

فاکتوریل

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

برنامه‌ای بنویسید که عدد n را از ورودی گرفته و فاکتوریل آن را محاسبه کرده و نمایش دهد.

منظور از فاکتوریل یک عدد مانند n که با نماد $n!$ نشان داده می‌شود؛ ضرب اعداد 1 تا n است. به عبارت دیگر:

$$n! = 1 \times 2 \times 3 \times \dots \times n$$

ورودی

در یک خط عدد n به شما داده می‌شود.

$$1 \leq n \leq 10$$

خروجی

مقدار n فاکتوریل را در خروجی چاپ کنید.

مثال

ورودی نمونه ۱

5

خروجی نمونه ۱

120

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$

یافتن عدد اول

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

برنامه‌ای بنویسید که عدد صحیح n را از ورودی بخواند و مجموع ارقام آن را در b قرار دهد. خروجی b امین عدد اول پس از n است.

ورودی

در یک خط عدد n به شما داده می‌شود.

$$1 \leq n \leq 10^5$$

خروجی

در تنها خط خروجی عدد مطلوب را چاپ کنید.

مثال

ورودی نمونه ۱

100

خروجی نمونه ۱

101

ورودی نمونه ۲

55

خروجی نمونه ۲

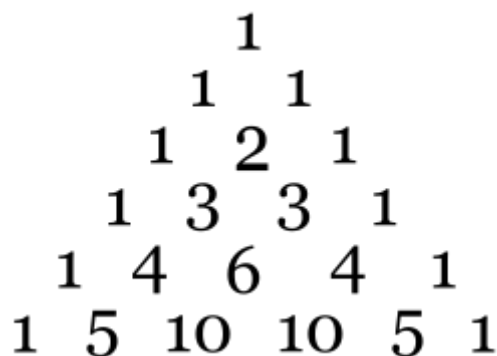
101

مثلث خیام پاسکال

- محدودیت زمان: ۰.۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

برنامه‌ای بنویسید که عدد n را از ورودی بگیرد و n سطر اول مثلث خیام پاسکال را نمایش دهد.

مثلث خیام پاسکال به این صورت است که ابتدا تنها یک عدد ۱ در سطر اول وجود دارد. سپس در سطر i ، عدد وجود دارد که عدد اول و آخر آن ۱، و هر کدام از اعداد دیگر جمع دو عدد بالایی خود می‌باشند. تصویر زیر، شش سطر اول مثلث خیام پاسکال می‌باشد.



ورودی

در تنها سطر ورودی عدد n می‌آید که نمایانگر تعداد سطرهایی است که باید چاپ شود.

$$1 \leq n \leq 100$$

خروجی

خروجی شامل n سطر است و باید در آن n سطر اول مثلث خیام پاسکال را خروجی دهید. فاصله‌ی بین هر دو عدد که در یک سطر قرار دارند باید یک فاصله باشد.

مثال

ورودی نمونه ۱

5

خروجی نمونه ۱

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

لوزی‌های ستاره‌ای

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

برنامه‌ای بنویسید که عدد n را از ورودی گرفته و دو لوزی به قطر n را در کنار هم با استفاده از کاراکتر * (مطابق خروجی نمونه) چاپ کند.

ورودی

در یک خط عدد فرد n به شما داده می‌شود.

$$1 \leq n \leq 19$$

خروجی

لوزی‌های کنار هم را در خروجی چاپ کنید.

مثال

ورودی نمونه

5

خروجی نمونه

```

*      *
***    ***
*****

```

*** **
* *