

# CodeForge - B01 - Format LocalDate Cơ Bản

**Độ khó:** ★ Easy

## Đề bài

Format LocalDate theo pattern "dd/MM/yyyy" sử dụng [DateTimeFormatter](#).

### ◊ Input

- Một chuỗi ngày (yyyy-MM-dd)

### ◊ Output

- In ra ngày theo format dd/MM/yyyy

### ◊ Constraints

- Ngày hợp lệ

## Ví dụ

Test case 1

### Input:

```
2024-12-25
```

### Output:

```
25/12/2024
```

Test case 2

### Input:

```
2024-01-01
```

### Output:

```
01/01/2024
```

### Test case 3

**Input:**

```
2000-02-29
```

**Output:**

```
29/02/2000
```

---

**Tags:** `datetime`, `formatter`, `format`, `pattern`

# CodeForge - B02 - Parse Ngày Từ Chuỗi

**Độ khó:** ★ Easy

## Đề bài

Parse chuỗi ngày theo format "dd/MM/yyyy" thành LocalDate.

### ◊ Input

- Một chuỗi ngày (dd/MM/yyyy)

### ◊ Output

- In ra LocalDate theo format ISO (yyyy-MM-dd)

### ◊ Constraints

- Ngày hợp lệ

## Ví dụ

Test case 1

### Input:

```
25/12/2024
```

### Output:

```
2024-12-25
```

Test case 2

### Input:

```
01/01/2000
```

### Output:

```
2000-01-01
```

### Test case 3

**Input:**

```
29/02/2024
```

**Output:**

```
2024-02-29
```

---

**Tags:** `datetime`, `formatter`, `parse`, `pattern`

# CodeForge - B03 - Format Với Tên Tháng

**Độ khó:** ★ ★ Medium

## Đề bài

Format LocalDate theo pattern "dd MMM, yyyy" (tên tháng viết tắt).

Ví dụ: "25 Dec, 2024"

### ◊ Input

- Một chuỗi ngày (yyyy-MM-dd)

### ◊ Output

- In ra ngày theo format "dd MMM, yyyy"

### ◊ Constraints

- Ngày hợp lệ

## Ví dụ

Test case 1

### Input:

```
2024-12-25
```

### Output:

```
25 Dec, 2024
```

Test case 2

### Input:

```
2024-01-01
```

### Output:

01 Jan, 2024

### Test case 3

**Input:**

2024-07-04

**Output:**

04 Jul, 2024

---

**Tags:** `datetime`, `formatter`, `month-name`, `pattern`

# CodeForge - B04 - Format DateTime Đầy Đủ

**Độ khó:** ★ ★ Medium

## Đề bài

Format LocalDateTime theo pattern "dd/MM/yyyy HH:mm:ss".

### ◊ Input

- Một chuỗi datetime ISO (yyyy-MM-ddTHH:mm:ss)

### ◊ Output

- In ra datetime theo format "dd/MM/yyyy HH:mm:ss"

### ◊ Constraints

- DateTime hợp lệ

## Ví dụ

Test case 1

### Input:

```
2024-12-25T14:30:45
```

### Output:

```
25/12/2024 14:30:45
```

Test case 2

### Input:

```
2024-01-01T00:00:00
```

### Output:

```
01/01/2024 00:00:00
```

### Test case 3

**Input:**

```
2024-07-04T23:59:59
```

**Output:**

```
04/07/2024 23:59:59
```

---

**Tags:** `datetime`, `formatter`, `localdatetime`, `pattern`

# CodeForge - B05 - Predefined Formatters

**Độ khó:** ★ Easy

## Đề bài

Sử dụng predefined formatter `DateTimeFormatter.ISO_LOCAL_DATE`.

### ◊ Input

- Một LocalDate (yyyy-MM-dd)

### ◊ Output

- In ra ngày sử dụng ISO\_LOCAL\_DATE formatter

### ◊ Constraints

- Ngày hợp lệ

## Ví dụ

Test case 1

### Input:

```
2024-12-25
```

### Output:

```
2024-12-25
```

Test case 2

### Input:

```
2024-01-01
```

### Output:

```
2024-01-01
```

**Tags:** `datetime`, `formatter`, `iso`, `predefined`

# CodeForge - B06 - Period Giữa Hai Ngày

**Độ khó:** ★ ★ Medium

## 📝 Đề bài

Tính Period giữa hai ngày và in ra số năm, tháng, ngày.

