

# CodeForge - B01 - In Số Từ 1 Đến N

**Độ khó:** ★ Easy

## Đề bài

In ra các số từ 1 đến N, mỗi số trên một dòng.

### ◊ Input

- Một số nguyên dương **N**

### ◊ Output

- In ra các số từ 1 đến N, mỗi số trên một dòng

### ◊ Constraints

- **1 ≤ N ≤ 1000**

## Ví dụ

Test case 1

**Input:**

```
5
```

**Output:**

```
1
2
3
4
5
```

Test case 2

**Input:**

```
3
```

**Output:**

```
1  
2  
3
```

### Test case 3

**Input:**

```
1
```

**Output:**

```
1
```

---

**Tags:** [for-loop](#), [basic-loop](#), [counting](#)

---

# CodeForge - B02 - In Số Từ N Xuống 1

---

**Độ khó:** ★ Easy

## Đề bài

In ra các số từ N xuống 1, mỗi số trên một dòng.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra các số từ N xuống 1, mỗi số trên một dòng

### ◊ Constraints

- $1 \leq N \leq 1000$

## Ví dụ

Test case 1

**Input:**

```
5
```

**Output:**

```
5
4
3
2
1
```

Test case 2

**Input:**

```
3
```

**Output:**

```
3  
2  
1
```

### Test case 3

**Input:**

```
1
```

**Output:**

```
1
```

---

**Tags:** *for-loop, reverse-counting*

---

# CodeForge - B03 - Tổng Các Số Từ 1 Đến N

---

**Độ khó:** ★ Easy

## Đề bài

Tính tổng các số từ 1 đến N.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra tổng  $1 + 2 + 3 + \dots + N$

### ◊ Constraints

- $1 \leq N \leq 10^6$

## Ví dụ

Test case 1

**Input:**

```
5
```

**Output:**

```
15
```

**Giải thích:**  $1 + 2 + 3 + 4 + 5 = 15$

Test case 2

**Input:**

```
10
```

**Output:**

55

### Test case 3

**Input:**

100

**Output:**

5050

---

**Tags:** [loop](#), [sum](#), [accumulation](#)

---

# CodeForge - B04 - Tích Các Số Từ 1 Đến N

---

**Độ khó:** ★ Easy

## Đề bài

Tính tích các số từ 1 đến N (giai thừa  $N!$ ).

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra  $N!$

### ◊ Constraints

- $0 \leq N \leq 20$

## Ví dụ

Test case 1

**Input:**

```
5
```

**Output:**

```
120
```

Test case 2

**Input:**

```
0
```

**Output:**

```
1
```

### Test case 3

**Input:**

```
10
```

**Output:**

```
3628800
```

---

**Tags:** factorial, loop, multiplication

---

# CodeForge - B05 - In Các Số Chẵn

---

**Độ khó:** ★ Easy

## Đề bài

In ra các số chẵn từ 1 đến N, mỗi số cách nhau bởi dấu cách.

### ◊ Input

- Một số nguyên dương **N**

### ◊ Output

- In ra các số chẵn từ 1 đến N trên một dòng, cách nhau bởi dấu cách

### ◊ Constraints

- 1 ≤ N ≤ 1000**

## Ví dụ

Test case 1

**Input:**

```
10
```

**Output:**

```
2 4 6 8 10
```

Test case 2

**Input:**

```
7
```

**Output:**

```
2 4 6
```

## Test case 3

**Input:**

```
1
```

**Output:**

**Giải thích:** Không có số chẵn nào

---

**Tags:** `loop`, `even-numbers`, `filter`

# CodeForge - B06 - In Các Số Lẻ

**Độ khó:** ★ Easy

## Đề bài

In ra các số lẻ từ 1 đến N, mỗi số cách nhau bởi dấu cách.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra các số lẻ từ 1 đến N trên một dòng, cách nhau bởi dấu cách

### ◊ Constraints

- $1 \leq N \leq 1000$

## Ví dụ

Test case 1

**Input:**

```
10
```

**Output:**

```
1 3 5 7 9
```

Test case 2

**Input:**

```
8
```

**Output:**

```
1 3 5 7
```

### Test case 3

**Input:**

```
5
```

**Output:**

```
1 3 5
```

---

**Tags:** `loop`, `odd-numbers`, `filter`

# CodeForge - B07 - Bảng Cửu Chương

**Độ khó:** ★ Easy

## Đề bài

In bảng cửu chương của số N từ 1 đến 10.

