

به نام خدا



دوره جامع DEEPCAMP

---

تمرین سری اول

---

مهلت ارسال: ۱۴۰۴/۰۶/۳۱

## تمرین‌ها

۱. برنامه‌ای بنویسید که عددی را از ورودی بگیرد و بررسی کند که آیا بر ۳ و ۹ بخش پذیر است یا خیر. اگر عدد بر ۳ و ۹ بخش پذیر باشد چاپ کند هم بر ۹ و هم بر ۳ بخش پذیر است. اگر بر ۳ بخش پذیر باشد چاپ کند فقط بر ۳ بخش پذیر است. در غیر این صورت چاپ کند بر هیچ کدام بخش پذیر نیست. این سوال باید با تابع پیاده‌سازی شود.

### ورودی

ورودی یک عدد صحیح و غیرمنفی است.

### خروجی

خروجی یک خط است که یکی از سه پیام بالا را با توجه به شرایط عدد نمایش می‌دهد.

### مثال‌ها

• ورودی نمونه ۱

9

خروجی نمونه ۱

9 is dividable by both 9 and 3

• ورودی نمونه ۲

2

خروجی نمونه ۲

2 is neither dividable by 9 nor 3

• ورودی نمونه ۳

6

خروجی نمونه ۳

6 is dividable by only 3

۲. برنامه‌ای بنویسید که عددی را به عنوان ورودی بگیرد (ممکن است عدد اعشاری باشد) و آن را به عنوان شعاع یک دایره در نظر بگیرد. سپس دو تابع برای محاسبه مساحت و محیط دایره پیاده‌سازی کنید. اگر شعاع وارد شده عدد صفر یا منفی باشد، باید همان ابتدا Out of range را چاپ کند و در غیر این صورت به محاسبات گفته شده بپردازد. همچنین بعد از انجام محاسبات، باید در صورت کمتر بودن مساحت دایره از ۱۰ متن Small Circle و در صورت بزرگ‌تر یا مساوی بودن مساحت با ۱۰ متن Big Circle را در خط بعدی چاپ کند.

#### ورودی

یک عدد (اعشاری یا صحیح) به عنوان شعاع دایره.

#### خروجی

• اگر شعاع غیرمنفی و غیرصفر باشد:

— خط اول: رشته‌ای با فرمت area: {مساحت دایره}، perimeter: {محیط دایره}

که مساحت و محیط دایره با دقت ۴ رقم اعشار (۰.۰۰۰۱) چاپ شود.

— خط دوم: Big Circle (اگر مساحت  $\leq 10$ ) یا Small Circle (اگر مساحت  $> 10$ ).

• در غیر این صورت: تنها Out of range چاپ شود.

#### مثال‌ها

• ورودی نمونه ۱:

10

خروجی نمونه ۱:

area: 314.1593, perimeter: 62.8319

Big Circle

● ورودی نمونه ۲:

1.1

خروجی نمونه ۲:

area: 3.8013, perimeter: 6.9115

Small Circle

● ورودی نمونه ۳:

-10

خروجی نمونه ۳:

Out of range

۳. در این سؤال باید برنامه‌ای بنویسید که یک لیست از اعداد را از ورودی گرفته و آن را به صورت صعودی مرتب کند. این مرتب‌سازی باید به شکلی انجام شود که در هر مرحله یک زیرلیست از لیست اصلی انتخاب شده و این زیرلیست مرتب شود. طول زیرلیست در هر مرحله باید افزایش یابد تا زمانی که کل لیست مرتب شود. همچنین، ورودی‌های نامعتبر (مثل لیست خالی یا غیر عددی) باید با try-except مدیریت شوند.

ورودی

ورودی یک لیست از اعداد (اعداد صحیح یا اعشاری) است که باید به صورت صعودی مرتب شوند.

خروجی

خروجی باید زیرلیست‌های مرتب‌شده را در هر مرحله، مطابق با مثال‌های ارائه شده، نمایش دهد.

