

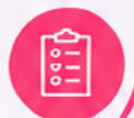
با ایران توشه
یک قدم جلوتر
باش!



ایران توشه
توشه ای برای موفقیت



ایران توشه، مرجع دانلود فایل های کمک آموزشی دانش آموزان و کنکوری ها



دانلود رایگان آزمون های آزمایشی
قلمچی، خیلی سبز، گاج و ...



دانلود رایگان گام به گام تمامی دروس



دانلود رایگان نمونه سوالات امتحانی
(نوبت اول و دوم) و امتحانات نهایی



دانلود رایگان جزوه های کمک آموزشی
و شب امتحانی

همه چیز
برای موفقیت
تو!

Google

دانلود گام به گام دوازدهم تجربی ایران توشه

دانلود گام به گام دوازدهم تجربی ایران توشه

کافیست هر فایلی که نیاز دارید
را در گوگل به این صورت جستجو کنید:

فایل مورد نیاز + ایران توشه

مثال:

دانلود گام به گام دوازدهم تجربی ایران توشه



irantooshe.ir



t.me/irantooshe



گام به گام کار و فناوری هفتم - پودمان ششم

برنامه نویسی

فهرست

۴	صفحه ۵۸ - کار کلاسی - شروع کار با پایتون	۱
۵	صفحه ۵۹ - کار کلاسی - خطای نحوی	۲
۶	صفحه ۶۱ - فعالیت گروهی - متغیر و حدس خروجی	۳
۷	صفحه ۶۱ - کار غیرکلاسی - برنامه های محاسباتی	۴
۸	صفحه ۶۲ - پرسش - تابع int و float	۵
۹	صفحه ۶۲ - کار غیرکلاسی - عملگرهای ریاضی	۶
۱۰	صفحه ۶۴ - پرسش - شرط if	۷
۱۱	صفحه ۶۴ - کار کلاسی - زوج یا فرد	۸
۱۱	صفحه ۶۴ - کار کلاسی - دستور if-else	۹
۱۱	صفحه ۶۵ - کار غیرکلاسی - بزرگتر از دو عدد	۱۰
۱۲	صفحه ۶۵ - کار غیرکلاسی - بزرگتر از سه عدد	۱۱
۱۲	صفحه ۶۵ - پروژه - ورود به سیستم	۱۲

۱۳	صفحه ۶۵ . پروژه — رسم مثلث	۱۳
۱۴	صفحه ۶۵ . پرسش — کتابخانهٔ لاک پشت	۱۴
۱۴	صفحه ۶۷ . کار کلاسی — طراحی کارت شناسایی	۱۵
۱۷	صفحه ۶۸ . پروژه — رسم با لاک	۱۶
۱۷	صفحه ۶۸ . پروژه — رسم با چند لاک	۱۷
۲۰	صفحه ۶۸ . پرسش — حلقهٔ تکرار	۱۸
۲۲	صفحه ۶۹ . کار کلاسی — حلقهٔ for	۱۹
۲۳	صفحه ۶۹ . کار غیرکلاسی — رسم شکل با حلقهٔ for	۲۰
۲۴	صفحه ۷۰ . پروژه — رسم طرح با لاک	۲۱



گام ۱. محیط ویرایشگر را باز کنید.

گام ۲. دستورات مورد نظر را مطابق شکل زیر بنویسید.

گام ۳. از فهرست File گزینه Save را انتخاب و برنامه را با نام دلخواه در محل مناسب ذخیره کنید.

گام ۴. از فهرست Run گزینه Run Module یا F5 را می‌زنیم.

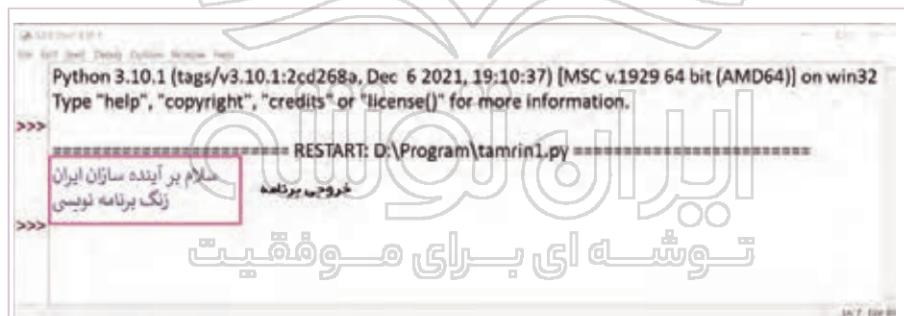
گام ۵. خروجی برنامه در پنجره Shell نمایش داده می‌شود.

در ویرایشگر IDLE از فهرست File گزینه New File را انتخاب می‌کنیم و دو دستور زیر را می‌نویسیم:

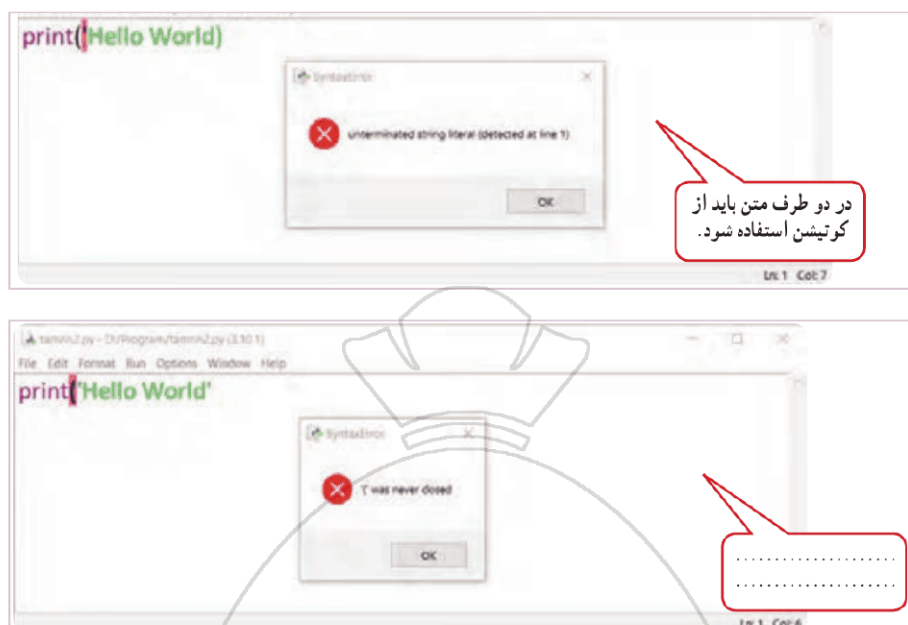
```
print('سلام بر آینده سازان ایران')
print('زنگ برنامه نویسی')
```