Format:  $N \times i = \text{kết quả}$

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra 10 dòng, mỗi dòng theo format trên

### ◊ Constraints

- $1 \leq N \leq 100$

## Ví dụ

Test case 1

**Input:**

```
5
```

**Output:**

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Test case 2

**Input:**

```
3
```

**Output:**

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

---

**Tags:** multiplication-table, loop, formatting

---

# CodeForge - B08 - Đếm Số Chữ Số

---

**Độ khó:** ★ Easy

## Đề bài

Đếm số lượng chữ số của một số nguyên không âm.

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra số lượng chữ số

### ◊ Constraints

- $0 \leq N \leq 10^{18}$

## Ví dụ

Test case 1

**Input:**

```
12345
```

**Output:**

```
5
```

Test case 2

**Input:**

```
0
```

**Output:**

```
1
```

### Test case 3

**Input:**

```
10000000000
```

**Output:**

```
10
```

---

**Tags:** digit-count, while-loop, number-manipulation

---

# CodeForge - B09 - Tổng Các Chữ Số

---

**Độ khó:** ★ Easy

## Đề bài

Tính tổng các chữ số của một số nguyên không âm.

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra tổng các chữ số

### ◊ Constraints

- $0 \leq N \leq 10^{18}$

## Ví dụ

Test case 1

**Input:**

```
12345
```

**Output:**

```
15
```

Test case 2

**Input:**

```
999
```

**Output:**

```
27
```

### Test case 3

**Input:**

```
0
```

**Output:**

```
0
```

---

**Tags:** `digit-sum`, `loop`, `modulo`

---

# CodeForge - B10 - Đảo Ngược Số

---

**Độ khó:** ★ ★ Medium

## Đề bài

Đảo ngược các chữ số của một số nguyên không âm.

Bỏ các số 0 đứng đầu sau khi đảo ngược.

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra số sau khi đảo ngược

### ◊ Constraints

- $0 \leq N \leq 10^{18}$

## Ví dụ

Test case 1

**Input:**

```
12345
```

**Output:**

```
54321
```

Test case 2

**Input:**

```
1000
```

**Output:**

```
1
```

### Test case 3

**Input:**

```
0
```

**Output:**

```
0
```

---

**Tags:** reverse-number, loop, digit-manipulation

---

# CodeForge - B11 - Kiểm Tra Số Palindrome

---

**Độ khó:** ★ ★ Medium

## Đề bài

Kiểm tra một số có phải là số đối xứng (palindrome) hay không.

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra **YES** nếu là palindrome
- In ra **NO** nếu không phải

### ◊ Constraints

- $0 \leq N \leq 10^{18}$

## Ví dụ

Test case 1

**Input:**

```
12321
```

**Output:**

```
YES
```

Test case 2

**Input:**

```
12345
```

**Output:**

```
NO
```

### Test case 3

**Input:**

```
0
```

**Output:**

```
YES
```

### Test case 4

**Input:**

```
1001
```

**Output:**

```
YES
```

---

**Tags:** palindrome, number-theory, loop

---

# CodeForge - B12 - Kiểm Tra Số Nguyên Tố

---

**Độ khó:** ★ ★ Medium

## Đề bài

Kiểm tra một số có phải là số nguyên tố hay không.

Số nguyên tố là số lớn hơn 1 và chỉ chia hết cho 1 và chính nó.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra **YES** nếu là số nguyên tố
- In ra **NO** nếu không phải

### ◊ Constraints

- $1 \leq N \leq 10^9$

## Ví dụ

Test case 1

**Input:**

```
7
```

**Output:**

```
YES
```

Test case 2

**Input:**

```
1
```

**Output:**

NO

### Test case 3

**Input:**

100

**Output:**

NO

### Test case 4

**Input:**

997

**Output:**

YES

---

**Tags:** prime-number, loop, optimization

---

# CodeForge - B13 - In Số Nguyên Tố Đến N

---

**Độ khó:** ★★ Medium

## Đề bài

In ra tất cả các số nguyên tố từ 2 đến N, mỗi số cách nhau bởi dấu cách.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra các số nguyên tố từ 2 đến N trên một dòng, cách nhau bởi dấu cách

### ◊ Constraints

- $2 \leq N \leq 10000$

## Ví dụ

Test case 1

**Input:**

```
10
```

**Output:**

```
2 3 5 7
```

Test case 2

**Input:**

```
20
```

**Output:**

```
2 3 5 7 11 13 17 19
```

### Test case 3

**Input:**

```
2
```

**Output:**

```
2
```

---

**Tags:** prime-numbers, nested-loop, sieve

---

# CodeForge - B14 - Số Fibonacci Thứ N

---

**Độ khó:** ★ ★ Medium

## Đề bài

Tìm số Fibonacci thứ N.