## مثال‌ها

### • ورودی نمونه ۱:

[5, 4, 3, 2, 1, 6]

### خروجی نمونه ۱

[5]

[4, 5]

[3, 4, 5]

[2, 3, 4, 5]

[1, 2, 3, 4, 5]

[1, 2, 3, 4, 5, 6]

### • ورودی نمونه ۲

[5, -4, 30, 23, -10, 6, 34, 99, -21]

### خروجی نمونه ۲

[5]

[-4, 5]

[-4, 5, 30]

[-4, 5, 23, 30]

[-10, -4, 5, 23, 30]

[-10, -4, 5, 6, 23, 30]

[-10, -4, 5, 6, 23, 30, 34]

[-10, -4, 5, 6, 23, 30, 34, 99]

[-21, -10, -4, 5, 6, 23, 30, 34, 99]

۴. در دنیای سیستم‌های چندرسانه‌ای، روش‌های مختلفی برای نمایش یک رنگ وجود دارد که یکی از معروف‌ترین آن‌ها مدل RGB24 است. این مدل هر رنگ را با سه عدد R (قرمز)، G (سبز)، و B (آبی) در بازه ۰ تا ۲۵۵ نمایش می‌دهد. فرض کنید یک دیکشنری به نام colors

وجود دارد که نام هر یک از رنگ‌های استاندارد را به اعداد مربوط به آن رنگ در مدل RGB24 به شکل زیر متصل کرده است:

```
colors = {  
'white': {'R': 255, 'G': 255, 'B': 255},  
'black': {'R': 0, 'G': 0, 'B': 0},  
'red': {'R': 255, 'G': 0, 'B': 0},  
'green': {'R': 0, 'G': 255, 'B': 0},  
'blue': {'R': 0, 'G': 0, 'B': 255},  
'yellow': {'R': 255, 'G': 255, 'B': 0},  
'gray': {'R': 160, 'G': 160, 'B': 160},  
'purple': {'R': 255, 'G': 0, 'B': 255},  
'light green': {'R': 128, 'G': 255, 'B': 0},  
}
```

دقت کنید که باید دیکشنری بالا را در کد خود کپی کنید.

برنامه‌ای بنویسید که یک دیکشنری به شکل `{ "R": 234, "G": 22, "B": 27 }` را از کاربر دریافت کرده و نام نزدیک‌ترین رنگ استاندارد به رنگ واردشده را با در نظر گرفتن معیار تفاوت زیر نمایش دهد:

$$D = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$$

در صورتی که کمترین تفاوت رنگ ورودی با رنگ‌های استاندارد یکتا نباشد، برنامه باید None برگرداند. به علاوه ورودی‌های نامعتبر (مثل مقادیر خارج از بازه ۰ تا ۲۵۵ یا فرمت نادرست دیکشنری) باید با try-except مدیریت شوند و خروجی None را برگردانند.

## ورودی

ورودی یک رشته است که به شکل دیکشنری است و باید با استفاده از `eval()` یا `json.loads()` به دیکشنری تبدیل شود. به مثال‌های ورودی توجه کنید.

## خروجی

خروجی یک رشته است که نام رنگی است که اعداد ورودی RGB به آن نزدیک است. اگر رنگ غیر یکتا یا ورودی نامعتبر باشد، `None` چاپ شود.

## مثال‌ها

### • ورودی نمونه ۱

```
{"R": 234, "G": 22, "B": 27}
```

### خروجی نمونه ۱

```
red
```

### • ورودی نمونه ۲

```
{"R": 0, "G": 0, "B": 0}
```

### خروجی نمونه ۲

```
black
```

۵. برنامه‌ای بنویسید که یک دیکشنری دریافت کند که مقادیر آن لیست‌هایی از اعداد هستند. شما باید کلیدی را پیدا کنید که لیست مربوط به آن بیشترین تعداد اعضای منحصر به فرد (unique) را داشته باشد. اگر چندین کلید با تعداد یکسان از اعضای منحصر به فرد وجود داشته باشند، کلیدی را برگردانید که در دیکشنری اول ظاهر شده است.

## ورودی

در تنها خط ورودی، یک دیکشنری داده می‌شود که کلیدهای آن رشته و مقادیر آن لیست‌هایی از اعداد هستند.

## خروجی

در تنها خط خروجی، کلیدی از دیکشنری چاپ کنید که لیست آن بیشترین تعداد اعضای منحصریه‌فرد را دارد.

## مثال‌ها

### • ورودی نمونه ۱

{'Gfg': [5, 7, 9, 4, 0], 'is': [6, 7, 4, 3, 3], 'Best': [9, 9, 6, 5, 5]}

### خروجی نمونه ۱

Gfg

### • ورودی نمونه ۲

{'Gfg': [5, 7, 7, 7, 7], 'is': [6, 7, 7, 7], 'Best': [9, 9, 6, 5, 5]}

### خروجی نمونه ۲

Best