سپس برنامه را با نامی مانند tamrin1.py ذخیره می‌کنیم (پسوند فایل برنامه پایتون py است) و با زدن کلید F5 یا گزینه Run Module آن را اجرا می‌کنیم. خروجی در پنجره Shell به صورت زیر نمایش داده می‌شود:

```
سلام بر آینده سازان ایران
زنگ برنامه نویسی
```



یکی از خطاهای رایج در برنامه‌نویسی، خطاهای نوشتاری است که به آن خطای نحوی (Syntax Error) گفته می‌شود. این خطا به دلیل اشتباهات حروف چینی رخ می‌دهد. با هم کلاسی‌های خود، خطاهای زیر را بررسی و دلیل آن را بیان کنید.



۱. شکل ۶-۱۲: متنی که داخل پرانتز نوشته شده، فقط با یک کوتیشن شروع شده و پایان آن کوتیشن ندارد؛ یعنی دستور به صورت `print('Hello World)` نوشته شده است. پایتون پیام `unterminated string literal` (رشته ناتمام) می‌دهد. متن (رشته) باید در دو طرف کوتیشن داشته باشد: `print('Hello World')`.
۲. شکل ۶-۱۳: متن درست داخل کوتیشن نوشته شده است اما پرانتز بازی که بعد از `print` آمده است بسته نشده؛ یعنی دستور به صورت `print('Hello World'` نوشته شده است. پایتون پیام `' was never closed` می‌دهد. جمله داخل کادر خالی شکل را می‌توان این‌گونه کامل کرد: «هر پرانتز بازی که باز می‌شود، باید بسته شود؛ در پایان دستور باید پرانتز بسته قرار داد: `print('Hello World')`».

خروجی هر کد را با دوستان خود حدس بزنید و درباره آن بحث و گفت‌وگو کنید.

```
n=1400
N=1401
print(n)
```

```
n=n + 1400
print(n)
```

```
n=1400
n= n + 1
print(n)
```

```
n=1400
N=1401
print(n)
```

کدها را از راست به چپ، به ترتیب شماره ۱ تا ۴ بررسی می‌کنیم:

۱. کد اول: متغیر n مقدار ۱۴۰۰ می‌گیرد. N متغیر دیگری است، زیرا پایتون به حروف کوچک و بزرگ حساس است و مقدار ۱۴۰۱ در آن ذخیره می‌شود. دستور `print(n)` مقدار متغیر n را چاپ می‌کند؛ خروجی: 1400.
۲. کد دوم: n برابر ۱۴۰۰ است. دستور `n=n+1` یک واحد به آن اضافه می‌کند و مقدار ۱۴۰۱ را دوباره در n می‌گذارد؛ خروجی: 1401. (توجه: دستور باید مثل خط‌های دیگر از ابتدای خط نوشته شود؛ اگر در ابتدای آن فاصله اضافی باشد پایتون خطای `IndentationError` می‌دهد.)
۳. کد سوم: در دستور `n=n+1400` مقدار قبلی n لازم است، اما هنوز مقداری به n داده نشده است. پایتون خطای `NameError: name 'n' is not defined` می‌دهد و چیزی چاپ نمی‌شود.
۴. کد چهارم: مانند کد اول است؛ n برابر ۱۴۰۰ و N متغیری جداگانه با مقدار ۱۴۰۱ است؛ خروجی: 1400.

برنامه‌های زیر را بنویسید.

۱. فاطمه کتاب داستانی را در ۶ ساعت مطالعه کرد و ۱۰ صفحه از آن باقی ماند. اگر این کتاب ۱۰۰ صفحه داشته باشد، فاطمه به طور متوسط در هر ساعت چند صفحه از آن را مطالعه کرده است؟
۲. محاسبه کن یک ساعت و بیست دقیقه و سی ثانیه، چند ثانیه است.
۳. مساحت و محیط یک دایره با شعاع ۵ چقدر است؟
۴. پس‌انداز هفتگی محمد، ۳۰۰۰ تومان است. او حساب کرد ۵ هفته پس‌انداز او، نصف قیمت کیفی است که دوست دارد آن را بخرد. برنامه‌ای بنویسید که قیمت کیف را محاسبه و چاپ کند.

۱. صفحه‌هایی که فاطمه خوانده است ۱۰۰ منهای ۱۰ یعنی ۹۰ صفحه است. این تعداد را بر ۶ ساعت تقسیم می‌کنیم:

```
all_pages = 100
remain = 10
hours = 6
speed = (all_pages - remain) / hours
print(speed)
```

خروجی: 15.0؛ یعنی فاطمه به طور متوسط در هر ساعت ۱۵ صفحه خوانده است.

۲. یک ساعت برابر ۳۶۰۰ ثانیه و یک دقیقه برابر ۶۰ ثانیه است:

```
seconds = 1 * 3600 + 20 * 60 + 30
print(seconds)
```

خروجی: 4830؛ یعنی یک ساعت و بیست دقیقه و سی ثانیه برابر ۴۸۳۰ ثانیه است.

۳. مساحت دایره برابر πr^2 و محیط آن برابر $2\pi r$ است. عدد π را تقریباً ۳٫۱۴ می‌گیریم:

```
r = 5
pi = 3.14
area = pi * r ** 2
perimeter = 2 * pi * r
print(area)
print(perimeter)
```

خروجی: 78.5 برای مساحت و 31.400000000000002 برای محیط. (عدد دوم همان ۳۱٫۴ است؛ رایانه در محاسبات اعشاری گاهی رقم‌های بسیار کوچکی در آخر عدد نشان می‌دهد.)

