

## Bài 1: Số fibonacci lớn hơn k

Dãy fibonacci là dãy số rất cơ bản cho nhiều bài toán trong tự nhiên vào khoa học. Dãy thường được định nghĩa một cách truy hồi như sau:

- $F(0) = 0$
- $F(1) = 1$
- $F(n) = F(n-1) + F(n-2)$  với  $n > 1$

Cho một số nguyên K, nhiệm vụ của bạn là tìm số fibonacci nhỏ nhất lớn hơn K.

### Input

Số K nguyên.

### Output

Số fibonacci nhỏ nhất F mà  $F > K$ .

### Giới hạn

Thời gian: 1s

Bộ nhớ: 100 MB

Mã nguồn: 50 KB

### Ví dụ

Input	Output
1	2
10	13
100	144
-1000	0

## Bài 2: Số bộ 5 CSC

Cho dãy A gồm N số nguyên đôi một khác nhau. Đếm xem có bao nhiêu cách lấy ra từ A đúng 5 phần tử để 5 phần tử đó tạo thành một cấp số cộng.

### Input

Dòng 1: Số N ( $N < 200$ )

Dòng 2: N số nguyên của A

### Output

Số cách chọn 5 phần tử từ A.

### Giới hạn

Thời gian: 1s

Bộ nhớ: 100 MB

Mã nguồn: 50 KB

### Ví dụ

Input	Output
5 5 1 3 2 4	1
6 10 7 0 -2 1 4	1
10 9 7 5 3 1 2 4 6 8 0	8

### Bài 3: Chồng gạch xếp cao nhất

Có  $N$  viên gạch, được đánh số từ 1 đến  $N$ , các viên gạch có độ cứng lần lượt là  $a[1], a[2], \dots, a[N]$  là các số nguyên không âm. Một viên gạch có độ cứng  $K$  có nghĩa là có thể đặt chồng lên nó tối đa  $K$  viên gạch khác. Hỏi có thể xếp chồng gạch tối đa bao nhiêu viên.

#### Input

Số  $N$  ( $N < 10000$ ) và dãy  $N$  số nguyên không âm  $a[1] a[2] \dots a[N]$ .

#### Output

Chiều cao chồng gạch tối đa có thể sắp được.

#### Giới hạn

Thời gian: 1s

Bộ nhớ: 100 MB

Mã nguồn: 50 KB

#### Ví dụ

Input	Output
3 1 2 1	3
6 0 0 0 0 0 0	1

### Bài 4: Số fibonacci bậc 3

Hàm  $f(n)$  với tham số  $n$  là số tự nhiên được định nghĩa như sau:

- $f(0) = 0$
- $f(1) = 1$
- $f(2) = 2$
- $f(3k) = f(2k)$  với  $k > 0$
- $f(3k+1) = f(2k) + f(2k+1)$  với  $k > 0$
- $f(3k+2) = f(2k) + f(2k+1) + f(2k+2)$  với  $k > 0$

Nhiệm vụ: Nhập số  $n$  và in ra giá trị của  $f(n)$ .

Ràng buộc:

- Giá trị của  $n$  không quá 2 tỉ ( $2 \times 10^9$ )
- Chương trình chạy không quá 2 giây

Quan sát các ví dụ dưới đây để biết quy cách vào ra dữ liệu của chương trình.

```
Nhap n = 10
f(10) = 16
-----
Process exited after 1.322 seconds with return value 0
Press any key to continue . . .
```

```
Nhap n = 10000
f(10000) = 2711824
-----
Process exited after 1.731 seconds with return value 0
Press any key to continue . . .
```

```
Nhap n = 1234567
f(1234567) = 5451940640
-----
Process exited after 9.467 seconds with return value 0
Press any key to continue . . .
```

## Bài 5: Hàm $g(n)$ chia 4

Hàm  $g(n)$  với tham số  $n$  nguyên dương được định nghĩa như sau:

- $g(1) = 1$
- $g(3) = 3$
- $g(2n) = g(n)$
- $g(4n+1) = 2g(2n+1) - g(n)$
- $g(4n+3) = 3g(2n+1) - 2g(n)$

Nhiệm vụ: Nhập số  $n$  và in ra giá trị của  $g(n)$ .

