

تمرین هفته دوم دوره مقدماتی پایتون

۱- تابعی بنویسید که دو عدد طبیعی را بعنوان ورودی بگیرد و مقدار کوچکترین مضرب مشترک (ک.م.م) آنها را محاسبه کند.

۲- با استفاده از دستور `while` تابعی به فرم `my_multiple(x,y)` بنویسید که عدد `x` را در `y` ضرب کند. دقت کنید که در این حالت فقط می‌توانید از عملکرد جمع استفاده کنید.

۳- با استفاده از حلقه `for` برنامه ای بنویسید که اعداد مضرب ۴ بین ۲۲ تا ۱۰۵ را چاپ کند.

۴- با استفاده از دو حلقه `for` تو در تو جدول ضرب اعداد ۱ تا ۱۰ را به صورت زیر چاپ کنید.

2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

۵- برنامه ای بنویسید که یک عدد صحیح بزرگتر از یک از کاربر بگیرد و خروجی آن یک مثلث قائم الزاویه به طول قائده های عدد ورودی باشد.

خروجی:	ورودی:
***** **** *** ** *	Enter your number: 5

۶- برنامه‌ای بنویسید که طول و ارتفاع یک متوازی الاضلاع را از کاربر دریافت می‌کند و آن متوازی الاضلاع را با کارکتر * و فاصله رسم کند.

خروجی:	ورودی:
<pre> ***** * * * * ***** </pre>	<pre> length: 5 height: 4 </pre>

۷- بازی حدس عدد: در این سوال قصد داریم بازی طراحی کنیم که یک عدد بین یک تا سی به صورت تصادفی (random) تولید شود، سپس از کاربر درخواست شود که این عدد را حدس بزند. کاربر تنها می‌تواند ۵ بار این عدد را حدس بزند. در صورتی که به درستی حدس زد به او پیغام **you win!** نمایش داده شود و در غیر اینصورت پیغام **gameOver**. برای تولید عدد تصادفی از ماژول **random** استفاده کنید.

۸- تابعی بنویسید که یک عدد حسابی و یک عدد طبیعی به عنوان طول را دریافت کند. خروجی این تابع باید رشته‌ای از اعداد باینری باشد که طول این باینری به اندازه آرگومان دوم و مقدار واقعی آن در واقع باینری شده عدد حسابی ورودی باشد. در صورتی که یک عدد را به باینری تبدیل کردید و طول آن از آرگومان دوم کمتر بود باید صفر به ابتدای عدد اضافه شود.

خروجی:	ورودی:
'۰۱۰۰'	F(8, 4)
'۰۰۰۰۰۰۰۰۱۱۱'	F(۷, ۱۱)

۹- از ایده سوال ۸ استفاده کنید و تابعی بنویسید که یک لیست دریافت کند و همه زیرلیست‌های ممکن از المان‌های آن لیست را بعنوان خروجی دهد.

۱۰- تابعی بنویسید که دو ماتریس مربعی را به عنوان ورودی بگیرد و حاصلضرب آن دو ماتریس را بعنوان خروجی بدهد.

۱۱- (اختیاری) کلاسی تحت عنوان Graph بنویسید که بتوان با استفاده از آن یک گراف که شامل تعدادی گره و یال بین این گره‌هاست را پیاده سازی کند. این کلاس باید شامل متدهای افزودن گره، افزودن یال، یافتن کوتاه‌ترین مسیر بین دو گره و تشخیص همبند یا ناهمبند بودن گراف باشد. (اگر اطلاعات کمی در مورد گراف دارید میتوانید با مطالع لینک https://en.wikipedia.org/wiki/Graph_theory اطلاعات کلی در مورد گرافها بدست آورید.)

راهنمایی: میتوانید برای ساخت یک گراف از مفهوم ماتریس مجاورت استفاده کنید. این ماتریس مجاورت بهتر است یک دیکشنری باشد با key که گره ها هستند و مقدار هر key در واقع لیستی از گره‌هایی که توسط یال به آن گره وصل هستند.