۴. اگر پس‌انداز ۵ هفته نصف قیمت کیف باشد، قیمت کیف دو برابر پس‌انداز ۵ هفته است:

```
weekly = 3000
weeks = 5
saving = weekly * weeks
price = 2 * saving
print(price)
```

خروجی: 30000؛ یعنی قیمت کیف ۳۰۰۰۰ تومان است.

پرسش

صفحه ۶۲

۵

اگر عدد ۲۵ به عنوان ورودی در دو دستور زیر داده شود. خروجی‌ها را با هم مقایسه و درباره آن گفت‌وگو کنید.

```
n=int(input('مقداری را وارد کنید'))  
print(n)
```

```
n=float(input('مقداری را وارد کنید'))  
print(n)
```

- دستور سمت راست (float): ورودی را به عدد اعشاری تبدیل می‌کند؛ خروجی 25.0 است.
- دستور سمت چپ (int): ورودی را به عدد صحیح تبدیل می‌کند؛ خروجی 25 است.
- نتیجه: مقدار هر دو یکی است، اما نوع داده فرق دارد. int عدد صحیح (بدون ممیز) و float عدد اعشاری (دارای ممیز) می‌سازد؛ به همین دلیل در خروجی float عدد با «.۰» نمایش داده می‌شود.



برنامه‌های زیر را بنویسید.

۱. برنامه‌ای بنویسید که شعاع قاعده و ارتفاع منبع آب استوانه‌ای را از ورودی دریافت و سپس محاسبه کند که چند مترمکعب آب می‌گیرد.

۲. در درس علوم آموختید که کار انجام شده با مقدار نیرو در اندازه جابه‌جایی برابر است. این رابطه را با تساوی $W=F.d$ نشان می‌دهیم. برنامه‌ای بنویسید که کار انجام شده و مقدار نیرو را از ورودی دریافت و سپس میزان جابه‌جایی را محاسبه و چاپ کند.

۳. طول یک فنر ۱۰ سانتی‌متر است. وقتی وزنه‌ای به جرم x به آن وصل شود، طول فنر از رابطه $y=0.8x + 10$ محاسبه می‌شود. برنامه‌ای بنویسید که از ورودی جرم وزنه‌ای برحسب کیلوگرم که به آن وصل شده دریافت و سپس، طول فنر را محاسبه و چاپ کند.

۱. حجم استوانه برابر مساحت قاعده ضرب در ارتفاع است: $V = \pi r^2 h$. چون شعاع و ارتفاع ممکن است اعشاری باشند، از float استفاده می‌کنیم.

```
r = float(input('شعاع قاعده (متر) = '))
h = float(input('ارتفاع (متر) = '))
v = 3.14 * r ** 2 * h
print('حجم آب (مترمکعب):', v)
```

مثال: برای $r=2$ و $h=5$ خروجی 62.8 مترمکعب است.

۲. از رابطه $W = F \times d$ نتیجه می‌شود $d = W / F$:

```
w = float(input('کار انجام شده = '))
f = float(input('مقدار نیرو = '))
d = w / f
print('جابه‌جایی:', d)
```

مثال: برای $W=100$ و $F=20$ خروجی 5.0 است.

۳. طول فنر از رابطه $y = 0.8x + 10$ به دست می‌آید:

```
x = float(input('جرم وزنه (کیلوگرم) = '))
y = 0.8 * x + 10
print('طول فنر (سانتی‌متر):', y)
```

مثال: برای $x=5$ خروجی 14.0 سانتی‌متر است.

خروجی قطعه کد روبه‌رو را حدس بزنید:

در این مثال temp درجه هواست.

در صورتی که درجه هوا کمتر از ۱۰ درجه باشد هوا سرد است.

اگر مقدار temp=12 باشد برنامه چه خروجی خواهد داشت؟

کدام دستورات در بلاک کد قرار دارند؟

```
print('سلام')
temp=8
if temp<10 :
    print('هوا سرد است')
    print('لباس گرم بپوشید')
    print('خدانگهدار')
```

به‌ازای temp=8 شرط temp<10 درست است و هر چهار خط چاپ می‌شود:

سلام
هوا سرد است
لباس گرم بپوشید
خدانگهدار

به‌ازای temp=12 شرط temp<10 نادرست است و دستورات بلاک if اجرا نمی‌شوند؛ فقط دو دستور print که بیرون از بلاک قرار دارند اجرا می‌شود:

سلام
خدانگهدار

دو دستور ('هوا سرد است') و ('لباس گرم بپوشید') print که با یک میزان تورفتگی زیر if temp<10 نوشته شده‌اند، در بلاک کد قرار دارند.

با هم کلاسی‌های خود برنامه‌ای بنویسید که یک عدد از ورودی دریافت کند و سپس مشخص کند که این عدد زوج است یا فرد.

راهنمایی: عدد زوج، عددی است که بر ۲ بخش پذیر باشد.

عدد بر ۲ بخش پذیر است اگر باقیمانده تقسیم آن بر ۲ صفر باشد؛ برای محاسبه باقیمانده از عملگر % و برای مقایسه از == استفاده می‌کنیم:

```
n = int(input('یک عدد وارد کنید: '))
if n % 2 == 0:
    print('عدد زوج است')
if n % 2 != 0:
    print('عدد فرد است')
```

مثال: ورودی 7 ← خروجی «عدد فرد است»؛ ورودی 10 ← خروجی «عدد زوج است».

با هم کلاسی‌های خود برنامه قبلی را با استفاده از if-else بنویسید.

چون عدد یا زوج است یا فرد، شرط فقط دو حالت دارد؛ پس با else یک شرط کافی است:

```
n = int(input('یک عدد وارد کنید: '))
if n % 2 == 0:
    print('عدد زوج است')
else:
    print('عدد فرد است')
```

برنامه‌ای بنویسید که دو عدد از ورودی دریافت و سپس عدد بزرگتر را چاپ کند.

```
a = int(input('عدد اول: '))
b = int(input('عدد دوم: '))
if a > b:
    print(a)
else:
    print(b)
```

اگر دو عدد برابر باشند، همان عدد (مقدار b) چاپ می‌شود. مثال: ورودی ۴ و ۹ ← خروجی ۹.