Ràng buộc:

- Giá trị của  $n$  không quá 2 tỉ ( $2 \times 10^9$ )
- Chương trình chạy không quá 2 giây

Quan sát các ví dụ dưới đây để biết quy cách vào ra dữ liệu của chương trình.

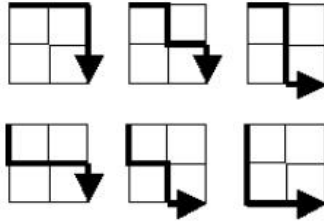
```
Nhap n = 10
g(10) = 5
-----
Process exited after 1.424 seconds with return value 0
Press any key to continue . . .
```

```
Nhap n = 71263
g(71263) = 128209
-----
Process exited after 2.754 seconds with return value 0
Press any key to continue . . .
```

```
Nhap n = 989898989
g(989898989) = 767901751
-----
Process exited after 8.306 seconds with return value 0
Press any key to continue . . .
```

## Bài 6: Đường đi trên lưới

Một lưới vuông cỡ  $2 \times 2$ , nếu đứng ở góc trái-trên  $(0, 0)$  và di chuyển theo các cạnh ô vuông để xuống góc phải-dưới  $(2, 2)$ , ta có tất cả 6 cách như hình dưới.



Nhiệm vụ: Nếu cỡ lưới ô là  $M \times N$ , thì sẽ có bao nhiêu cách di chuyển? Nhập giá trị  $M$  và  $N$  từ bàn phím sau đó in số cách di chuyển ra màn hình.

Ràng buộc:

- Giá trị của  $M$  và  $N$  không quá 50
- Chương trình chạy không quá 2 giây

Quan sát các ví dụ dưới đây để biết quy cách vào ra dữ liệu của chương trình.

```
Nhap M = 2
Nhap N = 3
So cach = 10
-----
Process exited after 2.384 seconds with return value 0
Press any key to continue . . .
```

```
Nhap M = 3
Nhap N = 3
So cach = 20
-----
Process exited after 3.491 seconds with return value 0
Press any key to continue . . .
```

## Bài 7: Phân tích N thành tổng M số

Nhập hai số nguyên dương N và M, hãy in ra màn hình tất cả các cách phân tích số N thành tổng của đúng M số nguyên dương.

Quan sát các ví dụ dưới đây để biết quy cách vào ra dữ liệu của chương trình.

```
Nhap n = 4
Nhap m = 3
4 = 1+1+2
4 = 1+2+1
4 = 2+1+1

-----
Process exited after 1.115 seconds with return value 0
Press any key to continue . . .
```

```
Nhap n = 4
Nhap m = 5

-----
Process exited after 8.617 seconds with return value 0
Press any key to continue . . .
```

```
Nhap n = 5
Nhap m = 3
5 = 1+1+3
5 = 1+2+2
5 = 1+3+1
5 = 2+1+2
5 = 2+2+1
5 = 3+1+1

-----
Process exited after 8.324 seconds with return value 0
Press any key to continue . . .
```

## Bài 8: Phân tích N thành tổng các số dương

Nhập hai số nguyên dương N, hãy in ra màn hình tất cả các cách phân tích số N thành tổng của các số nguyên dương (không giới hạn số lượng).

Chú ý:

- Mỗi cách phân tích không tính các hoán vị của chúng
- In các cách phân tích theo thứ tự từ điển ngược (xem ví dụ)

Quan sát các ví dụ dưới đây để biết quy cách vào ra dữ liệu của chương trình.

```
Nhap n = 3
3 = 3
3 = 2+1
3 = 1+1+1

-----
Process exited after 2.531 seconds with return value 0
Press any key to continue . . .
```

```
Nhap n = 5
5 = 5
5 = 4+1
5 = 3+2
5 = 3+1+1
5 = 2+2+1
5 = 2+1+1+1
5 = 1+1+1+1+1

-----
Process exited after 0.8999 seconds with return value 0
Press any key to continue . . .
```

## Bài 9: Số dãy con có tổng bằng M

Cho dãy số A có N số nguyên, nhập số M nguyên dương, hãy đếm xem có bao nhiêu dãy con của A có tổng đúng bằng M.