Dãy Fibonacci:  $F(0) = 0$ ,  $F(1) = 1$ ,  $F(n) = F(n-1) + F(n-2)$

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra số Fibonacci thứ N

### ◊ Constraints

- $0 \leq N \leq 90$

## Ví dụ

Test case 1

**Input:**

```
0
```

**Output:**

```
0
```

Test case 2

**Input:**

```
10
```

**Output:**

55

### Test case 3

**Input:**

20

**Output:**

6765

### Test case 4

**Input:**

50

**Output:**

12586269025

---

**Tags:** fibonacci, loop, sequence

# CodeForge - B15 - In Dãy Fibonacci

**Độ khó:** ★ ★ Medium

## Đề bài

In ra N số đầu tiên của dãy Fibonacci, mỗi số cách nhau bởi dấu cách.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra N số Fibonacci đầu tiên trên một dòng, cách nhau bởi dấu cách

### ◊ Constraints

- $1 \leq N \leq 50$

## Ví dụ

Test case 1

**Input:**

```
5
```

**Output:**

```
0 1 1 2 3
```

Test case 2

**Input:**

```
10
```

**Output:**

```
0 1 1 2 3 5 8 13 21 34
```

### Test case 3

**Input:**

```
1
```

**Output:**

```
0
```

---

**Tags:** fibonacci, sequence, loop

---

# CodeForge - B16 - Ước Số Của N

---

**Độ khó:** ★ ★ Medium

## Đề bài

In ra tất cả các ước số của N, mỗi số cách nhau bởi dấu cách.

### ◊ Input

- Một số nguyên dương **N**

### ◊ Output

- In ra các ước số theo thứ tự tăng dần, cách nhau bởi dấu cách

### ◊ Constraints

- 1 ≤ N ≤ 10^6**

## Ví dụ

Test case 1

**Input:**

```
12
```

**Output:**

```
1 2 3 4 6 12
```

Test case 2

**Input:**

```
1
```

**Output:**

```
1
```

## Test case 3

**Input:**

```
100
```

**Output:**

```
1 2 4 5 10 20 25 50 100
```

---

**Tags:** `divisors`, `loop`, `number-theory`

---

# CodeForge - B17 - Đếm Số Ước

---

**Độ khó:** ★ ★ Medium

## Đề bài

Đếm số lượng ước số của N.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra số lượng ước số

### ◊ Constraints

- $1 \leq N \leq 10^{12}$

## Ví dụ

Test case 1

**Input:**

```
12
```

**Output:**

```
6
```

Test case 2

**Input:**

```
1
```

**Output:**

```
1
```

### Test case 3

**Input:**

```
100
```

**Output:**

```
9
```

---

**Tags:** divisor-count, optimization, loop

---

# CodeForge - B18 - Tổng Các Ước Số

---

**Độ khó:** ★ ★ Medium

## Đề bài

Tính tổng tất cả các ước số của N.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra tổng các ước số

### ◊ Constraints

- $1 \leq N \leq 10^{12}$

## Ví dụ

Test case 1

**Input:**

```
12
```

**Output:**

```
28
```

**Giải thích:**  $1 + 2 + 3 + 4 + 6 + 12 = 28$

Test case 2

**Input:**

```
1
```

**Output:**

```
1
```

### Test case 3

**Input:**

```
28
```

**Output:**

```
56
```

---

**Tags:** divisor-sum, optimization, loop

# CodeForge - B19 - GCD (Ước Chung Lớn Nhất)

**Độ khó:** ★ ★ Medium

## Đề bài

Tìm ước chung lớn nhất (GCD) của hai số sử dụng thuật toán Euclid.