برنامه‌ای بنویسید که سه عدد از ورودی دریافت و سپس عدد بزرگ‌تر را چاپ کند.

فرض می‌کنیم عدد اول بزرگ‌ترین است؛ سپس هر عدد دیگر را با آن مقایسه می‌کنیم و اگر بزرگ‌تر بود، جای آن را می‌گیرد:

```
a = int(input('عدد اول: '))
b = int(input('عدد دوم: '))
c = int(input('عدد سوم: '))
max_number = a
if b > max_number:
    max_number = b
if c > max_number:
    max_number = c
print(max_number)
```

مثال: ورودی ۳، ۱۲ و ۷ ← خروجی ۱۲.

برنامه‌ای بنویسید که نام کاربری و گذرواژه را برای ورود به سیستم دریافت کند و در صورتی که نام کاربری Admin و گذرواژه 12345678 بود پیام «خوش آمدید» صادر شود و در غیر این صورت پیام «دسترسی غیر مجاز است» صادر شود.

هر دو شرط باید با هم برقرار باشند؛ برای این کار از and (و) استفاده می‌کنیم. چون گذرواژه یک رشته است، آن را بدون تبدیل به عدد و داخل کوتیشن مقایسه می‌کنیم:

```
username = input('نام کاربری: ')
password = input('گذرواژه: ')
if username == 'Admin' and password == '12345678':
    print('خوش آمدید')
else:
    print('دسترسی غیر مجاز است')
```

برنامه‌ای بنویسید که طول سه پاره خط را از ورودی دریافت کند و سپس مشخص کند که آیا می‌توان با این سه پاره خط مثلثی رسم کرد یا خیر.

با سه پاره خط وقتی می‌توان مثلث ساخت که مجموع هر دو ضلع از ضلع سوم بزرگ‌تر باشد:

```
a = float(input('طول پاره خط اول: '))
b = float(input('طول پاره خط دوم: '))
c = float(input('طول پاره خط سوم: '))
if a + b > c and a + c > b and b + c > a:
    print('می‌توان مثلث رسم کرد')
else:
    print('نمی‌توان مثلث رسم کرد')
```

مثال: ورودی ۳، ۴ و ۵ ← «می‌توان مثلث رسم کرد»؛ ورودی ۱، ۲ و ۵ ← «نمی‌توان مثلث رسم کرد».



قصد داریم با کدنویسی پایتون یک کارت شناسایی مشابه شکل روبه‌رو طراحی کنیم.
برای شروع چه کاری باید انجام دهیم؟
از کجا باید شروع کنیم؟

Name: Ali
Last Name: Bahrami
Student Code: 1234567890
Mobile: 09131111111

یکی از ایده‌های حل مسئله، تقسیم آن به قسمت‌های کوچک‌تر است. پس نخست مسئله را به بخش‌های کوچک‌تر تقسیم می‌کنیم:

۱. کارت شناسایی از دو بخش تشکیل شده است: کادر مستطیل شکل و متن‌هایی که درون کادر نوشته می‌شود.
۲. کادر مستطیل را هم می‌توان به چهار خط (چهار ضلع) تقسیم کرد؛ هر ضلع با حرکت به جلو و یک چرخش ۹۰ درجه رسم می‌شود.
۳. متن‌ها شامل نام، نام خانوادگی، کد دانش‌آموزی و شماره همراه، هر کدام در یک خط جداگانه نوشته می‌شوند.

برای رسم از کتابخانه turtle (لاک‌پشت) پایتون استفاده می‌کنیم؛ پس شروع کار این است که کتابخانه را با دستور import turtle به برنامه بیاوریم و یک لاک‌پشت بسازیم. سپس با محور مختصات جای شروع رسم را مشخص می‌کنیم، ابتدا کادر را می‌کشیم و بعد متن‌ها را می‌نویسیم.

طراحی کارت شناسایی

گام ۱- از کتابخانه turtle یک نمونه از نوع turtle بسازید.

laki=turtle.Turtle()

توجه: امکان ساخت لاک‌پشت‌های بیشتری هم هست. کافی است دستور بالا را کپی و تکرار کنیم و نام

لاک‌پشت جدید را بگذاریم laki2.

گام ۲- لاک‌ی می‌تواند به شکل‌های مختلف باشد. در این جا همان شکل لاک‌پشت را برای آن انتخاب می‌کنیم.
(shape یعنی شکل)

laki.shape('turtle')

توجه: وقتی می‌خواهیم لاک‌پشت کاری انجام دهد، یک نقطه جلوی نام آن قرار می‌دهیم و سپس آن کار را می‌نویسیم. به کارهایی که لاک‌پشت انجام می‌دهد، متد می‌گوییم. متدهای زیادی از پیش برای لاک‌پشت تعریف شده است که به تدریج با آن‌ها آشنا می‌شویم.

گام ۳- لاک‌ی را حرکت بدهید. دستور روبه‌رو لاک‌ی را ۵۰ پیکسل جلو می‌برد. (laki.forward(50) forward یعنی جلو رفتن)

گام ۴- برنامه را ذخیره و با کلید F5 برنامه را اجرا کنید. (تعداد گام را تغییر و دوباره اجرا کنید)

گام ۵- اگر بخواهیم لاک‌ی وقتی حرکت می‌کند، خطی نکشد، باید به او بگوییم قلم را از زمین بردارد. بنابراین از متد penup استفاده می‌کنیم. pen یعنی قلم و up یعنی بالا. laki.penup()

برنامه را اجرا کنید. چرا این دستور تغییری ایجاد نکرد؟ راه حلی پیدا کنید.

کنجکاوی: با () pendown هم به لاک‌ی می‌گوییم تا قلم را دوباره روی زمین بگذارد.

گام ۶- با دستورات چرخش به چپ یا راست رسم مستطیل را کامل کنید. (laki.left(90) و laki.right(90))

گام ۷- با استفاده از جدول زیر کارت شناسایی را تکمیل کنید.

