

1. Số chẵn (C04_01.CPP)

Viết chương trình nhập vào một số nguyên dương x. In ra chữ “YES” nếu x là số chẵn, ngược lại in số chữ “NO”.

Input: Một số nguyên dương x

Output: In ra YES/NO

Input	Output
123	NO
654	YES

2. Max của 3 số (C04_02.CPP)

Viết chương trình nhập vào ba số nguyên a, b, c. In ra số lớn nhất của 3 số đó.

Input: Ba số nguyên dương a, b, c

Output: Số lớn nhất

Input	Output
123 433 123	433
-12 -343 -121	-12

3. Min của 4 số (C04_03.CPP)

Viết chương trình nhập vào bốn số nguyên a, b, c, d. In ra số nhỏ nhất của 4 số đó.

Input: Ba số nguyên dương a, b, c, d

Output: Số nhỏ nhất

Input	Output
123 433 123 12	12
-12 -343 -121 1	-343

4. Tam giác vuông (C04_04.CPP)

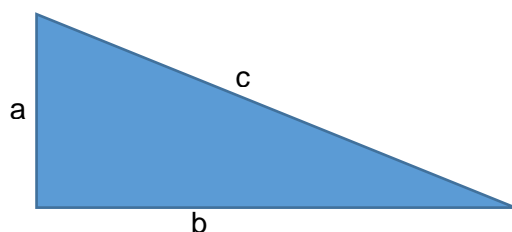
Viết chương trình nhập vào ba số nguyên dương a, b, c. In chữ “YES” nếu 3 số là chiều dài của 3 cạnh tam giác vuông, ngược lại in chữ “NO”.

Input: Ba số nguyên dương a, b, c

Output: In ra DEU/CAN/VUONG/THUONG

Input	Output
3 4 5	YES
4 5 4	NO

Hướng dẫn: Tam giác vuông khi bình phương cạnh huyền bằng tổng bình phương 2 cạnh góc vuông. $c^2 = a^2 + b^2$



5. Số chẵn (C04_05.CPP)

Viết chương trình nhập vào ba số nguyên dương a, b, c.

In chữ “DEU” nếu 3 số là chiều dài của 3 cạnh tam giác đều

In chữ “CAN” nếu 3 số là chiều dài của 3 cạnh tam giác cân.

In chữ “VUONG” nếu 3 số là chiều dài của 3 cạnh tam giác vuông

Ngược lại thì in chữ “THUONG”

Input: Ba số nguyên dương a, b, c
Output: In ra DEU/CAN/VUONG/THUONG

Input	Output
3 4 5	VUONG
4 5 4	CAN
6 6 6	DEU
7 9 8	THUONG

Hướng dẫn:

- Tam giác đều khi chiều dài 3 cạnh đều bằng nhau. $a = b = c$
- Tam giác cân khi chiều dài 2 cạnh đều bằng nhau. $a = b$ hoặc $b = c$ hoặc $a = c$

6. Tổng dãy số tự nhiên liên tiếp (C04_06.CPP)

Viết chương trình nhập vào một số nguyên x . Hãy xác định xem x có phải tổng của các số tự nhiên liên tiếp bắt đầu từ 1 hay không?

Input: Một số nguyên dương x
Output: In ra YES/NO

Input	Output
10	YES
14	NO

Giải thích: $10=1+2+3+4$ nên in YES, còn 14 không thể tạo được tổng các số tự nhiên liên tiếp.

7. Tìm số trên vòng tròn (C04_07.CPP)

Các chữ số từ 1 đến x được xếp thứ tự trên 1 vòng tròn. Một trò chơi như sau: Bạn xuất phát tại vị trí x ($1 \leq x \leq n$) và đi theo chiều kim đồng hồ k bước (mỗi bước qua một số).

Hãy xác định xem sau khi đi k bước thì bạn sẽ đứng tại vị trí số mấy?

Input: Ba số nguyên dương n, x và k
Output: Vị trí sau khi đi k bước

Input	Output
10 3 4	7
10 9 3	2

8. Sắp xếp 2 số (C04_08.CPP)

Viết chương trình nhập vào hai số nguyên a, b . In ra hai số đó theo thứ tự nhỏ đến lớn.

Input: Hai số nguyên dương a và b
Output: in 2 số nhỏ trước – lớn sau

Input	Output
10 21	10 21
10 9	9 10

9. Sắp xếp 3 số (C04_09.CPP)

Viết chương trình nhập vào hai số nguyên a, b, c . In ra ba số đó theo thứ tự giảm dần (lớn lùi lại nhỏ).

Input: Hai số nguyên dương a, b, c
Output: in 3 số giảm dần

Input	Output
12 32 13	32 13 12
122 111 222	222 122 111

10. Vị trí số lớn nhất (C04_10.CPP)

Viết chương trình nhập vào một số nguyên x có 3 chữ số. In ra số lớn nhất và nằm ở hàng nào?.

Input: Một số nguyên dương x
Output: Chữ số lớn nhất và vị trí

Input	Output
123	3 DONVI
765	7 TRAM
453	5 CHUC

11. Số chính phương (C04_11.CPP)

Viết chương trình nhập vào một số nguyên n . Kiểm tra xem n có phải là số chính phương hay không? (Số chính phương là bình phương của một số nguyên như $16 = 4^2$)?

Input: Một số nguyên dương n
Output: Là số chính phương thì in YES
Ngược lại in NO

Input	Output
4	YES
100	YES
123	NO

Hướng dẫn:

- Số chính phương là số khi lấy căn bậc 2 có kết quả là số nguyên.
- Sử dụng hàm `sqrt(n)` để tính căn bậc n (hàm `sqrt` nằm trong thư viện `<math.h>`)