### ◊ Input

- Hai số nguyên dương **a** và **b** trên hai dòng

### ◊ Output

- In ra  $\text{GCD}(a, b)$

### ◊ Constraints

- $1 \leq a, b \leq 10^{18}$

## Ví dụ

Test case 1

**Input:**

```
12
```

```
18
```

**Output:**

```
6
```

Test case 2

**Input:**

```
17
```

```
19
```

**Output:**

1

### Test case 3

**Input:**

```
100  
50
```

**Output:**

```
50
```

---

**Tags:** gcd, euclidean-algorithm, while-loop

---

# CodeForge - B20 - LCM (Bội Chung Nhỏ Nhất)

---

**Độ khó:** ★ ★ Medium

## Đề bài

Tìm bội chung nhỏ nhất (LCM) của hai số.

Công thức:  $\text{LCM}(a, b) = (a \times b) / \text{GCD}(a, b)$

### ◊ Input

- Hai số nguyên dương  $a$  và  $b$  trên hai dòng

### ◊ Output

- In ra  $\text{LCM}(a, b)$

### ◊ Constraints

- $1 \leq a, b \leq 10^9$

## Ví dụ

Test case 1

**Input:**

```
12
18
```

**Output:**

```
36
```

Test case 2

**Input:**

```
5
7
```

**Output:**

35

### Test case 3

**Input:**

```
10  
10
```

**Output:**

```
10
```

---

**Tags:** lcm, gcd, math

# CodeForge - B21 - In Hình Vuông Sao

**Độ khó:** ★★ Medium

## Đề bài

In hình vuông bằng dấu sao (\*) với kích thước  $N \times N$ .

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra hình vuông  $N \times N$  bằng dấu sao

### ◊ Constraints

- $1 \leq N \leq 50$

## Ví dụ

Test case 1

**Input:**

```
3
```

**Output:**

```
***  
***  
***
```

Test case 2

**Input:**

```
5
```

**Output:**

```
*****
*****
*****
*****
*****
```

### Test case 3

**Input:**

```
1
```

**Output:**

```
*
```

---

**Tags:** pattern, nested-loop, square

# CodeForge - B22 - In Hình Chữ Nhật Rỗng

**Độ khó:** ★ ★ Medium

## Đề bài

In hình chữ nhật rỗng bằng dấu sao (\*) với kích thước  $M \times N$ .

Chỉ in viền, bên trong rỗng.

### ◊ Input

- Dòng 1: Số dòng M
- Dòng 2: Số cột N

### ◊ Output

- In ra hình chữ nhật rỗng  $M \times N$

### ◊ Constraints

- $2 \leq M, N \leq 50$

## Ví dụ

Test case 1

**Input:**

```
4
5
```

**Output:**

```
*****
*   *
*   *
*****
```

Test case 2

**Input:**

```
3  
3
```

**Output:**

```
***  
* *  
***
```

**Test case 3****Input:**

```
2  
5
```

**Output:**

```
*****  
*****
```

---

**Tags:** pattern, hollow-rectangle, nested-loop

# CodeForge - B23 - In Tam Giác Vuông

**Độ khó:** ★ ★ Medium

## Đề bài

In tam giác vuông bằng dấu sao (\*) với chiều cao N.

```
*  
**  
***  
****
```

### ◊ Input

- Một số nguyên dương **N**

### ◊ Output

- In ra tam giác vuông với N dòng

### ◊ Constraints

- 1 ≤ N ≤ 50**

## Ví dụ

Test case 1

**Input:**

```
4
```

**Output:**

```
*  
**  
***  
****
```

Test case 2

**Input:**

```
5
```

**Output:**

```
*
```

  

```
**
```

  

```
***
```

  

```
****
```

  

```
*****
```

**Test case 3****Input:**

```
1
```

**Output:**

```
*
```

---

**Tags:** pattern, triangle, nested-loop

# CodeForge - B24 - In Tam Giác Vuông Ngược

**Độ khó:** ★ ★ Medium

## Đề bài

In tam giác vuông ngược bằng dấu sao (\*).

```
****  
***  
**  
*
```

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra tam giác vuông ngược với  $N$  dòng