جدول دستورات کارت شناسایی

کار	دستور
تغییر موقعیت لاک‌ی به مختصات (x,y)	laki.goto(x,y)
نوشتن متن با لاک‌ی	laki.write('...', font=('...', ...)) (متن موردنظر، نام فونت، اندازه)
تغییر رنگ خطوط و متن	laki.color('...') (نام رنگ)
تغییر ضخامت خطوط	laki.width(...) (ضخامت)
پنهان کردن لاک‌ی	laki.hideturtle()

چرا () penup تغییری ایجاد نکرد؟ اگر دستور () laki.penup بعد از laki.forward(50) نوشته شود، لاک‌ی پیش از آن حرکت کرده و خط را کشیده است؛ پس بالا بردن قلم اثری در خط قبلی ندارد. راه حل: دستور () penup را پیش از دستور forward(50) بنویسیم تا لاک‌ی بدون کشیدن خط جلو برود.

برنامه کامل: نخست قلم را بالا می‌بریم و لاک‌ی را به گوشه پایین چپ کارت می‌بریم، قلم را می‌گذاریم و مستطیل ۳۰۰ در ۱۴۰ را با چهار بار «جلو رفتن و چرخش ۹۰ درجه به چپ» می‌کشیم. سپس قلم را بالا می‌بریم، لاک‌ی را به ابتدای هر خط می‌بریم و متن را می‌نویسیم.

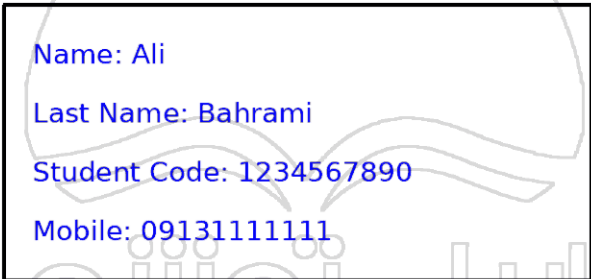
```
import turtle
laki = turtle.Turtle()
laki.shape('turtle')
laki.width(2)

# رسم کادر مستطیل
laki.penup()
```

```
laki.goto(-150, -70)
laki.pendown()
laki.forward(300)
laki.left(90)
laki.forward(140)
laki.left(90)
laki.forward(300)
laki.left(90)
laki.forward(140)
laki.left(90)

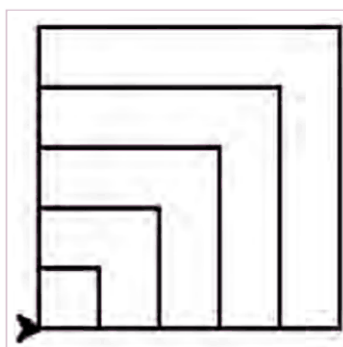
# نوشتن متن‌ها
laki.penup()
laki.color('blue')
laki.goto(-135, 40)
laki.write('Name: Ali', font=('Arial', 12, 'normal'))
laki.goto(-135, 10)
laki.write('Last Name: Bahrami', font=('Arial', 12, 'normal'))
laki.goto(-135, -20)
laki.write('Student Code: 1234567890', font=('Arial', 12, 'normal'))
laki.goto(-135, -50)
laki.write('Mobile: 09131111111', font=('Arial', 12, 'normal'))
laki.hideturtle()
```

خروجی برنامه:



Name: Ali
Last Name: Bahrami
Student Code: 1234567890
Mobile: 09131111111

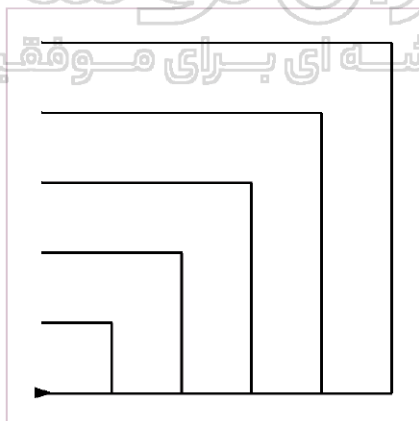
شکل‌های زیر را با لاک‌ی ترسیم کنید.



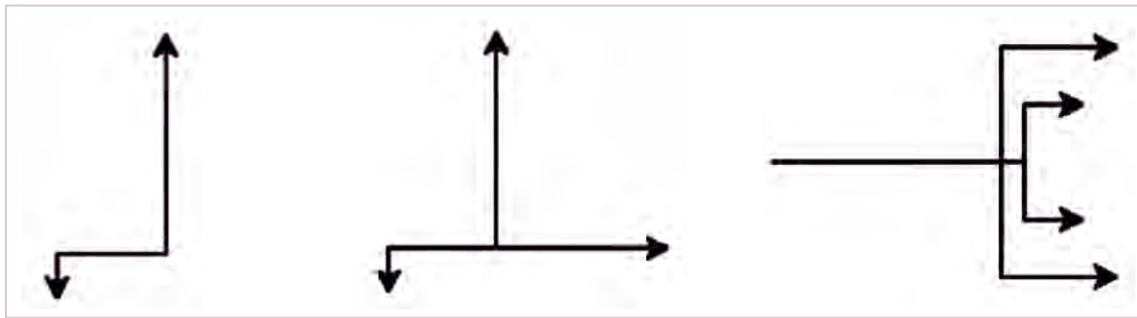
شکل از پنج «ل» تودرتو تشکیل شده که ضلع چپ ندارند و همه از یک نقطه (محل شروع لاک‌ی) آغاز می‌شوند. اندازه ضلع‌ها ۵۰، ۱۰۰، ۱۵۰، ۲۰۰ و ۲۵۰ است. هر بار سه ضلع (پایین، راست و بالا) را می‌کشیم و سپس بدون کشیدن خط به نقطه شروع برمی‌گردیم:

```
import turtle
laki = turtle.Turtle()
laki.width(2)
for i in range(1, 6):
    side = i * 50
    laki.forward(side)
    laki.left(90)
    laki.forward(side)
    laki.left(90)
    laki.forward(side)
    laki.penup()
    laki.goto(0, 0)
    laki.left(180)
    laki.pendown()
```

ایران توشه
توشه‌ای برای موفقیت



شکل‌های زیر را با لاک‌ی ترسیم کنید.



