

CodeForge - B01 - Random Integer Cơ Bản

Độ khó: ★ Easy

Đề bài

Sinh một số nguyên ngẫu nhiên sử dụng `Random.nextInt()`.

◇ Input

- Không có input

◇ Output

- In ra một số nguyên ngẫu nhiên

◇ Constraints

- Sử dụng `java.util.Random`

Ví dụ

Test case 1

Output:

```
-1234567890
```

Giải thích: Kết quả ngẫu nhiên, mỗi lần chạy khác nhau

Test case 2

Output:

```
987654321
```

Test case 3

Output:

```
0
```

Tags: `random`, `nextInt`, `basic`

CodeForge - B02 - Random Integer Trong Range [0, N)

Độ khó: ★ Easy

Đề bài

Sinh số nguyên ngẫu nhiên trong khoảng [0, N) sử dụng `Random.nextInt(bound)`.

◇ Input

- Một số nguyên dương N

◇ Output

- In ra số nguyên ngẫu nhiên từ 0 đến N-1

◇ Constraints

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

Ví dụ

Test case 1

Input:

10

Output:

7

Giải thích: Số từ 0 đến 9

Test case 2

Input:

100

Output:

42

Test case 3

Input:

6

Output:

3

Tags: `random`, `nextInt`, `bound`

CodeForge - B03 - Random Integer Trong Range [Min, Max]

Độ khó: ★ ★ Medium

Đề bài

Sinh số nguyên ngẫu nhiên trong khoảng [Min, Max] (bao gồm cả Min và Max).

Công thức: $\text{Min} + \text{random.nextInt}(\text{Max} - \text{Min} + 1)$

◇ Input

- Dòng 1: Số nguyên Min
- Dòng 2: Số nguyên Max ($\text{Min} \leq \text{Max}$)

◇ Output

- In ra số nguyên ngẫu nhiên trong [Min, Max]

◇ Constraints

- $-10^9 \leq \text{Min} \leq \text{Max} \leq 10^9$

Ví dụ

Test case 1

Input:

```
1
10
```

Output:

```
7
```

Test case 2

Input:

```
-5
5
```

Output:

```
-2
```

Test case 3**Input:**

```
100  
100
```

Output:

```
100
```

Tags: `random`, `range`, `min-max`

CodeForge - B04 - Random Double

Độ khó: ★ Easy

Đề bài

Sinh số thực ngẫu nhiên trong khoảng $[0.0, 1.0)$ sử dụng `Random.nextDouble()`.

◇ Input

- Không có input

◇ Output

- In ra số thực ngẫu nhiên (làm tròn 6 chữ số thập phân)

◇ Constraints

- Kết quả trong khoảng $[0.0, 1.0)$

Ví dụ

Test case 1

Output:

```
0.573921
```

Test case 2

Output:

```
0.123456
```

Test case 3

Output:

```
0.999999
```

Tags: `random`, `nextDouble`, `double`

CodeForge - B05 - Random Boolean

Độ khó: ★ Easy

Đề bài

Sinh giá trị boolean ngẫu nhiên sử dụng `Random.nextBoolean()`.

◇ Input

- Không có input

◇ Output

- In ra `true` hoặc `false`

◇ Constraints

- Xác suất 50-50

Ví dụ

Test case 1

Output:

```
true
```

Test case 2

Output:

```
false
```

Test case 3

Output:

```
true
```

Tags: `random`, `nextBoolean`, `boolean`

CodeForge - B06 - Tung Xúc Xắc

Độ khó: ★ Easy

Đề bài

Mô phỏng tung xúc xắc (1-6) N lần và in ra kết quả.

◇ Input

- Số lần tung N

◇ Output

- In ra N số, mỗi số trên một dòng

◇ Constraints

- $1 \leq N \leq 100$

Ví dụ

Test case 1

Input:

```
5
```

Output:

```
4
2
6
1
3
```

Test case 2

Input:

```
3
```

Output:


```
5  
5  
2
```

Tags: random, dice, simulation

CodeForge - B07 - Random Từ Mảng

Độ khó: ★ ★ Medium

Đề bài

Chọn ngẫu nhiên một phần tử từ mảng.

◇ Input

- Dòng 1: Số phần tử N
- Dòng 2: N chuỗi cách nhau bởi dấu cách

◇ Output

- In ra một phần tử được chọn ngẫu nhiên

◇ Constraints

- $1 \leq N \leq 1000$

Ví dụ

Test case 1

Input:

```
5
apple banana orange grape mango
```

Output:

```
banana
```

Test case 2

Input:

```
3
Java Python C++
```

Output:

Python

Tags: random, array, selection

CodeForge - B08 - Trộn Mảng (Shuffle)

Độ khó: ★ ★ Medium

Đề bài

Trộn ngẫu nhiên các phần tử trong mảng (Fisher-Yates shuffle).