Sử dụng `Period.between()`.

### ◊ Input

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

### ◊ Output

- Dòng 1: Số năm
- Dòng 2: Số tháng
- Dòng 3: Số ngày

### ◊ Constraints

- Ngày bắt đầu ≤ Ngày kết thúc

## 📊 Ví dụ

Test case 1

**Input:**

```
2020-01-01  
2024-12-25
```

**Output:**

```
4  
11  
24
```

Test case 2

**Input:**

```
2024-01-01  
2024-06-15
```

**Output:**

```
0  
5  
14
```

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

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

**Output:**

```
0  
0  
0
```

---

**Tags:** `datetime`, `period`, `between`, `calculation`

# CodeForge - B07 - Cộng Period

**Độ khó:** ★ ★ Medium

## Đề bài

Cộng một Period (years, months, days) vào một ngày.

### ◊ Input

- Dòng 1: Ngày gốc (yyyy-MM-dd)
- Dòng 2: Số năm cần cộng
- Dòng 3: Số tháng cần cộng
- Dòng 4: Số ngày cần cộng

### ◊ Output

- In ra ngày sau khi cộng Period

### ◊ Constraints

- Ngày hợp lệ
- $-100 \leq \text{years, months, days} \leq 100$

## Ví dụ

Test case 1

### Input:

```
2024-01-01
1
6
15
```

### Output:

```
2025-07-16
```

Test case 2

### Input:

```
2024-12-31  
0  
0  
1
```

**Output:**

```
2025-01-01
```

## Test case 3

**Input:**

```
2024-02-29  
1  
0  
0
```

**Output:**

```
2025-02-28
```

**Giải thích:** 2025 không phải năm nhuận

---

**Tags:** `datetime`, `period`, `plus`, `localdate`

# CodeForge - B08 - Duration Giữa Hai Thời Gian

**Độ khó:** ★ ★ Medium

## Đề bài

Tính Duration giữa hai LocalTime và in ra số giờ, phút, giây.

Sử dụng `Duration.between()`.

### ◊ Input

- Dòng 1: Thời gian bắt đầu (HH:mm:ss)
- Dòng 2: Thời gian kết thúc (HH:mm:ss)

### ◊ Output

- Dòng 1: Tổng số giờ
- Dòng 2: Số phút còn lại (sau khi trừ giờ)
- Dòng 3: Số giây còn lại (sau khi trừ giờ và phút)

### ◊ Constraints

- Thời gian hợp lệ

## Ví dụ

Test case 1

**Input:**

```
09:00:00
17:30:45
```

**Output:**

```
8
30
45
```

Test case 2

**Input:**

```
23:00:00  
01:00:00
```

**Output:**

```
-22  
0  
0
```

**Giải thích:** Kết thúc trước bắt đầu

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

```
10:15:30  
10:15:30
```

**Output:**

```
0  
0  
0
```

---

**Tags:** `datetime`, `duration`, `between`, `localtime`

# CodeForge - B09 - Duration To Seconds

**Độ khó:** ★ Easy

## Đề bài

Tính tổng số giây trong một Duration.

Sử dụng `Duration.toSeconds()`.

### ◊ Input

- Dòng 1: Thời gian bắt đầu (HH:mm:ss)
- Dòng 2: Thời gian kết thúc (HH:mm:ss)

### ◊ Output

- In ra tổng số giây

### ◊ Constraints

- Thời gian hợp lệ

## Ví dụ

Test case 1

**Input:**

```
09:00:00
10:00:00
```

**Output:**

```
3600
```

Test case 2

**Input:**

```
09:00:00
09:01:30
```

**Output:**

```
90
```

## Test case 3

**Input:**

```
00:00:00  
23:59:59
```

**Output:**

```
86399
```

---

**Tags:** `datetime`, `duration`, `toSeconds`, `conversion`

# CodeForge - B10 - ZonedDateTime Basics

**Độ khó:** ★ ★ Medium

## Đề bài

Tạo ZonedDateTime cho timezone cụ thể và in ra.

