

# Bài Tập Codeforces 1A: Theatre Square "Lát Sân Quảng Trường"

## C++ Competitive Programming Series - Elo 800

Học Cùng C++

# 1. Hình Dung Bài Toán

## Đề bài

Bạn có sân kích thước  $n \times m$ . Cần lát kín bằng các viên gạch vuông  $a \times a$ .

- Không được cắt gạch.
- Được phép lát thừa ra ngoài.
- Gạch phải đặt thẳng hàng.

## Mục tiêu

Tìm số lượng viên gạch ÍT NHẤT cần dùng.

## 2. Sai Lầm Thường Gặp: Tính Diện Tích

Nhiều bạn nghĩ:

$$\text{Số gạch} = \frac{\text{Diện tích sân}}{\text{Diện tích gạch}} = \frac{n \times m}{a \times a}$$

Tại sao sai?

- Ví dụ: Sân  $1 \times 5$ , Gạch  $2 \times 2$ .
- Diện tích sân = 5. Diện tích gạch = 4.
- Phép chia:  $5/4 = 1.25$ .
- Thực tế: Chiều rộng 1 (cần 1 gạch 2m). Chiều dài 5 (cần 3 gạch 2m).
- Tổng cần:  $1 \times 3 = 3$  viên. (Khác xa so với 1.25).

### 3. Tư Duy Đúng: Chia Theo Cạnh

Chúng ta phải tính riêng cho từng chiều:

- ① Chiều dài  $n$  cần bao nhiêu viên gạch cạnh  $a$ ?
- ② Chiều rộng  $m$  cần bao nhiêu viên gạch cạnh  $a$ ?

Gọi số gạch dọc là *doc*, số gạch ngang là *ngang*.

$$\text{Tổng số gạch} = doc \times ngang$$

#### Vấn đề phép chia

Nếu  $n = 6$ ,  $a = 4$ .  $6/4 = 1$  (dư 2). Vì không được cắt gạch, phần dư 2 mét này **bắt buộc** phải dùng thêm 1 viên nguyên nữa.  $\rightarrow$  Phải làm tròn lên (Ceiling).

## 4. Ẩn Dụ: Đi Xe Buýt

Hãy tưởng tượng bài toán Lát gạch giống hệt bài toán **Thuê Xe Buýt**:

- $n$  người cần đi, mỗi xe chở được  $a$  người.
- Hỏi cần thuê ít nhất bao nhiêu xe? (Không bỏ lại ai).

### Trường hợp 1: Chia hết

$$n = 10, a = 5.$$

- $10/5 = 2$  dư 0.
- $\rightarrow$  Cần đúng 2 xe.

### Trường hợp 2: Có dư

$$n = 13, a = 5.$$

- $13/5 = 2$  dư 3.
- 3 người này cần thêm 1 xe nữa.
- $\rightarrow$  Cần  $2 + 1 = 3$  xe.

## 5. Tư Duy Lập Trình: If-Else

Với máy tính, ta biểu diễn logic "Đi Xe Buýt" như sau:

```
1 long long soGach;  
2  
3 if (n % a == 0) {  
4     // Trường hợp chia hết: Vừa đủ  
5     soGach = n / a;  
6 } else {  
7     // Trường hợp có dư: Phải thêm 1 viêñ  
8     soGach = n / a + 1;  
9 }
```

→ Cách này **dễ hiểu** và **an toàn** nhất cho người mới bắt đầu.

## 6. Cách Viết Tắt (Nâng Cao)

Dân "Chuyên Tin" thường gộp logic if-else trên thành 1 dòng duy nhất để code nhanh hơn:

Công thức Ceiling

$$\text{Kết quả} = (n + a - 1) / a$$

Tại sao nó đúng?

- Logic: "Đổ thêm  $a - 1$  đơn vị vào để làm tràn ly".
- Nếu chia hết: Cộng  $a - 1$  không đủ tràn.
- Nếu dư: Cộng  $a - 1$  sẽ đẩy thương số lên +1.

## 7. Cạm Bẫy Lớn Nhất: Tràn Số

- Đề bài cho  $n, m$  lên tới  $10^9$ .
- Nếu dùng int, tích  $n \times m$  có thể lên tới  $10^{18}$ .
- int chỉ chứa được  $\approx 2 \times 10^9$ .
- → Bị **Tràn số (Overflow)** → Kết quả sai hoặc âm.

### Giải pháp

Bắt buộc dùng long long trong C++.

## 8. Lời Giải Tham Khảo

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // 1. Dùng long long để tránh tràn số
6     long long n, m, a;
7     cin >> n >> m >> a;
8
9     // 2. Tính số gạch cho chiều Dọc (n)
10    long long doc;
11    if (n % a == 0) doc = n / a;
12    else doc = n / a + 1;
13
14    // 3. Tính số gạch cho chiều Ngang (m)
15    long long ngang;
16    if (m % a == 0) ngang = m / a;
17    else ngang = m / a + 1;
18
19    // 3. Kết quả là tích của hai chiều
20    cout << doc * ngang;
```