◇ Input

- Dòng 1: Số phần tử N
- Dòng 2: N số nguyên cách nhau bởi dấu cách

◇ Output

- In ra mảng sau khi trộn

◇ Constraints

- $1 \leq N \leq 1000$
- $-10^9 \leq \text{mỗi phần tử} \leq 10^9$

Ví dụ

Test case 1

Input:

```
5
1 2 3 4 5
```

Output:

```
3 1 5 2 4
```

Test case 2

Input:

```
3
10 20 30
```

Output:

```
20 30 10
```

Tags: random, shuffle, array

CodeForge - B09 - Math.random()

Độ khó: ★ Easy

Đề bài

Sinh số nguyên ngẫu nhiên trong [Min, Max] sử dụng `Math.random()`.

Công thức: `(int)(Math.random() * (Max - Min + 1)) + Min`

◇ Input

- Dòng 1: Min
- Dòng 2: Max

◇ Output

- In ra số ngẫu nhiên trong [Min, Max]

◇ Constraints

- $0 \leq \text{Min} \leq \text{Max} \leq 1000$

Ví dụ

Test case 1

Input:

```
1
10
```

Output:

```
7
```

Test case 2

Input:

```
50
100
```

Output:

78

Tags: random, math-random, range

CodeForge - B10 - Mật Khẩu Ngẫu Nhiên

Độ khó: ★ ★ Medium

Đề bài

Sinh mật khẩu ngẫu nhiên gồm N ký tự (chữ cái và số).

◇ Input

- Độ dài mật khẩu N

◇ Output

- In ra mật khẩu ngẫu nhiên

◇ Constraints

- $4 \leq N \leq 20$

Ví dụ

Test case 1

Input:

8

Output:

aB3xY9zQ

Test case 2

Input:

12

Output:

k7Lm2Pq5Rs8T

Test case 3

Input:

6

Output:

Jx4K9m

Tags: random, password, string-generation

CodeForge - B11 - LocalDate Hôm Nay

Độ khó: ★ Easy

Đề bài

Lấy ngày hôm nay sử dụng `LocalDate.now()`.

◇ Input

- Không có input

◇ Output

- In ra ngày hôm nay theo format: yyyy-MM-dd

◇ Constraints

- Sử dụng `java.time.LocalDate`

Ví dụ

Test case 1

Output:

```
2024-12-22
```

Test case 2

Output:

```
2024-01-15
```

Tags: `datetime`, `localdate`, `now`

CodeForge - B12 - Tạo LocalDate Từ Năm Tháng Ngày

Độ khó: ★ Easy

Đề bài

Tạo `LocalDate` từ năm, tháng, ngày sử dụng `LocalDate.of()`.

◇ Input

- Dòng 1: Năm (year)
- Dòng 2: Tháng (month, 1-12)
- Dòng 3: Ngày (day)

◇ Output

- In ra `LocalDate` theo format: yyyy-MM-dd

◇ Constraints

- $1900 \leq \text{year} \leq 2100$
- $1 \leq \text{month} \leq 12$
- $1 \leq \text{day} \leq 31$ (hợp lệ cho tháng đó)

Ví dụ

Test case 1

Input:

```
2024
12
25
```

Output:

```
2024-12-25
```

Test case 2

Input:

```
2000
1
```

```
1
```

Output:

```
2000-01-01
```

Tags: `datetime`, `localdate`, `of`

CodeForge - B13 - Lấy Năm Tháng Ngày

Độ khó: ★ Easy

Đề bài

Parse LocalDate từ chuỗi và lấy năm, tháng, ngày.

Sử dụng `getYear()`, `getMonthValue()`, `getDayOfMonth()`.

◇ Input

- Một chuỗi ngày tháng format: yyyy-MM-dd

◇ Output

- Dòng 1: Năm
- Dòng 2: Tháng
- Dòng 3: Ngày

◇ Constraints

- Ngày hợp lệ

Ví dụ

Test case 1

Input:

```
2024-12-25
```

Output:

```
2024
12
25
```

Test case 2

Input:

```
2000-01-01
```

Output:

```
2000  
1  
1
```

Tags: `datetime`, `localdate`, `getters`

CodeForge - B14 - Ngày Trong Tuần

Độ khó: ★ Easy

Đề bài

Tìm thứ trong tuần của một ngày sử dụng `getDayOfWeek()`.

◇ Input

- Một chuỗi ngày tháng format: yyyy-MM-dd

◇ Output

- In ra thứ trong tuần (MONDAY, TUESDAY, ...)