Ràng buộc:

- Chương trình chạy không quá 2 giây
- Chỉ cần chỉ ra số lượng dãy con của A tổng bằng M
- Dãy con có thể gồm các phần tử không liên tiếp

Quan sát các ví dụ dưới đây để biết quy cách vào ra dữ liệu của chương trình.

```
Nhap m = 5
Nhap n = 4
a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 4
Co tat ca 2 cach phan tich.
-----
Process exited after 3.276 seconds with return value 0
Press any key to continue . . .
```

```
Nhap m = 7
Nhap n = 6
a[0] = 1
a[1] = 1
a[2] = 2
a[3] = 2
a[4] = 3
a[5] = 4
Co tat ca 8 cach phan tich.
-----
Process exited after 9.532 seconds with return value 0
Press any key to continue . . .
```

## Bài 10: Dãy con chung tổng lớn nhất

Cho hai dãy số nguyên dương  $A = (a_1, a_2, \dots, a_m)$  và  $B = (b_1, b_2, \dots, b_n)$ . Hãy tìm dãy con chung của A và B có tổng giá trị phần tử lớn nhất.

Dữ liệu nhập vào là các số nguyên dương theo thứ tự như sau: số m và m phần tử của A, số n và n phần tử của B.

Dữ liệu in ra: tổng lớn nhất tìm được.

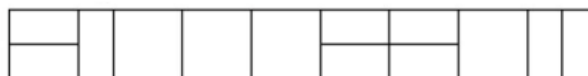
Quan sát các ví dụ dưới đây để biết quy cách vào ra dữ liệu của chương trình.

```
Nhap m = 4
a[1] = 3
a[2] = 4
a[3] = 1
a[4] = 1
Nhap n = 5
b[1] = 3
b[2] = 1
b[3] = 1
b[4] = 2
b[5] = 4
Day con co tong lon nhat = 7
-----
```

## Bài 11: Lát gạch đường đôi

Ở công viên thành phố có một bức tường, trên bức tường người ta gắn một bức phù điêu tái hiện sự kiện lịch sử của thành phố.

Ở dưới chân bức tường người ta dự định ốp gạch trang trí trên khoảng hình chữ nhật có kích thước  $2 \times N$ , có 2 loại gạch  $1 \times 2$  và  $2 \times 2$ . Hình dưới đây là một ví dụ về cách ốp gạch với  $N = 17$ .



Yêu cầu: Nhập giá trị của N và tính số lượng cách ốp gạch khác nhau cho hình  $2 \times N$ .

Quan sát các ví dụ dưới đây để biết quy cách vào ra dữ liệu của chương trình.

```
N = 4
So phuong an = 11
-----
Process exited after 1.946 seconds with return value 0
Press any key to continue . . .

N = 12
So phuong an = 2731
-----
Process exited after 1.278 seconds with return value 0
Press any key to continue . . .
```



## Bài 12: Phân tích K thành tổng các số dương khác nhau

Cho số nguyên dương K, hãy đếm xem có bao nhiêu cách phân tích K thành tổng ít nhất 2 số nguyên dương đôi một khác nhau.

*Chú ý: hai cách phân tích là hoán vị của nhau chỉ được tính là một, chẳng hạn  $3 = 1 + 2$  và  $3 = 2 + 1$  được xem là một cách.*

Quan sát các ví dụ dưới đây để biết quy cách vào ra dữ liệu của chương trình.

```
Nhap k = 2
Co tat ca 0 cach phan tich.
-----
Process exited after 1.36 seconds with return value 0
Press any key to continue . . .
```

```
Nhap k = 3
Co tat ca 1 cach phan tich.
-----
Process exited after 2.749 seconds with return value 0
Press any key to continue . . .
```

```
Nhap k = 7
Co tat ca 4 cach phan tich.
-----
Process exited after 2.294 seconds with return value 0
Press any key to continue . . .
```

## Bài 13: Tìm phần tử trên chuỗi xoắn

### CHUOIXOAN – CHUỖI XOẮN VÔ TẬN

Cho trước một chuỗi  $W$ , ta gọi phép “Xoắn” trên  $W$  là phép gấp đôi độ dài của  $W$  bằng cách đảo ngược nội dung chuỗi  $W$  rồi chuỗi đó vào tiếp ngay sau  $W$  (như vậy chuỗi mới sẽ có dạng  $WW^{-1}$ , trong đó  $W^{-1}$  là nghịch đảo của  $W$ ).

