3 \dots n$

Ex 2.6: Nhập hai số tự nhiên n, k rồi tính tổng các số tự nhiên không lớn hơn n và chia hết cho k .

Ex 2.7: Nhập số tự nhiên n và thực hiện:

- a) Liệt kê các ước số của nó
- b) Đếm xem số đó có bao nhiêu ước số?

Ex 2.8: Viết chương trình C++ tìm ước số chung lớn nhất và bội số chung nhỏ nhất của hai số nguyên dương nhập vào từ bàn phím.

Ex 2.9: Viết chương trình C++ nhập số nguyên tùy ý từ bàn phím và kiểm tra số vừa nhập có phải số nguyên tố hay không.

Ex 2.10. Hãy phân tích một số nguyên thành các thừa số nguyên tố

Ví dụ: Số 128 được phân tích thành $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$

Ex 2.11. Nhập vào một giá trị số nguyên dương n và liệt kê n số nguyên tố đầu tiên tìm được.

Ex 2.12. Dãy số Fibonacci được định nghĩa như sau:

$$f(n) = \begin{cases} 0 & \text{với } n = 0 \\ 1 & \text{với } n = 1 \\ f_{n-1} + f_{n-2} & \text{với } n \geq 2 \end{cases}$$

Hãy viết chương trình C++ nhập vào giá trị nguyên dương n tìm số Fibonacci thứ n .

Ex 2.13. Một số được gọi là số thuận nghịch nếu ta đọc từ trái sang phải hay từ phải sang trái số đó ta vẫn nhận được một số giống nhau. Hãy viết chương trình C++ nhập một số và kiểm tra xem số đó có phải số thuận nghịch hay không.

Ex 2.14: Nhập số tự nhiên n rồi tính $n!$ theo công thức:

$$n! = 1 \quad \text{nếu } n = 0$$

$$n! = n * (n-1) * (n-2) * \dots * 2 * 1 \quad \text{nếu } n > 0$$

Ex 2.15: Hãy viết chương trình tính tổng các chữ số của một số nguyên bất kỳ.

Ví dụ: Số 123456789 có tổng các chữ số là: $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45$

Ex 2.16: Nhập một số thực $c > 0$ (ví dụ $c = 0.0001$) rồi tính số π theo công thức:

$$\pi = 4 * \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + (-1)^n \frac{1}{2n+1} \right)$$

Tổng được tính với n đủ lớn đến khi bất đẳng thức $\frac{1}{2n+1} \geq c$ còn thỏa mãn.

So sánh với giá trị π có sẵn trong thư viện C++.

Ex 2.17: Nhập một số $c > 0$ (ví dụ $c = 0.0001$) và một số thực x rồi tính

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$

Tổng được tính với n đủ lớn sao cho bất đẳng thức $\left| \frac{x^n}{n!} \right| \leq c$ thỏa mãn.

So sánh với giá trị e^x của bạn với giá trị e^x trong thư viện C++.

Ex 2.18: Nhập một số $c > 0$ (ví dụ $c = 0.0001$) và một số thực x rồi tính

$$\sin x = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{(2n+1)}}{(2n+1)!}$$

Tổng được tính với n đủ lớn sao cho bất đẳng thức $\left| \frac{x^{(2n+1)}}{(2n+1)!} \right| \geq c$ còn thỏa mãn.

So sánh kết quả trên đây với giá trị hàm $\sin(x)$ có sẵn trong thư viện C++.

C++ Tutorial