Sử dụng `ZonedDateTime.now(ZoneId.of())`.

### ◊ Input

- Một timezone ID (ví dụ: "Asia/Ho\_Chi\_Minh")

### ◊ Output

- In ra ZonedDateTime hiện tại cho timezone đó

### ◊ Constraints

- Timezone hợp lệ

## Ví dụ

Test case 1

**Input:**

```
Asia/Ho_Chi_Minh
```

**Output:**

```
2024-12-25T14:30:00+07:00[Asia/Ho_Chi_Minh]
```

Test case 2

**Input:**

```
America/New_York
```

**Output:**

```
2024-12-25T02:30:00-05:00[America/New_York]
```

## Test case 3

**Input:**

```
UTC
```

**Output:**

```
2024-12-25T07:30:00Z[UTC]
```

---

**Tags:** `datetime`, `zoneddatetime`, `timezone`, `zoneid`

# CodeForge - B11 - Chuyển Timezone

**Độ khó:** ★ ★ ★ Hard

## Đề bài

Chuyển đổi thời gian từ timezone này sang timezone khác.

Sử dụng `withZoneSameInstant()`.

### ◊ Input

- Dòng 1: Datetime ISO (yyyy-MM-ddTHH:mm:ss)
- Dòng 2: Timezone gốc
- Dòng 3: Timezone đích

### ◊ Output

- In ra datetime ở timezone đích

### ◊ Constraints

- Datetime và timezone hợp lệ

## Ví dụ

Test case 1

**Input:**

```
2024-12-25T14:00:00
Asia/Ho_Chi_Minh
America/New_York
```

**Output:**

```
2024-12-25T02:00:00-05:00[America/New_York]
```

**Giải thích:** UTC+7 → UTC-5 = -12 giờ

Test case 2

**Input:**

```
2024-12-25T12:00:00
UTC
Asia/Tokyo
```

**Output:**

```
2024-12-25T21:00:00+09:00[Asia/Tokyo]
```

---

**Tags:** `datetime`, `zoneddatetime`, `timezone-conversion`

# CodeForge - B12 - Instant Hiện Tại

**Độ khó:** ★ Easy

## Đề bài

Lấy timestamp hiện tại sử dụng `Instant.now()`.

### ◊ Input

- Không có input

### ◊ Output

- In ra Instant hiện tại

### ◊ Constraints

- N/A

## Ví dụ

Test case 1

**Output:**

```
2024-12-25T07:30:45.123456Z
```

Test case 2

**Output:**

```
2024-01-01T00:00:00.000000Z
```

---

**Tags:** `datetime`, `instant`, `timestamp`, `now`

# CodeForge - B13 - Instant To EpochMilli

**Độ khó:** ★ Easy

## Đề bài

Chuyển Instant sang epoch milliseconds.

Sử dụng `toEpochMilli()`.

### ◊ Input

- Một Instant (ISO format)

### ◊ Output

- In ra epoch milliseconds

### ◊ Constraints

- Instant hợp lệ

## Ví dụ

Test case 1

**Input:**

```
1970-01-01T00:00:00Z
```

**Output:**

```
0
```

Test case 2

**Input:**

```
2024-01-01T00:00:00Z
```

**Output:**

```
1704067200000
```

### Test case 3

**Input:**

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

**Output:**

```
946684800000
```

---

**Tags:** `datetime`, `instant`, `epoch`, `conversion`

# CodeForge - B14 - EpochMilli To Instant

**Độ khó:** ★ Easy

## Đề bài

Tạo Instant từ epoch milliseconds.

Sử dụng `Instant.ofEpochMilli()`.

### ◊ Input

- Một số long (epoch milliseconds)

### ◊ Output

- In ra Instant

### ◊ Constraints

- Epoch hợp lệ

## Ví dụ

Test case 1

**Input:**

```
0
```

**Output:**

```
1970-01-01T00:00:00Z
```

Test case 2

**Input:**

```
1704067200000
```

**Output:**

2024-01-01T00:00:00Z

---

**Tags:** `datetime`, `instant`, `epoch`, `conversion`

# CodeForge - B15 - LocalDateTime To Instant

**Độ khó:** ★ ★ Medium

## 📝 Đề bài

Chuyển LocalDateTime sang Instant với timezone.