هر پاره‌خط‌دارِ فلش‌دار را با یک لاک‌ی (با شکل arrow) رسم می‌کنیم. اندازه‌ها تقریبی‌اند.

شکل اول (دو لاک‌ی: یکی برای خط عمودی و یکی برای خط افقی کوتاه و فلش پایین):

```
import turtle
laki1 = turtle.Turtle()
laki1.shape('arrow')
laki2 = turtle.Turtle()
laki2.shape('arrow')
laki1.left(90)
laki1.forward(150)
laki2.left(180)
laki2.forward(80)
laki2.left(90)
laki2.forward(30)
```

شکل دوم (سه لاک‌ی: فلش بالا، فلش راست و خط چپ با فلش پایین):

```
import turtle
laki1 = turtle.Turtle()
laki1.shape('arrow')
laki2 = turtle.Turtle()
laki2.shape('arrow')
laki3 = turtle.Turtle()
laki3.shape('arrow')
laki1.left(90)
laki1.forward(150)
laki2.forward(120)
laki3.left(180)
laki3.forward(80)
laki3.left(90)
laki3.forward(30)
```

شکل سوم (پنج لاک‌ی: یک خط افقی بلند، دو شاخهٔ بزرگ بالا و پایین و دو شاخهٔ کوچک وسط؛ لاک‌های ۴ و ۵ بدون کشیدن خط به نقطهٔ شروع می‌روند):

```
import turtle
laki1 = turtle.Turtle()
laki2 = turtle.Turtle()
laki3 = turtle.Turtle()
laki4 = turtle.Turtle()
laki5 = turtle.Turtle()
```

```
laki1.shape('arrow')  
laki2.shape('arrow')  
laki3.shape('arrow')  
laki4.shape('arrow')  
laki5.shape('arrow')
```

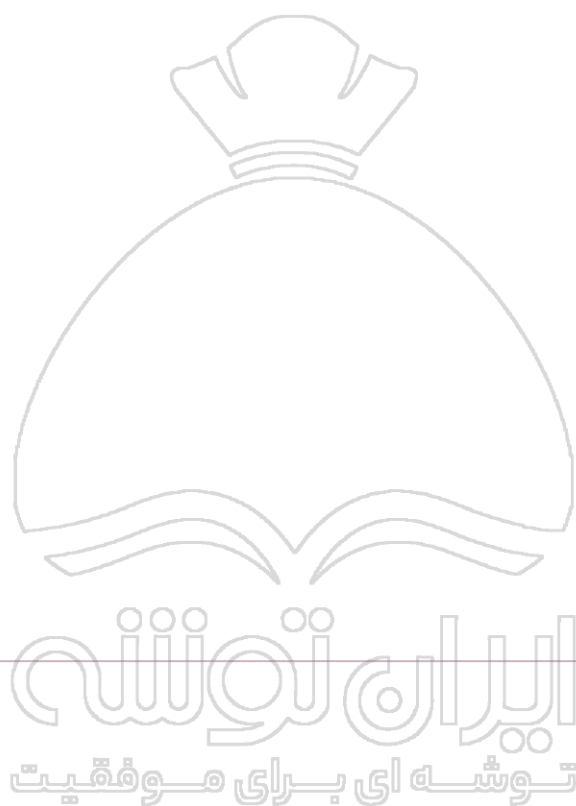
```
laki1.penup()  
laki1.goto(-130, 0)  
laki1.pendown()  
laki1.forward(155)
```

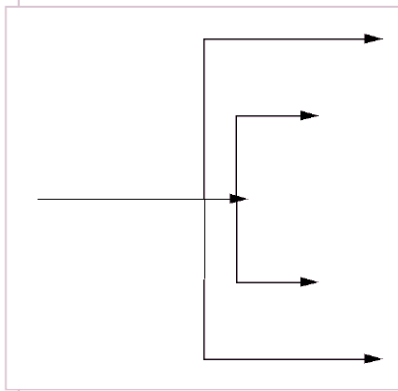
```
laki2.left(90)  
laki2.forward(125)  
laki2.right(90)  
laki2.forward(130)
```

```
laki3.right(90)  
laki3.forward(125)  
laki3.left(90)  
laki3.forward(130)
```

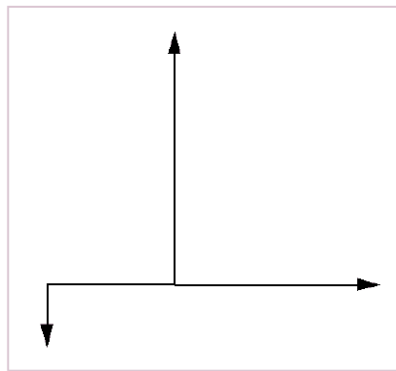
```
laki4.penup()  
laki4.goto(25, 0)  
laki4.pendown()  
laki4.left(90)  
laki4.forward(65)  
laki4.right(90)  
laki4.forward(55)
```

```
laki5.penup()  
laki5.goto(25, 0)  
laki5.pendown()  
laki5.right(90)  
laki5.forward(65)  
laki5.left(90)  
laki5.forward(55)
```

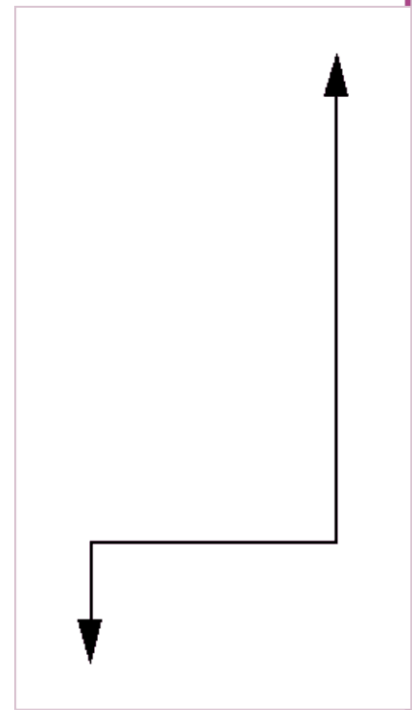




شکل سوم



شکل دوم



شکل اول

پرسش

صفحه ۶۸

۱۸

به کدهای روبه‌رو نگاه کنید و کدهای تکراری متوالی را مشخص کنید.
کد صفحه قبل یک مربع در وسط صفحه رسم می‌کند. حالا خودتان یک شش ضلعی و سپس یک هشت ضلعی رسم کنید.

```
import turtle
laki=turtle.Turtle()
laki.shape('turtle')
laki.color('green')
laki.width(3)
laki.penup()
laki.goto(-100,-100)
laki.pendown()
laki.forward(200)
laki.left(90)
laki.forward(200)
laki.left(90)
laki.forward(200)
laki.left(90)
laki.forward(200)
laki.left(90)
```

