



# CSES - Dynamic Programming

[Info](#) [Statistics](#) [Rankings](#) [Participation](#) [Submissions](#) [Leave contest](#)

[🔒 Thầy Trần Phùng Bính - GV THPT Gia Định - TP.HCM](#)



## Dice Combinations | Kết hợp xúc xắc

Submit

**Time limit:** 1.0s / **Memory limit:** 512M

**Points:** 100

Nhiệm vụ của bạn là đếm số cách tạo ra tổng  $n$  bằng cách gieo xúc xắc một hoặc nhiều lần. Mỗi lần gieo cho ra số từ 1 đến 6.

Ví dụ, nếu  $n = 3$ , có 4 cách:

- $1 + 1 + 1$
- $1 + 2$
- $2 + 1$
- $3$

### Input

- Dòng đầu vào duy nhất có số nguyên  $n$ .

### Output

- In số cách chia lấy dư cho  $10^9 + 7$

### Giới hạn

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

### Sample Input

3

Copy



4

Copy

## ✓ Minimizing Coins | Giảm thiểu đồng xu

Submit

**Time limit:** 1.0s / **Memory limit:** 512M**Points:** 100

Hãy xét một hệ thống tiền bao gồm  $n$  đồng xu. Mỗi đồng xu có giá trị là một số nguyên dương. Nhiệm vụ của bạn là tạo ra một khoản tiền  $x$  bằng cách sử dụng các đồng xu có sẵn sao cho số lượng đồng xu là tối thiểu.

Ví dụ: nếu các đồng xu là  $\{1, 5, 7\}$  và tổng mong muốn là 11, một giải pháp tối ưu là  $5 + 5 + 1$ , cần 3 đồng xu.

### Input

- Dòng đầu vào đầu tiên có hai số nguyên  $n$  và  $x$ : số lượng đồng xu và tổng số tiền mong muốn.
- Dòng thứ hai có  $n$  số nguyên phân biệt  $c_1, c_2, \dots, c_n$ : giá trị của mỗi đồng xu.

### Output

- In một số nguyên: số lượng đồng xu tối thiểu. Nếu không thể tạo ra tổng mong muốn, hãy in  $-1$ .

### Constraints

- $1 \leq n \leq 100$
- $1 \leq x \leq 10^6$
- $1 \leq c_i \leq 10^6$

### Sample Input

```
3 11
1 5 7
```

Copy

### Sample Output



## ✓ Coin Combinations I | Kết hợp đồng xu I

[Submit](#)**Time limit:** 1.0s / **Memory limit:** 512M**Points:** 100

Hãy xem xét một hệ thống tiền bao gồm  $n$  đồng xu. Mỗi đồng xu có giá trị là một số nguyên dương. Nhiệm vụ của bạn là tính số lượng các cách khác nhau mà bạn có thể tạo ra một khoản tiền  $x$  bằng cách sử dụng các đồng xu có sẵn.

Ví dụ: nếu các đồng xu là  $\{2, 3, 5\}$  và tổng mong muốn là 9, có 8 cách:

- $2 + 2 + 5$
- $2 + 5 + 2$
- $5 + 2 + 2$
- $3 + 3 + 3$
- $2 + 2 + 2 + 3$
- $2 + 2 + 3 + 2$
- $2 + 3 + 2 + 2$
- $3 + 2 + 2 + 2$

### Input

- Dòng đầu vào đầu tiên có hai số nguyên  $n$  và  $x$ : số lượng đồng xu và tổng số tiền mong muốn.
- Dòng thứ hai có  $n$  số nguyên phân biệt  $c_1, c_2, \dots, c_n$ : giá trị của mỗi đồng xu.

### Output

- In một số nguyên: số lượng cách, chia lấy dư cho  $10^9 + 7$ .

### Constraints

- $1 \leq n \leq 100$
- $1 \leq x \leq 10^6$
- $1 \leq c_i \leq 10^6$

### Sample Input

```
3 9
2 3 5
```

[Copy](#)



8

Copy

## ✓ Coin Combinations II | Kết hợp đồng xu II

Submit

**Time limit:** 1.0s / **Memory limit:** 512M**Points:** 100

Xét một hệ thống tiền tệ với  $n$  loại đồng xu. Mỗi đồng xu có giá trị là một số nguyên dương. Hãy tính số cách khác nhau, *không kể thứ tự* để tạo ra tổng tiền  $x$  từ những đồng xu này.