Sử dụng `atZone()` và `toInstant()`.

### ◊ Input

- Dòng 1: LocalDateTime (yyyy-MM-ddTHH:mm:ss)
- Dòng 2: Timezone ID

### ◊ Output

- In ra Instant

### ◊ Constraints

- DateTime và timezone hợp lệ

## 💻 Ví dụ

Test case 1

**Input:**

```
2024-12-25T14:00:00
Asia/Ho_Chi_Minh
```

**Output:**

```
2024-12-25T07:00:00Z
```

**Giải thích:** UTC+7 → UTC

Test case 2

**Input:**

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

**Output:**

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

---

**Tags:** `datetime`, `instant`, `localdatetime`, `conversion`

# CodeForge - B16 - Instant To LocalDateTime

**Độ khó:** ★ ★ Medium

## Đề bài

Chuyển Instant sang LocalDateTime với timezone.

Sử dụng `atZone()` và `toLocalDateTime()`.

### ◊ Input

- Dòng 1: Instant (ISO format)
- Dòng 2: Timezone ID

### ◊ Output

- In ra LocalDateTime

### ◊ Constraints

- Instant và timezone hợp lệ

## Ví dụ

Test case 1

**Input:**

```
2024-12-25T07:00:00Z
Asia/Ho_Chi_Minh
```

**Output:**

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

**Giải thích:** UTC → UTC+7

Test case 2

**Input:**

```
2024-12-25T07:00:00Z
UTC
```

**Output:**

```
2024-12-25T07:00:00
```

---

**Tags:** `datetime`, `instant`, `localdatetime`, `conversion`

# CodeForge - B17 - Duration Của Một Ngày

**Độ khó:** ★ Easy

## Đề bài

Tạo Duration của N ngày và in ra tổng số giờ.

Sử dụng `Duration.ofDays()`.

### ◊ Input

- Số nguyên dương N

### ◊ Output

- In ra tổng số giờ trong N ngày

### ◊ Constraints

- $1 \leq N \leq 365$

## Ví dụ

Test case 1

**Input:**

```
1
```

**Output:**

```
24
```

Test case 2

**Input:**

```
7
```

**Output:**

168

### Test case 3

**Input:**

30

**Output:**

720

---

**Tags:** `datetime`, `duration`, `ofDays`, `conversion`

# CodeForge - B18 - Cộng Duration

**Độ khó:** ★ ★ Medium

## Đề bài

Cộng một Duration (giờ, phút, giây) vào LocalTime.

### ◊ Input

- Dòng 1: Thời gian gốc (HH:mm:ss)
- Dòng 2: Số giờ cần cộng
- Dòng 3: Số phút cần cộng
- Dòng 4: Số giây cần cộng

### ◊ Output

- In ra thời gian sau khi cộng Duration

### ◊ Constraints

- Thời gian hợp lệ
- $-24 \leq \text{hours} \leq 24$
- $-60 \leq \text{minutes}, \text{seconds} \leq 60$

## Ví dụ

Test case 1

**Input:**

```
10:00:00
2
30
45
```

**Output:**

```
12:30:45
```

Test case 2

**Input:**

```
23:00:00  
2  
0  
0
```

**Output:**

```
01:00:00
```

**Giải thích:** Qua ngày mới

Test case 3

**Input:**

```
10:30:00  
-1  
-15  
0
```

**Output:**

```
09:15:00
```

---

**Tags:** `datetime`, `duration`, `plus`, `localtime`

# CodeForge - B19 - So Sánh Period

**Độ khó:** ★ ★ Medium

## Đề bài

Tính Period cho 2 cặp ngày và so sánh xem cặp nào dài hơn.