### ◊ Constraints

- $1 \leq N \leq 50$

## Ví dụ

Test case 1

**Input:**

```
4
```

**Output:**

```
****  
***  
**  
*
```

Test case 2

**Input:**

```
5
```

**Output:**

```
*****
****
 ***
 **
 *
```

**Test case 3****Input:**

```
1
```

**Output:**

```
*
```

---

**Tags:** pattern, reverse-triangle, nested-loop

# CodeForge - B25 - In Tam Giác Cân

**Độ khó:** ★ ★ ★ Hard

## Đề bài

In tam giác cân bằng dấu sao (\*) với chiều cao N.

```
*  
***  
*****
```

### ◊ Input

- Một số nguyên dương **N**

### ◊ Output

- In ra tam giác cân với N dòng

### ◊ Constraints

- 1 ≤ N ≤ 50**

## Ví dụ

Test case 1

**Input:**

```
3
```

**Output:**

```
*  
***  
*****
```

Test case 2

**Input:**

5

**Output:**

```
*  
***  
*****  
*****  
*****
```

**Test case 3****Input:**

1

**Output:**

```
*
```

---

**Tags:** pattern, pyramid, nested-loop

# CodeForge - B26 - In Tam Giác Cân Ngược

**Độ khó:** ★ ★ ★ Hard

## Đề bài

In tam giác cân ngược bằng dấu sao (\*).

```
*****
```

```
***
```

```
*
```

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra tam giác cân ngược với  $N$  dòng

### ◊ Constraints

- $1 \leq N \leq 50$

## Ví dụ

Test case 1

**Input:**

```
3
```

**Output:**

```
*****
```

```
***
```

```
*
```

Test case 2

**Input:**

5

**Output:**

```
*****  
*****  
****  
***  
*
```

**Test case 3****Input:**

1

**Output:**

```
*
```

---

**Tags:** pattern, inverted-pyramid, nested-loop

# CodeForge - B27 - In Hình Kim Cương

**Độ khó:** ★ ★ ★ Hard

## Đề bài

In hình kim cương bằng dấu sao (\*) với chiều cao  $2N-1$ .

```
*  
***  
*****  
***  
*
```

### ◊ Input

- Một số nguyên dương **N** ( $N \geq 2$ )

### ◊ Output

- In ra hình kim cương

### ◊ Constraints

- $2 \leq N \leq 50$

## Ví dụ

Test case 1

**Input:**

```
3
```

**Output:**

```
*  
***  
*****  
***  
*
```

Test case 2

**Input:**

```
5
```

**Output:**

```
*
```

```
***
```

```
*****
```

```
*****
```

```
*****
```

```
*****
```

```
***
```

```
*
```

## Test case 3

**Input:**

```
2
```

**Output:**

```
*
```

```
***
```

```
*
```

---

**Tags:** pattern, diamond, nested-loop

# CodeForge - B28 - In Hình Số Tam Giác

**Độ khó:** ★ ★ ★ Hard

## Đề bài

In tam giác bằng các số.

```
1
12
123
1234
```

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra tam giác số với  $N$  dòng

### ◊ Constraints

- $1 \leq N \leq 9$

## Ví dụ

Test case 1

### Input:

```
4
```

### Output:

```
1
12
123
1234
```

Test case 2

### Input:

```
5
```

**Output:**

```
1  
12  
123  
1234  
12345
```

Test case 3

**Input:**

```
1
```

**Output:**

```
1
```

---

**Tags:** pattern, number-triangle, nested-loop

# CodeForge - B29 - In Bảng Số Floyd

**Độ khó:** ★ ★ ★ Hard

## Đề bài

In bảng số Floyd (tam giác số tăng dần liên tiếp).

```
1
2 3
4 5 6
7 8 9 10
```

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra bảng số Floyd với  $N$  dòng

### ◊ Constraints

- $1 \leq N \leq 20$

## Ví dụ

Test case 1

**Input:**

```
4
```

**Output:**

```
1
2 3
4 5 6
7 8 9 10
```

Test case 2

**Input:**

```
5
```

**Output:**

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

## Test case 3

**Input:**

```
1
```

**Output:**

```
1
```

---

**Tags:** floyd-triangle, pattern, nested-loop

---

# CodeForge - B30 - In Tam Giác Pascal (Dòng Thứ N)

---

**Độ khó:** ★ ★ ★ Hard

## Đề bài

In dòng thứ N của tam giác Pascal.