ایران توشه
توشه ای برای موفقیت

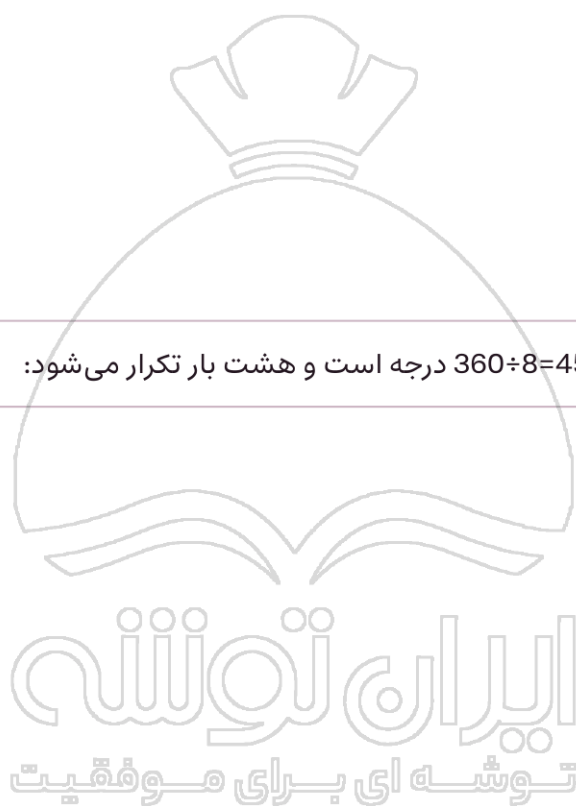
کدهای تکراری متوالی: دو دستور laki.forward(200) و laki.left(90) پشت سر هم چهار بار تکرار شده‌اند (برای چهار ضلع مربع).

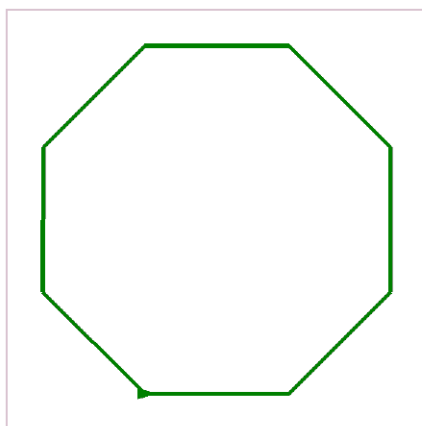
شش ضلعی: شش ضلعی منتظم شش ضلع برابر دارد و زاویه چرخش در هر گوشه $360 \div 6 = 60$ درجه است؛ پس شش بار «جلو رفتن و چرخش ۶۰ درجه» را تکرار می‌کنیم. برای این که شکل وسط صفحه قرار بگیرد، لاکي را ابتدا به نقطه مناسب می‌بریم (ضلع ۱۰۰):

```
import turtle
laki = turtle.Turtle()
laki.shape('turtle')
laki.color('green')
laki.width(3)
laki.penup()
laki.goto(-50, -87)
laki.pendown()
laki.forward(100)
laki.left(60)
laki.forward(100)
laki.left(60)
laki.forward(100)
laki.left(60)
laki.forward(100)
laki.left(60)
laki.forward(100)
laki.left(60)
laki.forward(100)
laki.left(60)
```

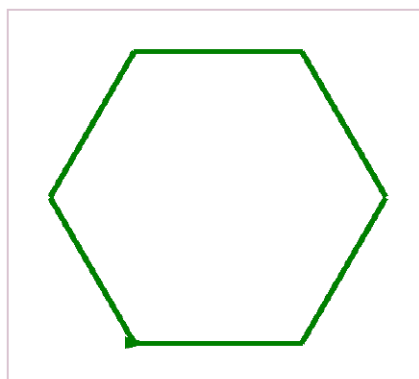
هشت ضلعی: زاویه چرخش $360 \div 8 = 45$ درجه است و هشت بار تکرار می‌شود:

```
import turtle
laki = turtle.Turtle()
laki.shape('turtle')
laki.color('green')
laki.width(3)
laki.penup()
laki.goto(-50, -121)
laki.pendown()
laki.forward(100)
laki.left(45)
laki.forward(100)
laki.left(45)
laki.forward(100)
laki.left(45)
laki.forward(100)
laki.left(45)
laki.forward(100)
laki.left(45)
laki.forward(100)
laki.left(45)
laki.forward(100)
laki.left(45)
laki.forward(100)
laki.left(45)
laki.forward(100)
laki.left(45)
```





هشت ضلعی



شش ضلعی

کار کلاسی

صفحه ۶۹

۱۹

با استفاده از دستور for، کدهای رسم شش ضلعی و هشت ضلعی را بازنویسی کنید.

در شش ضلعی دو دستور forward(100) و left(60) شش بار و در هشت ضلعی دو دستور forward(100) و left(45) هشت بار تکرار می‌شوند؛ پس تعداد تکرار را در range می‌نویسیم و دو دستور را با تورفتگی داخل بلاک for قرار می‌دهیم.

