

1. **BT 1:** Viết chương trình nhập vào ngày, tháng, năm. Kiểm tra ngày và tháng nhập có hợp lệ hay không. Tính thứ trong tuần của ngày đó.

Năm nhuận (leap year) tính theo lịch Gregorian (từ 1582): năm phải chia hết cho 4 và không chia hết cho 100, hoặc năm phải chia hết cho 400.

**Thứ trong tuần tính theo công thức Zeller3 :**

$\text{dayofweek} = (d + y + y/4 - y/100 + y/400 + (31 * m)/12) \% 7$  với:

$d = (14 - \text{month})/12$

$y = \text{year} - d$

$m = \text{month} + 12*d - 2$

dayofweek: 0 (chúa nhật), 1 (thứ hai), 2 (thứ ba),...

**Test :**

Nhap ngay, thang va nam: 20 4 1976 ↵

**Out:** Hop le

Thu 3

2. **BT 2:** Viết chương trình tìm các số hoàn hảo (perfect number) nhỏ hơn một số nguyên dương n cho trước. Biết số hoàn hảo là số nguyên dương, bằng tổng các ước số thực sự của nó (ví dụ:  $28 = 14 + 7 + 4 + 2 + 1$ ).

**Test :** Nhap n: 10000 ↵

Cac so hoan hao nho hon 10000: 6 28 496 812

3. **BT 3:** Nhập vào một số tự nhiên n (n khai báo kiểu unsigned long)

a. Số tự nhiên n có bao nhiêu chữ số.

b. Hãy tìm chữ số cuối cùng của n.

c. Hãy tìm chữ số đầu tiên của n.

d. Tính tổng các chữ số của n.

e. Hãy tìm số đảo ngược của n.

4. **BT 4:** Nhập vào hai số nguyên dương a, b. Tính ước số chung lớn nhất và bội số chung nhỏ nhất của a, b

**Test:** Nhap cap (a, b): 12 8 ↵

USCLN (a, b): 4

BSCNN (a, b): 24

5. **BT 5:** Nhập vào tử số, mẫu số (đều khác 0) của một phân số. Hãy rút gọn phân số này. Chọn dạng xuất thích hợp trong trường hợp mẫu số bằng 1 và phân số có dấu

**Test:** Nhap tu so, mau so: -3 -15 ↵

Rut gon: 1/5

Nhap tu so, mau so: 8 -2 ↵

Rut gon: -4

6. **BT 6:** Nhập vào một số nguyên dương  $n$ , phân tích  $n$  thành các thừa số nguyên tố.

**Tets :** Nhập  $n$ : 12345 ↵  
3 \* 5 \* 823

7. **BT 7:** Viết chương trình mô phỏng hàm ROUND của Microsoft Excel, dùng làm tròn một số double với một số  $n$  cho trước.
- Nếu  $n > 0$ , số làm tròn sẽ có  $n$  chữ số phân thập phân.
  - Nếu  $n = 0$ , số làm tròn sẽ là số nguyên gần nhất.
  - Nếu  $n < 0$ , số làm tròn là số nguyên làm tròn từ vị trí thứ  $n$  tính từ phải sang.

**Test :** Nhập số thực  $x$ : 3.1415926535 ↵

Do chính xác: 7 ↵  
3.1415927

**Nhập số thực  $x$ :** -4.932 ↵

Do chính xác: 0 ↵  
-5

**Nhập số thực  $x$ :** 21.5 ↵

Do chính xác: -1 ↵  
20

8. **BT 8:** Cho  $n_i$  là một số nguyên dương, với định nghĩa:

$$n_{i+1} = \begin{cases} n_i / 2 & n_i = 2k + 1 \\ 3n_i + 1 & n_i = 2k \end{cases}$$

Chuỗi trên sẽ ngừng khi  $n_i$  có trị 1. Các số được sinh ra gọi là hailstones (mưa đá) và quá trình trên đã được chứng minh là luôn luôn dừng. Viết chương trình sinh ra chuỗi hailstones với  $n_i$  nhập vào từ bàn phím.

**Test :** Nhập  $n$ : 15 ↵

15 46 23 70 35 106  
53 160 80 40 20 10  
5 16 8 4 2 1

**Hailstones sinh được:** 18

9. **BT 9:** Số tự nhiên có  $n$  chữ số là một số Armstrong (còn gọi là narcissistic numbers hoặc pluperfect digital invariants - PPDI) nếu tổng các lũy thừa bậc  $n$  của các chữ số của nó bằng chính nó. Hãy tìm tất cả các số Armstrong có 3, 4 chữ số.

**Ví dụ:** 153 là số Armstrong có 3 chữ số vì:  $1^3 + 5^3 + 3^3 = 153$

**Test :** Số Armstrong có 3, 4 chữ số: 153 370 371 407 1634 8208 9474

**10.BT 10:** Viết chương trình kiểm tra một số nguyên dương  $n$  có là số nguyên tố hay không. Nếu không thì phải xác định số nguyên tố gần  $n$  nhất và bé hơn  $n$ .

**Test:** Nhập  $n$ : 822 ↵

822 không là số nguyên tố

Số nguyên tố bé hơn gần nhất: 821

**11.BT 11:** Viết chương trình nhập số nguyên dương  $n$ . Tìm số nguyên dương  $m$  lớn nhất sao cho:  $1 + 2 + \dots + m < n$ .

**Test:** Nhập  $n$ : 22 ↵

$1 + 2 + 3 + 4 + 5 + 6 = 21 < 22$

$m = 6$

**12.BT 12:** Nhập vào một số tiền  $n$  (nghìn đồng,  $n > 5$ ) nguyên dương. Đổi số tiền này ra ba loại tiền giấy 1000VNĐ, 2000VNĐ, 5000VNĐ. Tìm phương án đổi tiền sao cho loại tiền 2000VNĐ chiếm hơn phân nửa số tờ bạc phải đổi ít nhất.

**Test:** Nhập  $n$  (nghìn đồng,  $n > 5$ ): 137 ↵

(0, 21, 19): 40

**13.BT 13:** Tìm các bộ (trâu đứng, trâu nằm, trâu già) thỏa mãn bài toán cổ:

Trăm trâu ăn trăm bó cỏ

Trâu đứng ăn năm

Trâu nằm ăn ba

Lũy khứ trâu già

Ba con một bó

Thử tìm cách giảm số vòng lặp khi tính toán xuống.

**Test :** (4, 18, 78)

(8, 11, 81)

(12, 4, 84)

**14.BT 14:** Viết chương trình tìm cách thay thế các dấu hỏi (?) bởi các dấu 4 phép tính số học +, -, \*, /, trong biểu thức dưới đây sao cho biểu thức có giá trị bằng 36.

$((((1 ? 2) ? 3) ? 4) ? 5) ? 6$

**Test :**  $((((1 - 2) + 3) + 4) * 5) + 6 = 36$

$((((1 - 2) * 3) + 4) + 5) * 6 = 36$

$((((1 * 2) + 3) - 4) + 5) * 6 = 36$

$((((1 / 2) * 3) * 4) * 5) + 6 = 36$