◇ Constraints

- Ngày hợp lệ

Ví dụ

Test case 1

Input:

2024-12-25

Output:

WEDNESDAY

Test case 2

Input:

2024-01-01

Output:

MONDAY

Test case 3

Input:

2000-01-01

Output:

SATURDAY

Tags: `datetime`, `localdate`, `day-of-week`

CodeForge - B15 - Cộng Ngày

Độ khó: ★ ★ Medium

Đề bài

Cộng N ngày vào một ngày sử dụng `plusDays()`.

◇ Input

- Dòng 1: Ngày gốc (yyyy-MM-dd)
- Dòng 2: Số ngày cần cộng N

◇ Output

- In ra ngày sau khi cộng

◇ Constraints

- $-10000 \leq N \leq 10000$

Ví dụ

Test case 1

Input:

```
2024-12-25
7
```

Output:

```
2025-01-01
```

Test case 2

Input:

```
2024-01-01
-1
```

Output:

```
2023-12-31
```

Test case 3

Input:

```
2024-02-28  
1
```

Output:

```
2024-02-29
```

Giải thích: 2024 là năm nhuận

Tags: `datetime`, `localdate`, `plusDays`

CodeForge - B16 - Trừ Tháng

Độ khó: ★ ★ Medium

Đề bài

Trừ N tháng từ một ngày sử dụng `minusMonths()`.

◇ Input

- Dòng 1: Ngày gốc (yyyy-MM-dd)
- Dòng 2: Số tháng cần trừ N

◇ Output

- In ra ngày sau khi trừ

◇ Constraints

- $0 \leq N \leq 1200$

Ví dụ

Test case 1

Input:

```
2024-12-25
6
```

Output:

```
2024-06-25
```

Test case 2

Input:

```
2024-03-31
1
```

Output:

2024-02-29

Giải thích: Tháng 2 năm 2024 có 29 ngày

Test case 3

Input:

2024-01-01
1

Output:

2023-12-01

Tags: `datetime`, `localdate`, `minusMonths`

CodeForge - B17 - So Sánh Ngày

Độ khó: ★ ★ Medium

Đề bài

So sánh hai ngày sử dụng `isAfter()`, `isBefore()`, `isEqual()`.

◇ Input

- Dòng 1: Ngày thứ nhất (yyyy-MM-dd)
- Dòng 2: Ngày thứ hai (yyyy-MM-dd)

◇ Output

- In ra **AFTER** nếu ngày 1 sau ngày 2
- In ra **BEFORE** nếu ngày 1 trước ngày 2
- In ra **EQUAL** nếu bằng nhau

◇ Constraints

- Ngày hợp lệ

Ví dụ

Test case 1

Input:

```
2024-12-25
2024-01-01
```

Output:

```
AFTER
```

Test case 2

Input:

```
2024-01-01
2024-12-31
```

Output:

BEFORE

Test case 3

Input:

2024-12-25
2024-12-25

Output:

EQUAL

Tags: `datetime`, `localdate`, `comparison`

CodeForge - B18 - Kiểm Tra Năm Nhuận

Độ khó: ★ Easy

Đề bài

Kiểm tra năm có phải năm nhuận không sử dụng `LocalDate.isLeapYear()`.

◇ Input

- Một số nguyên year

◇ Output

- In ra `LEAP YEAR` nếu là năm nhuận
- In ra `NOT LEAP YEAR` nếu không phải

◇ Constraints

- $1 \leq \text{year} \leq 3000$

Ví dụ

Test case 1

Input:

2024

Output:

LEAP YEAR

Test case 2

Input:

1900

Output:

NOT LEAP YEAR

Test case 3

Input:

2000

Output:

LEAP YEAR

Tags: `datetime`, `localdate`, `leap-year`

CodeForge - B19 - LocalTime Hiện Tại

Độ khó: ★ Easy

Đề bài

Lấy thời gian hiện tại sử dụng `LocalTime.now()`.

◇ Input

- Không có input

◇ Output

- In ra thời gian hiện tại (HH:mm:ss)

◇ Constraints

- Sử dụng `java.time.LocalTime`

Ví dụ

Test case 1

Output:

```
14:30:45
```

Test case 2

Output:

```
09:15:23
```

Tags: `datetime`, `localtime`, `now`

CodeForge - B20 - LocalDateTime

Độ khó: ★ ★ Medium

Đề bài

Tạo `LocalDateTime` và tách ra `LocalDate` và `LocalTime`.

Sử dụng `toLocalDate()` và `toLocalTime()`.

◇ Input

- Một chuỗi datetime format: yyyy-MM-ddTHH:mm:ss

◇ Output

- Dòng 1: `LocalDate` (yyyy-MM-dd)
- Dòng 2: `LocalTime` (HH:mm:ss)

◇ Constraints

- Datetime hợp lệ

Ví dụ

Test case 1

Input:

```
2024-12-25T14:30:00
```

Output:

```
2024-12-25  
14:30:00
```

Test case 2

Input:

```
2000-01-01T00:00:00
```

Output:

```
2000-01-01  
00:00:00
```

Tags: `datetime`, `localdatetime`, `split`

CodeForge - B21A - Random Với Seed

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Sinh chuỗi số ngẫu nhiên với seed để có kết quả lặp lại được.