Dòng 0: 1 Dòng 1: 1 1 Dòng 2: 1 2 1 Dòng 3: 1 3 3 1 Dòng 4: 1 4 6 4 1

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra dòng thứ N của tam giác Pascal, các số cách nhau bởi dấu cách

### ◊ Constraints

- $0 \leq N \leq 30$

## Ví dụ

Test case 1

**Input:**

```
4
```

**Output:**

```
1 4 6 4 1
```

Test case 2

**Input:**

```
5
```

**Output:**

```
1 5 10 10 5 1
```

### Test case 3

**Input:**

```
0
```

**Output:**

```
1
```

---

**Tags:** pascal-triangle, combinatorics, loop

# CodeForge - B31 - Tìm Số Hoàn Hảo Đầu Tiên

**Độ khó:** ★ ★ ★ Hard

## Đề bài

Số hoàn hảo là số bằng tổng các ước thực sự của nó.

Tìm N số hoàn hảo đầu tiên.

Ví dụ:  $6 = 1 + 2 + 3$

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra N số hoàn hảo đầu tiên, mỗi số trên một dòng

### ◊ Constraints

- $1 \leq N \leq 4$
- 4 số hoàn hảo đầu: 6, 28, 496, 8128

## Ví dụ

Test case 1

**Input:**

```
2
```

**Output:**

```
6  
28
```

Test case 2

**Input:**

```
4
```

**Output:**

```
6  
28  
496  
8128
```

## Test case 3

**Input:**

```
1
```

**Output:**

```
6
```

---

**Tags:** perfect-number, divisors, loop

# CodeForge - B32 - Phân Tích Thừa Số Nguyên Tố

**Độ khó:** ★ ★ ★ Hard

## Đề bài

Phân tích một số thành tích các thừa số nguyên tố.

In ra các thừa số theo thứ tự tăng dần, mỗi thừa số xuất hiện theo số mũ của nó.

### ◊ Input

- Một số nguyên dương  $N$  ( $N \geq 2$ )

### ◊ Output

- In ra các thừa số nguyên tố, cách nhau bởi dấu cách

### ◊ Constraints

- $2 \leq N \leq 10^{12}$

## Ví dụ

Test case 1

**Input:**

```
12
```

**Output:**

```
2 2 3
```

Test case 2

**Input:**

```
100
```

**Output:**

```
2 2 5 5
```

### Test case 3

**Input:**

```
17
```

**Output:**

```
17
```

---

**Tags:** prime-factorization, loop, number-theory

---

# CodeForge - B33 - Đếm Số Chính Phương Trong Khoảng

---

**Độ khó:** ★ ★ ★ Hard

## Đề bài

Đếm số lượng số chính phương trong khoảng [A, B].

Số chính phương là số có căn bậc hai là số nguyên.

### ◊ Input

- Dòng 1: Số nguyên không âm A
- Dòng 2: Số nguyên không âm B ( $A \leq B$ )

### ◊ Output

- In ra số lượng số chính phương trong [A, B]

### ◊ Constraints

- $0 \leq A \leq B \leq 10^{18}$

## Ví dụ

Test case 1

**Input:**

```
1
25
```

**Output:**

```
5
```

**Giải thích:** 1, 4, 9, 16, 25

Test case 2

**Input:**

```
10  
50
```

**Output:**

```
4
```

**Giải thích:** 16, 25, 36, 49

Test case 3

**Input:**

```
0  
0
```

**Output:**

```
1
```

---

**Tags:** perfect-square, counting, math

---

# CodeForge - B34 - Tổng Bình Phương Các Chữ Số

---

**Độ khó:** ★ ★ ★ Hard

## Đề bài

Tính tổng bình phương các chữ số của một số.

Ví dụ:  $123 \rightarrow 1^2 + 2^2 + 3^2 = 1 + 4 + 9 = 14$

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra tổng bình phương các chữ số

### ◊ Constraints

- $0 \leq N \leq 10^{18}$

## Ví dụ

Test case 1

**Input:**

```
123
```

**Output:**

```
14
```

Test case 2

**Input:**

```
999
```

**Output:**

243

### Test case 3

**Input:**

0

**Output:**

0

---

**Tags:** digit-manipulation, sum-of-squares, loop

# CodeForge - B35A - Số Armstrong

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Số Armstrong là số bằng tổng các chữ số lũy thừa bậc bằng số chữ số.

Ví dụ:  $153 = 1^3 + 5^3 + 3^3$

Kiểm tra một số có phải là số Armstrong không.

