

عدد کامل

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

عدد کامل: هر عدد صحیح مثبتی که مجموع مقسوم علیه‌های سره آن برابر با خود آن عدد شود را عدد کامل می‌نامیم. مجموعه مقسوم علیه‌های سره یک عدد مجموعه ای شامل تمام مقسوم علیه‌های آن عدد به جز خود آن عدد می‌باشد.

حال به کمک توابع پایتون، برنامه ای بنویسید که عددی را به عنوان ورودی گرفته و کامل بودن یا نبودن آن را تعیین نماید.

ورودی

ورودی تنها شامل یک خط است که در آن یک عدد طبیعی n آمده است.

خروجی

خروجی برنامه‌ی شما باید شامل یک خط باشد و در صورتی که عدد داده شده در ورودی، کامل باشد True و در غیر این صورت False را چاپ نماید.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده می‌شود.

ورودی نمونه ۱

6

خروجی نمونه ۱

True

از آنجایی که مقسوم علیه‌های سره عدد ۶، اعداد ۱، ۲ و ۳ می‌باشند و مجموع این اعداد برابر ۶ است، پس این عدد کامل می‌باشد.

ورودی نمونه ۲

10

خروجی نمونه ۲

False

مجموعه مقسوم علیه‌های سره عدد ۱۰، اعداد ۱، ۲ و ۵ است که مجموع آن‌ها برابر ۸ می‌باشد که با خود عدد داده شده برابر نیست.

دنباله فیبوناچی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

تابع فیبوناچی، تابعی معروف است که نمایش بازگشتی آن به این صورت است که هر جمله‌ی آن با توجه به دو جمله قبلی‌اش محاسبه می‌شود.

$$fib(1) = fib(2) = 1$$

$$fib(n) = fib(n - 1) + fib(n - 2)$$

جملات دنباله فیبوناچی به شکل زیر می‌باشند:

1, 1, 2, 3, 5, 8, 13, 21, ...

حال، به کمک توابع بازگشتی، برنامه‌ای بنویسید که با گرفتن عدد طبیعی n ، عدد فیبوناچی n -ام را در خروجی چاپ نماید.

ورودی

ورودی تنها شامل یک خط است که در آن عدد طبیعی n درج شده است.

$$1 \leq n \leq 100$$

خروجی

خروجی برنامه‌ی شما شامل یک خط بوده که در آن عدد فیبوناچی n -ام چاپ شده است.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده می‌شود.

ورودی نمونه ۱

7

خروجی نمونه ۱

13

برج هانوی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

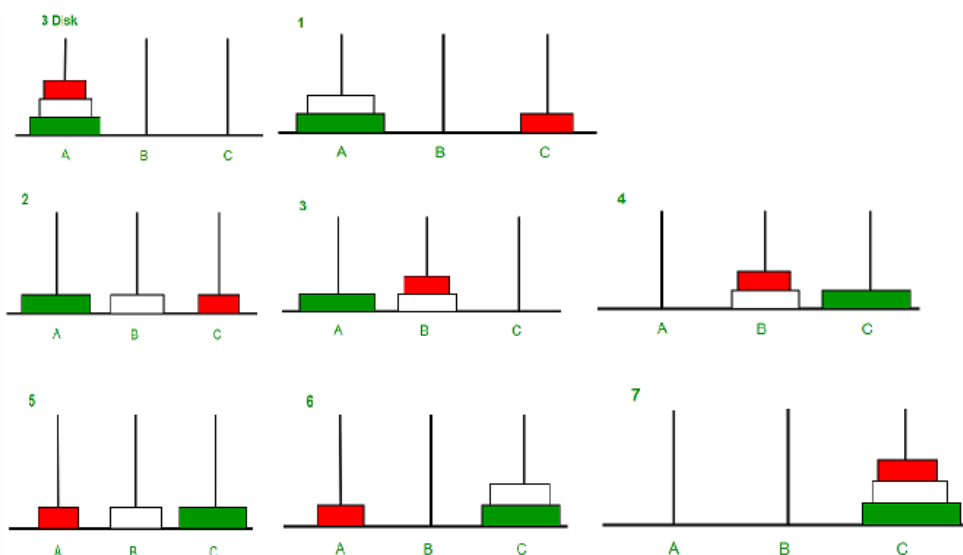
برج هانوی یک مسئله ریاضیاتی است که از سه میله و دیسک با بزرگی های مختلف تشکیل شده است. هدف این مسئله آن است که دیسک‌ها را با حفظ ترتیب قرارگیری فعلی آن‌ها روی یک دیگر، از میله ای به میله ای دیگر منتقل کنیم به شرطی که:

(۱) در هر بار حرکت یک و فقط یک دیسک را می‌توانیم جا به جا کنیم.

(۲) هر حرکت تنها شامل برداشتن بالاترین دیسک از یک میله و گذاشتن آن در میله ای دیگر می‌باشد. به عبارتی دیگر، تنها دیسک‌هایی را می‌توان جا به جا نمود که در بالاترین قسمت یکی از میله‌ها قرار دارند.

(۳) هیچ کدام از دیسک‌ها نمی‌توانند در بالای دیسکی که از آن کوچکتر است، قرار بگیرند.

به مراحل زیر که برای حل این مسئله با ۳ دیسک ایجاد شده است، توجه نمایید.



توجه: به نام گذاری ستون‌ها در تصویر فوق نیز دقت فرمایید.

حال به کمک توابع بازگشتی، برنامه‌ای بنویسید که قادر به حل کردن مسئله فوق باشد.

ورودی

ورودی تنها شامل یک خط است که در آن عدد طبیعی n که به منزله تعداد دیسک‌ها می‌باشد، به برنامه داده می‌شود.

$$1 \leq n \leq 20$$

خروجی

خروجی برنامه‌ی شما باید شامل چندین خط باشد که در هر خط آن، دستور حرکت آن مرحله از بازی بایستی چاپ شود. به عبارتی دیگر در هر خط، حرکت مناسب در هر مرحله از بازی چاپ می‌شود.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده می‌شود.

ورودی نمونه ۱

3

خروجی نمونه ۱

```
move disk 1 from A to B
move disk 2 from A to C
move disk 1 from B to C
move disk 3 from A to B
move disk 1 from C to A
move disk 2 from C to B
move disk 1 from A to B
```

ورودی نمونه ۲

2

خروجی نمونه ۲

```
move disk 1 from A to C
move disk 2 from A to B
move disk 1 from C to B
```