Tạo `Random(seed)` và sinh N số trong khoảng $[0, 100)$.

◇ Input

- Dòng 1: Seed (long)
- Dòng 2: Số lượng số cần sinh N

◇ Output

- In ra N số, mỗi số trên một dòng

◇ Constraints

- $1 \leq N \leq 100$
- $0 \leq seed \leq 10^{18}$

Ví dụ

Test case 1

Input:

```
12345
5
```

Output:

```
51
80
41
28
55
```

Giải thích: Với cùng seed, kết quả luôn giống nhau

Test case 2

Input:

```
12345  
5
```

Output:

```
51  
80  
41  
28  
55
```

Giải thích: Cùng seed → cùng kết quả

Tags: random, seed, reproducible

CodeForge - B22A - ThreadLocalRandom

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Sử dụng `ThreadLocalRandom` để sinh số ngẫu nhiên trong multithreading context.

Sinh N số trong khoảng [Min, Max].

◇ Input

- Dòng 1: Min
- Dòng 2: Max
- Dòng 3: Số lượng N

◇ Output

- In ra N số ngẫu nhiên, mỗi số trên một dòng

◇ Constraints

- $1 \leq N \leq 100$
- $-10^9 \leq \text{Min} \leq \text{Max} \leq 10^9$

Ví dụ

Test case 1

Input:

```
1
10
5
```

Output:

```
7
3
9
2
5
```

Test case 2

Input:

```
-5  
5  
3
```

Output:

```
-2  
4  
0
```

Tags: `random`, `threadlocalrandom`, `concurrent`

CodeForge - B23A - Tính Số Ngày Giữa Hai Ngày

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Tính số ngày giữa hai ngày sử dụng `ChronoUnit.DAYS.between()` hoặc `Period.between()`.

◇ Input

- Dòng 1: Ngày bắt đầu (yyyy-MM-dd)
- Dòng 2: Ngày kết thúc (yyyy-MM-dd)

◇ Output

- In ra số ngày chênh lệch (có thể âm nếu ngày 1 sau ngày 2)

◇ Constraints

- Ngày hợp lệ

Ví dụ

Test case 1

Input:

```
2024-01-01
2024-12-31
```

Output:

```
365
```

Test case 2

Input:

```
2024-12-31
2024-01-01
```

Output:


```
-365
```

Test case 3

Input:

```
2024-02-28  
2024-03-01
```

Output:

```
2
```

Tags: `datetime`, `localdate`, `between`, `chronounit`

CodeForge - B24A - Tìm Ngày Cuối Tháng

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Tìm ngày cuối cùng của tháng trong một ngày cho trước.

Sử dụng `withDayOfMonth()` và `lengthOfMonth()`.

◇ Input

- Một chuỗi ngày (yyyy-MM-dd)

◇ Output

- In ra ngày cuối tháng (yyyy-MM-dd)

◇ Constraints

- Ngày hợp lệ

Ví dụ

Test case 1

Input:

```
2024-02-15
```

Output:

```
2024-02-29
```

Giải thích: Năm nhuận, tháng 2 có 29 ngày

Test case 2

Input:

```
2023-02-15
```

Output:

2023-02-28

Test case 3

Input:

2024-12-01

Output:

2024-12-31

Tags: `datetime`, `localdate`, `end-of-month`

CodeForge - B25A - Tính Tuổi

Độ khó: ★ ★ ★ Hard (Advanced)

Đề bài

Tính tuổi của một người dựa trên ngày sinh và ngày hiện tại.

Sử dụng `Period.between()`.

◇ Input

- Dòng 1: Ngày sinh (yyyy-MM-dd)
- Dòng 2: Ngày hiện tại (yyyy-MM-dd)

◇ Output

- Dòng 1: Số năm
- Dòng 2: Số tháng (sau khi trừ năm)
- Dòng 3: Số ngày (sau khi trừ năm và tháng)

◇ Constraints

- Ngày sinh ≤ Ngày hiện tại

Ví dụ

Test case 1

Input:

```
2000-01-01
2024-12-22
```

Output:

```
24
11
21
```

Giải thích: 24 năm, 11 tháng, 21 ngày

Test case 2

Input:

```
2020-06-15
2024-06-14
```

Output:

```
3
11
30
```

Test case 3

Input:

```
2024-01-01
2024-01-01
```

Output:

```
0
0
0
```

Tags: `datetime`, `period`, `age-calculation`, `between`