Ví dụ: nếu các đồng xu là  $\{2, 3, 5\}$  và tổng mong muốn là 9, có 3 cách:

- $2 + 2 + 5$
- $3 + 3 + 3$
- $2 + 2 + 2 + 3$

### Input

Định dạng đầu vào:

- Dòng đầu tiên chứa hai số nguyên  $n$  và  $x$ : số lượng đồng xu và tổng số tiền mong muốn.
- Dòng thứ hai chứa  $n$  số nguyên riêng biệt  $c_1, c_2, \dots, c_n$ : giá trị của mỗi đồng xu.

### Output

- In một số nguyên duy nhất: số lượng cách, chia lấy dư cho  $10^9 + 7$ .

### Constraints

- $1 \leq n \leq 100$
- $1 \leq x \leq 10^6$
- $1 \leq c_i \leq 10^6$

### Sample Input

```
3 9
2 3 5
```

Copy



3

Copy

## Removing Digits | Loại bỏ chữ số

Submit

**Time limit:** 1.0s / **Memory limit:** 512M**Points:** 100

Bạn được cho một số nguyên  $n$ . Ở mỗi bước, bạn có thể trừ  $n$  đi một lượng bằng một trong các chữ số của nó. Cần bao nhiêu bước để làm cho  $n$  bằng 0?

### Input

- Gồm một dòng duy nhất chứa số nguyên  $n$ .

### Output

- In ra một số nguyên duy nhất là số bước tối thiểu cần dùng.

### Constraints

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

## Sample Input

27

Copy

## Sample Output

5

Copy

## Giải thích

- Một giải pháp tối ưu là  $27 \rightarrow 20 \rightarrow 18 \rightarrow 10 \rightarrow 9 \rightarrow 0$



# Grid Paths | Đường đi trên lưới

[Submit](#)**Time limit:** 1.0s / **Memory limit:** 512M**Points:** 100

Xét một lưới ô vuông kích thước  $n \times n$ , trong đó một số ô có thể có bẫy. Ta không được phép đi qua một ô có bẫy.

Hãy tính số lượng đường đi từ góc trên trái đến góc dưới phải của lưới, biết rằng ta chỉ được đi sang phải hoặc đi xuống dưới.

## Input

- Dòng đầu tiên chứa một số nguyên  $n$ : kích thước của lưới.
- $n$  dòng sau, mỗi dòng chứa  $n$  kí tự mô tả lưới: ☐ biểu thị một ô trống và ☒ biểu thị một cái bẫy.

## Output

- In ra một số nguyên duy nhất là số lượng đường đi chia lấy dư cho  $10^9 + 7$ .

## Constraints

- $1 \leq n \leq 1000$

## Sample Input

4

....

.\*..

...\*

\*...

[Copy](#)

## Sample Output

3

[Copy](#)



**Time limit:** 1.0s / **Memory limit:** 512M

**Points:** 100

Bạn đang ở trong một hiệu sách bán  $n$  cuốn sách khác nhau. Bạn biết giá và số trang của mỗi cuốn sách.

Bạn quyết định tổng số tiền mua sách của bạn tối đa là  $x$ . Tổng số trang tối đa bạn có thể mua là bao nhiêu? Bạn chỉ có thể mua mỗi cuốn sách nhiều nhất một lần.

## Input

- Dòng đầu tiên chứa hai số nguyên  $n$  và  $x$ : số lượng sách và tổng số tiền tối đa.
- Dòng tiếp theo chứa  $n$  số nguyên  $h_1, h_2, \dots, h_n$ : giá cả của mỗi cuốn sách.
- Dòng cuối cùng chứa  $n$  số nguyên  $s_1, s_2, \dots, s_n$ : số trang của mỗi cuốn sách.

## Output

- In một số nguyên duy nhất: tổng số trang tối đa.