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra YES nếu là số Armstrong
- In ra NO nếu không phải

### ◊ Constraints

- $0 \leq N \leq 10^9$

## Ví dụ

Test case 1

**Input:**

```
153
```

**Output:**

```
YES
```

Test case 2

**Input:**

```
9474
```

**Output:**

```
YES
```

## Test case 3

**Input:**

```
123
```

**Output:**

```
NO
```

## Test case 4

**Input:**

```
0
```

**Output:**

```
YES
```

---

**Tags:** `armstrong-number`, `digit-manipulation`, `power`

# CodeForge - B36A - In Tất Cả Số Armstrong Đến N

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

In ra tất cả các số Armstrong từ 1 đến N, mỗi số trên một dòng.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra các số Armstrong từ 1 đến  $N$ , mỗi số trên một dòng

### ◊ Constraints

- $1 \leq N \leq 100000$

## Ví dụ

Test case 1

**Input:**

```
200
```

**Output:**

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
153
```

Test case 2

**Input:**

```
10000
```

**Output:**

```
1
2
3
4
5
6
7
8
9
153
370
371
407
1634
8208
9474
```

---

**Tags:** [armstrong-number](#), [loop](#), [list-generation](#)

# CodeForge - B37A - Số Nguyên Tố Thứ K

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Tìm số nguyên tố thứ K trong dãy số nguyên tố.

Số nguyên tố đầu tiên là 2.

### ◊ Input

- Một số nguyên dương  $K$

### ◊ Output

- In ra số nguyên tố thứ K

### ◊ Constraints

- $1 \leq K \leq 100000$
- Số nguyên tố thứ 100,000 là 1,299,709

## Ví dụ

Test case 1

**Input:**

```
1
```

**Output:**

```
2
```

Test case 2

**Input:**

```
10
```

**Output:**

29

### Test case 3

**Input:**

100

**Output:**

541

### Test case 4

**Input:**

10000

**Output:**

104729

---

**Tags:** prime-numbers, kth-element, sieve

# CodeForge - B38A - Số Siêu Nguyên Tố

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Số siêu nguyên tố là số nguyên tố mà khi bỏ dần chữ số từ phải sang trái, mỗi số thu được vẫn là số nguyên tố.

Ví dụ: 233 → 23 → 2 (đều là số nguyên tố)

Kiểm tra một số có phải là số siêu nguyên tố không.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra YES nếu là số siêu nguyên tố
- In ra NO nếu không phải

### ◊ Constraints

- $1 \leq N \leq 10^9$

## Ví dụ

Test case 1

**Input:**

```
2
```

**Output:**

```
YES
```

Test case 2

**Input:**

```
23
```

**Output:**

```
YES
```

## Test case 3

**Input:**

```
233
```

**Output:**

```
YES
```

## Test case 4

**Input:**

```
123
```

**Output:**

```
NO
```

---

**Tags:** super-prime, prime-numbers, digit-manipulation

---

# CodeForge - B39A - Tổng Ước Hoàn Hảo

---

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Tính tổng tất cả các số hoàn hảo từ 1 đến N.

Số hoàn hảo là số bằng tổng các ước thực sự của nó.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra tổng các số hoàn hảo từ 1 đến N

### ◊ Constraints

- $1 \leq N \leq 10000$

## Ví dụ

Test case 1

**Input:**

```
10
```

**Output:**

```
6
```

Test case 2

**Input:**

```
100
```

**Output:**

34

**Giải thích:**  $6 + 28 = 34$

Test case 3

**Input:**

5

**Output:**

0

---

**Tags:** perfect-number, sum, loop

# CodeForge - B40A - Dãy Collatz

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

In dãy Collatz bắt đầu từ N cho đến khi về 1.

Quy tắc:

- Nếu N chẵn:  $N = N / 2$
- Nếu N lẻ:  $N = 3N + 1$

### ◊ Input

- Một số nguyên dương **N**

### ◊ Output

- In ra dãy Collatz, mỗi số cách nhau bởi dấu cách

### ◊ Constraints

- $1 \leq N \leq 10^6$

## Ví dụ

Test case 1

**Input:**