### ◊ Input

- Dòng 1-2: Cặp ngày thứ nhất (start1, end1)
- Dòng 3-4: Cặp ngày thứ hai (start2, end2)

### ◊ Output

- In ra FIRST LONGER nếu period 1 dài hơn
- In ra SECOND LONGER nếu period 2 dài hơn
- In ra EQUAL nếu bằng nhau

### ◊ Constraints

- Ngày hợp lệ

## Ví dụ

Test case 1

**Input:**

```
2020-01-01
2024-01-01
2020-01-01
2023-01-01
```

**Output:**

```
FIRST LONGER
```

Test case 2

**Input:**

```
2024-01-01
2024-06-01
```

```
2024-01-01
```

```
2024-12-01
```

**Output:**

```
SECOND LONGER
```

---

**Tags:** [datetime](#), [period](#), [comparison](#)

# CodeForge - B20 - Format Tùy Chỉnh

**Độ khó:** ★ ★ Medium

## Đề bài

Tạo custom DateTimeFormatter với pattern do người dùng cung cấp.

### ◊ Input

- Dòng 1: Ngày (yyyy-MM-dd)
- Dòng 2: Pattern format

### ◊ Output

- In ra ngày theo pattern

### ◊ Constraints

- Ngày và pattern hợp lệ

## Ví dụ

Test case 1

**Input:**

```
2024-12-25
dd-MM-yyyy
```

**Output:**

```
25-12-2024
```

Test case 2

**Input:**

```
2024-12-25
MMMM dd, yyyy
```

**Output:**

December 25, 2024

### Test case 3

**Input:**

```
2024-12-25  
EEEE, MMM dd, yyyy
```

**Output:**

```
Wednesday, Dec 25, 2024
```

---

**Tags:** `datetime`, `formatter`, `custom-pattern`

# CodeForge - B21A - Tính Thời Gian Làm Việc

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Tính tổng thời gian làm việc giữa hai datetime (loại trừ cuối tuần và ngoài giờ hành chính 9-17h).

Đơn giản hóa: chỉ tính các ngày làm việc (T2-T6) trong khoảng 9:00-17:00.

### ◊ Input

- Dòng 1: Start datetime (yyyy-MM-ddTHH:mm:ss)
- Dòng 2: End datetime (yyyy-MM-ddTHH:mm:ss)

### ◊ Output

- In ra số giờ làm việc (làm tròn 2 chữ số thập phân)

### ◊ Constraints

- Start  $\leq$  End
- Trong cùng tuần

## Ví dụ

Test case 1

**Input:**

```
2024-12-23T09:00:00
2024-12-23T17:00:00
```

**Output:**

```
8.00
```

**Giải thích:** Thứ 2, 9h-17h = 8 giờ

Test case 2

**Input:**

```
2024-12-23T14:00:00  
2024-12-24T10:00:00
```

**Output:**

```
4.00
```

**Giải thích:** T2: 14h-17h=3h, T3: 9h-10h=1h

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

```
2024-12-21T09:00:00  
2024-12-22T17:00:00
```

**Output:**

```
0.00
```

**Giải thích:** T7-CN không tính

---

**Tags:** `datetime`, `duration`, `business-logic`, `working-hours`

# CodeForge - B22A - Timezone List

**Độ khó:** ★ ★ ★ Hard (Advanced)

## 📝 Đề bài

Hiển thị thời gian hiện tại ở N timezone khác nhau.

Sử dụng `ZoneId.getAvailableZoneIds()` để lấy danh sách timezone.

### ◊ Input

- Dòng 1: Số lượng timezone N
- N dòng tiếp theo: Các timezone ID

### ◊ Output

- In ra N dòng, mỗi dòng format: "TimezoneID: yyyy-MM-dd HH:mm:ss"

### ◊ Constraints

- $1 \leq N \leq 10$
- Timezone hợp lệ

## 📊 Ví dụ

Test case 1

**Input:**

```
3
Asia/Ho_Chi_Minh
America/New_York
Europe/London
```

**Output:**

```
Asia/Ho_Chi_Minh: 2024-12-25 14:00:00
America/New_York: 2024-12-25 02:00:00
Europe/London: 2024-12-25 07:00:00
```

Test case 2

**Input:**

```
2
UTC
Asia/Tokyo
```

**Output:**

```
UTC: 2024-12-25 07:00:00
Asia/Tokyo: 2024-12-25 16:00:00
```

---

**Tags:** [datetime](#), [zoneddatetime](#), [multiple-timezones](#)

# CodeForge - B23A - Meeting Scheduler

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Tìm thời gian meeting chung giữa 2 người ở 2 timezone khác nhau.