## Constraints

- $1 \leq n \leq 1000$
- $1 \leq x \leq 10^5$
- $1 \leq h_i, s_i \leq 1000$

## Sample Input

```
4 10
4 8 5 3
5 12 8 1
```

[Copy](#)

## Sample Output

```
13
```

[Copy](#)

## Giải thích

- Bạn có thể mua các cuốn sách 1 và 3. Giá của chúng là  $4 + 5 = 9$  và số lượng trang là  $5 + 8 = 13$ .



# Array Description | Mô tả mảng Submit

**Time limit:** 1.0s / **Memory limit:** 512M

**Points:** 100

Cho trước một mảng độ dài  $n$  trong đó có một số vị trí chưa được xác định giá trị. Hãy đếm số cách điền giá trị vào những vị trí đó thỏa mãn điều kiện sau:

- Các giá trị trong mảng là số nguyên trong khoảng từ 1 đến  $m$
- Chênh lệch giữa hai phần tử liên kề không quá 1

## Input

- Dòng đầu tiên có hai số nguyên  $n$  và  $m$ : kích thước mảng và giới hạn trên cho mỗi giá trị.
- Dòng tiếp theo có  $n$  số nguyên  $x_1, x_2, \dots, x_n$ : nội dung của mảng. Giá trị 0 biểu thị một giá trị không xác định.

## Output

- In một số nguyên: số lượng dãy (cũng là số lượng cách điền) chia lấy dư cho  $10^9 + 7$ .

## Constraints

- $1 \leq n \leq 10^5$
- $1 \leq m \leq 100$
- $0 \leq x_i \leq m$

## Sample Input

```
3 5
2 0 2
```

Copy

## Sample Output

```
3
```

Copy

## Giải thích

- Các dãy  $[2, 1, 2]$ ,  $[2, 2, 2]$ ,  $[2, 3, 2]$  khớp với mô tả.



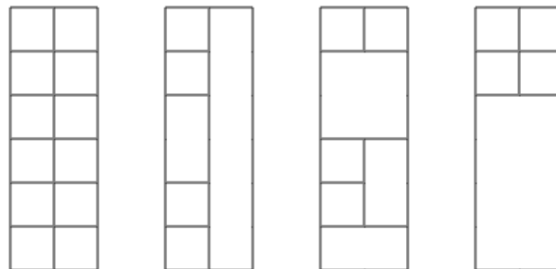


# Counting Towers | Đếm tháp

[Submit](#)**Time limit:** 1.0s / **Memory limit:** 512M**Points:** 100

Nhiệm vụ của bạn là xây dựng một tòa tháp có chiều rộng là 2 và chiều cao là  $n$ . Bạn có nguồn cung cấp không giới hạn các khối có chiều rộng và chiều cao là số nguyên.

Ví dụ: đây là một số giải pháp khả thi cho  $n = 6$ :



Cho trước  $n$ , hỏi bạn có thể xây được bao nhiêu tòa tháp khác nhau? Với hai tháp A, B *trông* khác nhau, nếu A đối xứng với B, hoặc sau khi *xoay* A, ta thu được tháp B, thì ta vẫn tính A và B là hai tháp *riêng biệt*.

## Input

- Dòng đầu tiên chứa một số nguyên  $t$ : số lượng test.
- $t$  dòng tiếp theo, mỗi dòng chứa một số nguyên  $n$ : chiều cao của tháp.

## Output

- Đối với mỗi test, in số lượng tòa tháp sau khi chia lấy dư cho  $10^9 + 7$  *trên một dòng riêng*.

## Constraints

- $1 \leq t \leq 100$
- $1 \leq n \leq 10^6$

## Sample Input

```
3
2
6
1337
```

[Copy](#)

## Sample Output



2864

640403945

# Edit Distance | Khoảng cách chỉnh sửa

[Submit](#)**Time limit:** 1.0s / **Memory limit:** 512M**Points:** 100

*Khoảng cách chỉnh sửa* giữa hai xâu là số lượng thao tác tối thiểu cần thiết để chuyển đổi một xâu thành xâu kia.