```
6
```

**Output:**

```
6 3 10 5 16 8 4 2 1
```

Test case 2

**Input:**

```
10
```

**Output:**

```
10 5 16 8 4 2 1
```

Test case 3

**Input:**

```
1
```

**Output:**

```
1
```

---

**Tags:** [collatz-sequence](#), [while-loop](#), [mathematical-sequence](#)

# CodeForge - B41A - Số Chính Phương Liên Tiếp

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

In ra N số chính phương đầu tiên, mỗi số cách nhau bởi dấu cách.

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra N số chính phương đầu tiên trên một dòng, cách nhau bởi dấu cách

### ◊ Constraints

- $1 \leq N \leq 10000$

## Ví dụ

Test case 1

**Input:**

```
5
```

**Output:**

```
1 4 9 16 25
```

Test case 2

**Input:**

```
10
```

**Output:**

```
1 4 9 16 25 36 49 64 81 100
```

### Test case 3

**Input:**

```
1
```

**Output:**

```
1
```

---

**Tags:** perfect-square, sequence, loop

# CodeForge - B42A - In Hình Số Hình Thoi

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

In hình thoi bằng các số.

```
1
121
12321
1234321
12321
121
1
```

### ◊ Input

- Một số nguyên dương  $N$  ( $N \geq 2$ )

### ◊ Output

- In ra hình thoi số với chiều cao  $2N-1$

### ◊ Constraints

- $2 \leq N \leq 9$

## Ví dụ

Test case 1

**Input:**

```
4
```

**Output:**

```
1
121
12321
1234321
12321
121
1
```

## Test case 2

**Input:**

```
3
```

**Output:**

```
1
121
12321
121
1
```

## Test case 3

**Input:**

```
2
```

**Output:**

```
1
121
1
```

---

**Tags:** pattern, diamond, numbers, nested-loop

# CodeForge - B43A - Tổng Chữ Số Đến Một Chữ Số

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Tính tổng các chữ số liên tục cho đến khi còn một chữ số.

Ví dụ:

- $38 \rightarrow 3+8=11 \rightarrow 1+1=2$

### ◊ Input

- Một số nguyên không âm  $N$

### ◊ Output

- In ra kết quả cuối cùng (một chữ số)

### ◊ Constraints

- $0 \leq N \leq 10^{18}$

## Ví dụ

Test case 1

**Input:**

```
38
```

**Output:**

```
2
```

Test case 2

**Input:**

```
9999
```

**Output:**

9

### Test case 3

**Input:**

0

**Output:**

0

### Test case 4

**Input:**

12345

**Output:**

6

---

**Tags:** [digit-sum](#), [recursive-sum](#), [loop](#)

# CodeForge - B44A - In Hình Lưới Cờ Vua

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

In hình lưới cờ vua  $N \times N$  bằng dấu \* và khoảng trắng.

```
* * * *
* * *
* * * *
* * *
```

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra hình lưới cờ vua  $N \times N$

### ◊ Constraints

- $2 \leq N \leq 50$

## Ví dụ

Test case 1

**Input:**

```
4
```

**Output:**

```
* * * *
* * *
* * * *
* * *
```

Test case 2

**Input:**

```
5
```

**Output:**

```
* * * * *
* * * *
* * * * *
* * * *
* * * * *
```

## Test case 3

**Input:**

```
3
```

**Output:**

```
* * *
* *
* * *
```

---

**Tags:** pattern, chessboard, nested-loop

# CodeForge - B45A - Tìm Chu Kỳ Số

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Cho một số N, tìm độ dài chu kỳ khi áp dụng phép toán:

- Nếu N chẵn:  $N = N / 2$
- Nếu N lẻ:  $N = 3N + 1$

Đếm số bước cho đến khi về 1 (dãy Collatz).

### ◊ Input

- Một số nguyên dương  $N$

### ◊ Output

- In ra số bước để về 1

### ◊ Constraints

- $1 \leq N \leq 10^6$

## Ví dụ

Test case 1

**Input:**

```
6
```

**Output:**

```
8
```

**Giải thích:** 6 → 3 → 10 → 5 → 16 → 8 → 4 → 2 → 1 (8 bước)

Test case 2

**Input:**

```
10
```

**Output:**

```
6
```

Test case 3

**Input:**

```
1
```

**Output:**

```
0
```

Test case 4

**Input:**

```
27
```

**Output:**

```
111
```

---

**Tags:** [collatz-sequence](#), [counting](#), [while-loop](#)