شش ضلعی:

```
import turtle
laki = turtle.Turtle()
laki.shape('turtle')
laki.color('green')
laki.width(3)
laki.penup()
laki.goto(-50, -87)
laki.pendown()
for i in range(6):
    laki.forward(100)
    laki.left(60)
```

ایران توشه
توشه ای برای موفقیت

هشت ضلعی:

```
import turtle
laki = turtle.Turtle()
laki.shape('turtle')
laki.color('green')
laki.width(3)
laki.penup()
laki.goto(-50, -121)
laki.pendown()
for i in range(8):
    laki.forward(100)
    laki.left(45)
```

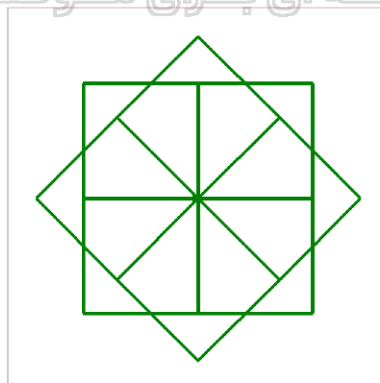
شکل روبه‌رو را با کدهایی که تا این جا آموختید، رسم کنید.
راهنمایی: این شکل از ۸ مربع با زاویهٔ چرخش ۴۵ درجه تشکیل شده است.



یک مربع با چهار بار «جلو رفتن و چرخش ۹۰ درجه» رسم می‌شود. پس از هر مربع، لاکِی را ۴۵ درجه می‌چرخانیم و مربع بعدی را می‌کشیم؛ این کار را ۸ بار تکرار می‌کنیم. چون در پایان هر مربع لاکِی ۹۰ درجه می‌چرخد، در آخرین چرخش مجموع $۹۰ + ۴۵ = ۱۳۵$ درجه می‌چرخد:

```
import turtle
laki = turtle.Turtle()
laki.shape('turtle')
laki.color('green')
laki.width(3)
for i in range(8):
    laki.forward(100)
    laki.left(90)
    laki.forward(100)
    laki.left(90)
    laki.forward(100)
    laki.left(90)
    laki.forward(100)
    laki.left(135)
```

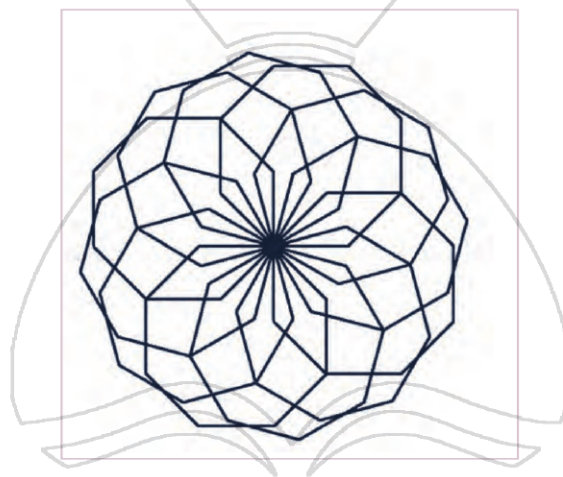
ایران توشه
توشه‌ای برای موفقیت



برنامه‌ای بنویسید که تعداد اضلاع شکل و تعداد تکرار آن را از ورودی دریافت و سپس طرح کامل را ترسیم کند. از دستورات جدول زیر کمک بگیرید.
راهنمایی: شکل زیر یک هشت ضلعی است که ۱۲ مرتبه تکرار شده است.

دستورات کمکی

کار	دستور
تغییر سرعت لاک	laki.speed(...) (سرعت)
رسم دایره با لاک	laki.circle(...) (شعاع)
دریافت از ورودی توسط لاک	laki.textinput('...', '...') (عنوان کادر، پیام)

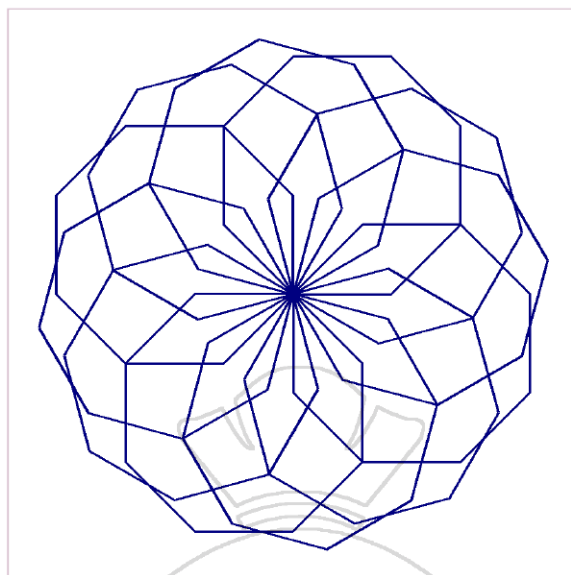


هر شکل با حلقه داخلی رسم می‌شود (زاویه چرخش در هر گوشه $360/sides$ است). پس از کشیدن هر شکل، لاک به اندازه $360/repeat$ درجه می‌چرخد تا شکل بعدی در جهت تازه کشیده شود و شکل‌ها دور یک نقطه مشترک بچرخند. برای افزایش سرعت از `laki.speed(0)` استفاده می‌کنیم. دریافت از ورودی با کادر ورودی لاک پشت به صورت `turtle.textinput` نوشته می‌شود و مقدار دریافتی رشته است، پس با `int` به عدد صحیح تبدیل می‌شود (می‌توان به جای آن از `input()` هم استفاده کرد).

```
import turtle
laki = turtle.Turtle()
laki.speed(0)
laki.color('navy')
laki.width(2)
sides = int(turtle.textinput('ورودی: ', 'تعداد اضلاع شکل را وارد کنید'))
repeat = int(turtle.textinput('ورودی: ', 'تعداد تکرار شکل را وارد کنید'))
for i in range(repeat):
    for j in range(sides):
        laki.forward(80)
```

```
laki.left(360 / sides)
laki.left(360 / repeat)
laki.hideturtle()
```

با ورودی‌های ۸ (اضلاع) و ۱۲ (تکرار) خروجی برنامه به صورت زیر است:



این فایل توسط سایت ایران توشه تهیه شده است.

irantooshe.ir



ایران توشه
توشه ای برای موفقیت