Các thao tác được phép là:

- Thêm một ký tự vào xâu.
- Xóa một ký tự khỏi xâu.
- Thay thế một ký tự trong xâu.

Ví dụ: khoảng cách chỉnh sửa giữa `LOVE` và `MOVIE` là 2, vì trước tiên bạn có thể thay thế `L` bằng `M`, sau đó thêm `I`.

Nhiệm vụ của bạn là tính toán khoảng cách chỉnh sửa giữa hai xâu.

## Input

- Dòng đầu tiên có một xâu chứa  $n$  ký tự trong khoảng từ `A` - `Z`.
- Dòng thứ hai có một xâu chứa các ký tự  $m$  trong khoảng từ `A` - `Z`.

## Output

- In một số nguyên: khoảng cách chỉnh sửa giữa các xâu.

## Constraints

- $1 \leq n \leq 5000$

## Sample Input

```
LOVE
MOVIE
```

[Copy](#)



2

Copy

## Rectangle Cutting | Cắt hình chữ nhật

Submit

**Time limit:** 1.0s / **Memory limit:** 512M**Points:** 100

Với một hình chữ nhật  $a \times b$ , nhiệm vụ của bạn là cắt nó thành các hình vuông. Trong mỗi bước, bạn có thể chọn một hình chữ nhật và cắt nó thành hai hình chữ nhật sao cho độ dài các cạnh vẫn là số nguyên. Số bước tối thiểu là bao nhiêu?

### Input

- Gồm một dòng duy nhất chứa hai số nguyên  $a$  và  $b$ .

### Output

- In một số nguyên: số lần di chuyển tối thiểu.

### Constraints

- $1 \leq a, b \leq 500$

### Sample Input

3 5

Copy

### Sample Output

3

Copy

## Money Sums | Khoản tiền

Submit

**Points:** 100

Bạn có  $n$  đồng xu với các giá trị nhất định. Nhiệm vụ của bạn là tìm tất cả các khoản tiền bạn có thể tạo bằng những đồng xu này.

## Input

- Dòng đầu vào đầu tiên có một số nguyên  $n$ : số lượng đồng xu.
- Dòng tiếp theo có  $n$  số nguyên  $x_1, x_2, \dots, x_n$ : giá trị của các đồng xu.

## Output

- Dòng đầu tiên in ra một số nguyên  $k$ : số lượng khoản tiền khác nhau có thể tạo
- Dòng tiếp theo in ra  $k$  số nguyên: các khoản tiền có thể tạo được, theo thứ tự tăng dần

## Constraints

- $1 \leq n \leq 100$
- $1 \leq x_i \leq 1000$

## Sample Input

```
4
4 2 5 2
```

[Copy](#)

## Sample Output

```
9
2 4 5 6 7 8 9 11 13
```

[Copy](#)

# Removal Game | Trò chơi loại bỏ

[Submit](#)**Time limit:** 1.0s / **Memory limit:** 512M**Points:** 100



đó. Cả hai người chơi đều muốn tối đa hóa điểm của họ.

Số điểm tối đa có thể của người chơi thứ nhất là bao nhiêu nếu cả hai người đều chơi tối ưu?

## Input

- Dòng đầu vào đầu tiên chứa một số nguyên  $n$ : kích thước của dãy.
- Dòng tiếp theo có  $n$  số nguyên  $x_1, x_2, \dots, x_n$ : các phần tử của dãy.

## Output

- In số điểm tối đa có thể của người chơi thứ nhất.

## Constraints

- $1 \leq n \leq 5000$
- $-10^9 \leq x_i \leq 10^9$

## Sample Input

```
4
4 5 1 3
```

[Copy](#)

## Sample Output

```
8
```

[Copy](#)

## Increasing Subsequence | Dãy con tăng

[Submit](#)

**Time limit:** 1.0s / **Memory limit:** 512M

**Points:** 100



Một dãy con là một dãy có thể thu được từ mảng bằng cách xóa một số phần tử mà vẫn không thay đổi thứ tự của các phần tử còn lại.