Bằng cách áp dụng vô tận các phép Xoắn trên một xâu  $W$  ban đầu, ta có thể tạo được một chuỗi  $Z$  có độ dài vô tận có nội dung phát sinh từ  $W$ .

Vấn đề: Hãy xác định giá trị của kí tự thứ  $N$  trong chuỗi  $Z$ , biết chuỗi  $W$  ban đầu. Thứ tự được đánh số từ 0 theo cách làm thông thường của C/C++.

Nhập dữ liệu từ bàn phím:

- Dòng 1: chuỗi  $W$ .
- Dòng 2: số  $M$  là số lượng các vị trí cần tìm giá trị.
- $M$  dòng tiếp theo: lần lượt ghi các số  $N$ ,  $0 < N < 10^{12}$ .

In dữ liệu ra màn hình:

Mỗi dòng ghi một ký tự tương ứng với các vị trí  $N$  cần tìm.

Ví dụ:

Input	Output
abcd	d
5	c
4	b
5	a
6	a
7	
8	

Giải thích:

- Dãy  $Z$  được lập từ  $abcd$  sẽ là  $abcdcdcbabcdcdcbab...$
- Cần xác định vị trí của 5 phần tử
- Phần tử ở vị trí 4 là  $d$
- Phần tử ở vị trí 5 là  $c$
- Phần tử ở vị trí 6 là  $b$
- Phần tử ở vị trí 7 là  $a$
- Phần tử ở vị trí 8 là  $a$

## Bài 14: Tìm cách đi lợi ích nhất cho robot

### PHANTHUONG – Phần thưởng tối đa của robot

Một bản đồ trò chơi gồm có  $N$  điểm dừng:

- Các điểm dừng được đánh số từ 1 đến  $N$ .
- Trên mỗi điểm dừng đặt một món phần thưởng có giá trị  $10000 > V_k > 0$  ứng với số thứ tự  $k$  của điểm dừng đó.
- Giữa hai điểm dừng thứ  $p$  và thứ  $q$ , nếu có đường đi giữa chúng thì việc di chuyển sẽ mất thời gian là  $T_{p,q}$  ( $10000 > T_{p,q} = T_{q,p} > 0$ ).

Một robot xuất phát ở điểm dừng số 1 có nhiệm vụ di chuyển đến điểm dừng số  $N$ , nếu robot đi qua điểm dừng nào đó, nó có thể nhận được món phần thưởng ở điểm dừng đó.

Nhiệm vụ: Nhập thông tin về bản đồ trò chơi, hãy tính xem robot di chuyển đến điểm dừng số  $N$  nhanh nhất có thể nhận được tổng giá trị thưởng tối đa là bao nhiêu?

Nhập dữ liệu từ bàn phím:

- Dòng 1: số  $N$  ( $N \leq 1000$ )
- Dòng 2: các giá trị  $V_1 V_2 \dots V_N$
- Dòng 3: số  $M$  (số lượng đường đi trên bản đồ)
- $M$  dòng tiếp theo: mỗi dòng ghi 3 số  $p, q$  và  $u$  trong đó  $u$  là thời gian di chuyển từ điểm  $p$  đến  $q$ .

In kết quả ra màn hình:

Duy nhất một con số là giá trị tối đa của robot di chuyển đến  $N$  nhanh nhất, ghi 0 nếu không có đường đi.

Ví dụ

Input	Output
8 1 2 3 4 7 7 5 8 11 1 2 1 1 3 1 1 4 1 2 5 1 2 6 1 3 6 1 3 7 1 4 7 1	19