Cho thời gian làm việc của mỗi người (9-17h local time), tìm overlap.

### ◊ Input

- Dòng 1: Timezone người 1
- Dòng 2: Timezone người 2
- Dòng 3: Ngày meeting (yyyy-MM-dd)

### ◊ Output

- In ra khoảng thời gian overlap ở cả 2 timezone
- Format: "TZ1: HH:mm-HH:mm | TZ2: HH:mm-HH:mm"
- Nếu không có overlap, in "NO OVERLAP"

### ◊ Constraints

- Timezone hợp lệ

## Ví dụ

Test case 1

**Input:**

```
Asia/Ho_Chi_Minh
America/New_York
2024-12-25
```

**Output:**

```
Asia/Ho_Chi_Minh: 09:00-17:00 | America/New_York: 21:00-05:00
NO OVERLAP
```

**Giải thích:** NYC giờ làm việc là 21h-5h sáng hôm sau theo giờ VN

Test case 2

**Input:**

```
Asia/Tokyo  
Asia/Ho_Chi_Minh  
2024-12-25
```

**Output:**

```
Asia/Tokyo: 11:00-17:00 | Asia/Ho_Chi_Minh: 09:00-15:00
```

**Giải thích:** Tokyo sớm hơn VN 2 giờ

---

**Tags:** `datetime`, `zoneddatetime`, `timezone-conversion`, `overlap`

# CodeForge - B24A - Parse Multiple Formats

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Parse ngày tháng từ nhiều format khác nhau.

Thử parse với các format: "dd/MM/yyyy", "yyyy-MM-dd", "MM-dd-yyyy", "dd-MMM-yyyy"

### ◊ Input

- Một chuỗi ngày (có thể ở bất kỳ format nào)

### ◊ Output

- In ra ngày theo format ISO (yyyy-MM-dd)
- Nếu không parse được, in "INVALID FORMAT"

### ◊ Constraints

- Chuỗi độ dài ≤ 20

## Ví dụ

Test case 1

**Input:**

```
25/12/2024
```

**Output:**

```
2024-12-25
```

Test case 2

**Input:**

```
2024-12-25
```

**Output:**

2024-12-25

### Test case 3

**Input:**

25-Dec-2024

**Output:**

2024-12-25

### Test case 4

**Input:**

invalid date

**Output:**

INVALID FORMAT

---

**Tags:** `datetime`, `parser`, `multiple-formats`, `error-handling`

# CodeForge - B25A - Event Countdown

**Độ khó:** ★ ★ ★ Hard (Advanced)

## Đề bài

Tính thời gian còn lại đến một sự kiện.

In ra số năm, tháng, ngày, giờ, phút, giây còn lại.

### ◊ Input

- Dòng 1: Thời gian hiện tại (yyyy-MM-ddTHH:mm:ss)
- Dòng 2: Thời gian sự kiện (yyyy-MM-ddTHH:mm:ss)

### ◊ Output

- In ra format: "X years, Y months, Z days, H hours, M minutes, S seconds"
- Nếu sự kiện đã qua, in "EVENT PASSED"

### ◊ Constraints

- Datetime hợp lệ

## Ví dụ

Test case 1

**Input:**

```
2024-01-01T10:00:00
2024-12-31T23:59:59
```

**Output:**

```
0 years, 11 months, 30 days, 13 hours, 59 minutes, 59 seconds
```

Test case 2

**Input:**

```
2024-12-25T14:00:00
2025-01-01T00:00:00
```

**Output:**

```
0 years, 0 months, 6 days, 10 hours, 0 minutes, 0 seconds
```

## Test case 3

**Input:**

```
2024-12-31T23:59:59  
2024-01-01T00:00:00
```

**Output:**

```
EVENT PASSED
```

---

**Tags:** `datetime`, `period`, `duration`, `countdown`, `calculation`