### Input

- Dòng đầu tiên chứa một số nguyên  $n$  ( $1 \leq n \leq 2 \times 10^5$ ) - kích thước của mảng.
- Sau đó có  $n$  số nguyên  $x_1, x_2, \dots, x_n$  ( $1 \leq x_i \leq 10^9$ ) - các phần tử của mảng.

### Output

- In độ dài của dãy con tăng dài nhất.

## Sample Input

```
8
7 3 5 3 6 2 9 8
```

[Copy](#)

## Sample Output

```
4
```

[Copy](#)

## Two Sets II | Hai tập hợp II

[Submit](#)

**Time limit:** 1.0s / **Memory limit:** 512M

**Points:** 100



Ví dụ, với  $n = 7$ , có 4 cách chia:

- $\{1, 3, 4, 6\}$  và  $\{2, 5, 7\}$
- $\{1, 2, 5, 6\}$  và  $\{3, 4, 7\}$
- $\{1, 2, 4, 7\}$  và  $\{3, 5, 6\}$
- $\{1, 6, 7\}$  và  $\{2, 3, 4, 5\}$

### Input

- Gồm một dòng duy nhất chứa số nguyên  $n$ .

### Output

- In đáp án - số cách thoả mãn chia lấy dư cho  $10^9 + 7$ .

### Constraints

- $1 \leq n \leq 500$

## Sample Input

7

Copy

## Sample Output

4

Copy

## Projects | Dự án Submit

**Time limit:** 1.0s / **Memory limit:** 512M

**Points:** 100



Số tiền tối đa mà bạn có thể kiếm được là bao nhiêu?

### Input

- Dòng đầu tiên chứa một số nguyên  $n$ : số lượng dự án.
- $n$  dòng tiếp theo, mỗi dòng chứa ba số nguyên  $a_i$ ,  $b_i$  và  $p_i$ : ngày bắt đầu, ngày kết thúc và phần thưởng.

### Output

- In một số nguyên: số tiền tối đa mà bạn có thể kiếm được.

### Constraints

- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq a_i \leq b_i \leq 10^9$
- $1 \leq p_i \leq 10^9$

### Sample Input

```
4
2 4 4
3 6 6
6 8 2
5 7 3
```

[Copy](#)

### Sample Output

```
7
```

[Copy](#)

## Elevator Rides | Đi thang máy

[Submit](#)

**Time limit:** 1.0s / **Memory limit:** 512M

**Points:** 100





nhiều?

## Input

- Dòng đầu tiên có hai số nguyên  $n$  và  $x$ : số lượng người và trọng lượng tối đa cho phép trong thang máy.
- Dòng thứ hai chứa  $n$  số nguyên  $w_1, w_2, \dots, w_n$ : trọng lượng của mỗi người.

## Output

- In một số nguyên: số lần đi tối thiểu.

## Constraints

- $1 \leq n \leq 20$
- $1 \leq x \leq 10^9$
- $1 \leq w_i \leq x$

## Sample Input

```
4 10
4 8 6 1
```

[Copy](#)

## Sample Output

```
2
```

[Copy](#)

# Counting Tilings | Đếm cách lát gạch

[Submit](#)

**Time limit:** 1.0s / **Memory limit:** 512M

**Points:** 100



## Input

- Dòng đầu vào duy nhất chứa hai số nguyên  $n$  và  $m$ .

## Output

- In một số nguyên: số lượng cách lát, chia lấy dư cho  $10^9 + 7$ .

## Constraints

- $1 \leq n \leq 10$
- $1 \leq m \leq 1000$

## Sample Input

4 7

Copy

## Sample Output

781

Copy

# Counting Numbers | Đếm số

Submit

**Time limit:** 1.0s / **Memory limit:** 512M

**Points:** 100



## Input

- Gồm một dòng duy nhất chứa hai số nguyên  $a$  và  $b$ .

## Output

- In một số nguyên: đáp án cho vấn đề.

## Constraints

- $0 \leq a \leq b \leq 10^{18}$

## Sample Input

```
123 321
```

[Copy](#)

## Sample Output

```
171
```

[Copy](#)

proudly powered by **DMOJ** | follow us on **Github** and **Facebook**