



برنامه نویسی پایتون



```
while int(input('A: ')) == 0:
    print("A can't be zero!")
while int(input('B: ')) == 0:
    print("B can't be zero!")
while int(input('C: ')) == 0:
    print("C can't be zero!")
while int(input('D: ')) == 0:
    print("D can't be zero!")
while int(input('E: ')) == 0:
    print("E can't be zero!")
while int(input('F: ')) == 0:
    print("F can't be zero!")
while int(input('G: ')) == 0:
    print("G can't be zero!")
while int(input('H: ')) == 0:
    print("H can't be zero!")
while int(input('I: ')) == 0:
    print("I can't be zero!")
while int(input('J: ')) == 0:
    print("J can't be zero!")
while int(input('K: ')) == 0:
    print("K can't be zero!")
while int(input('L: ')) == 0:
    print("L can't be zero!")
while int(input('M: ')) == 0:
    print("M can't be zero!")
while int(input('N: ')) == 0:
    print("N can't be zero!")
while int(input('O: ')) == 0:
    print("O can't be zero!")
while int(input('P: ')) == 0:
    print("P can't be zero!")
while int(input('Q: ')) == 0:
    print("Q can't be zero!")
while int(input('R: ')) == 0:
    print("R can't be zero!")
while int(input('S: ')) == 0:
    print("S can't be zero!")
while int(input('T: ')) == 0:
    print("T can't be zero!")
while int(input('U: ')) == 0:
    print("U can't be zero!")
while int(input('V: ')) == 0:
    print("V can't be zero!")
while int(input('W: ')) == 0:
    print("W can't be zero!")
while int(input('X: ')) == 0:
    print("X can't be zero!")
while int(input('Y: ')) == 0:
    print("Y can't be zero!")
while int(input('Z: ')) == 0:
    print("Z can't be zero!")
while int(input('A: ')) == 0:
    print("A can't be zero!")
while int(input('B: ')) == 0:
    print("B can't be zero!")
while int(input('C: ')) == 0:
    print("C can't be zero!")
while int(input('D: ')) == 0:
    print("D can't be zero!")
while int(input('E: ')) == 0:
    print("E can't be zero!")
while int(input('F: ')) == 0:
    print("F can't be zero!")
while int(input('G: ')) == 0:
    print("G can't be zero!")
while int(input('H: ')) == 0:
    print("H can't be zero!")
while int(input('I: ')) == 0:
    print("I can't be zero!")
while int(input('J: ')) == 0:
    print("J can't be zero!")
while int(input('K: ')) == 0:
    print("K can't be zero!")
while int(input('L: ')) == 0:
    print("L can't be zero!")
while int(input('M: ')) == 0:
    print("M can't be zero!")
while int(input('N: ')) == 0:
    print("N can't be zero!")
while int(input('O: ')) == 0:
    print("O can't be zero!")
while int(input('P: ')) == 0:
    print("P can't be zero!")
while int(input('Q: ')) == 0:
    print("Q can't be zero!")
while int(input('R: ')) == 0:
    print("R can't be zero!")
while int(input('S: ')) == 0:
    print("S can't be zero!")
while int(input('T: ')) == 0:
    print("T can't be zero!")
while int(input('U: ')) == 0:
    print("U can't be zero!")
while int(input('V: ')) == 0:
    print("V can't be zero!")
while int(input('W: ')) == 0:
    print("W can't be zero!")
while int(input('X: ')) == 0:
    print("X can't be zero!")
while int(input('Y: ')) == 0:
    print("Y can't be zero!")
while int(input('Z: ')) == 0:
    print("Z can't be zero!")
```





مجموعه کتاب‌های علامه حلی

برنامه نویسی پایتون (۲)

- محمدرضا جهانگیر
- علیرضا صالحه
- علی‌اکبر محمدی ساعی





شناسنامه
کتاب

سرشناسه : جهانگیر، محمدرضا، ۱۳۵۴
عنوان و نام پدیدآور : برنامه‌نویسی، پایتون (۲) / مؤلفان محمدرضا جهانگیر، علیرضا صالحه، علی‌اکبر محمدی ساعی.
مشخصات نشر : تهران: انتشارات حلی: دانش پژوهان جوان، ۱۴۰۰.
مشخصات ظاهری : ۱۰۴ ص.: تصویر، جدول، نمودار
فروست : مجموعه کتاب‌های علامه‌حلی
شابک : ۹۷۸-۶۰۰-۴۹۶-۲۳۳-۹
وضعیت فهرست نویسی : فیپا
یادداشت : چاپ قبلی: انتشارات حلی، ۱۳۹۶ (۱۰۸ ص) (چاپ هفتم)
موضوع : پایتون (زبان برنامه‌نویسی کامپیوتر)
موضوع : Python (Computer program language)
شناسه افزوده : صالحه، علیرضا، ۱۳۷۲-
شناسه افزوده : محمدی ساعی، علی‌اکبر، ۱۳۶۹-
رده‌بندی کنگره : QA ۷۶/۷۳
رده بندی دیویی : ۰۰۵/۱۳۳
شماره کتابشناسی ملی : ۸۷۴۶۰۴۸



عنوان کتاب : برنامه‌نویسی: پایتون (۲)
ناشر : انتشارات حلی
ناشر همکار : انتشارات دانش پژوهان جوان
مؤلف : محمدرضا جهانگیر، علیرضا صالحه، علی‌اکبر محمدی ساعی
مسئول هم‌آهنگی : سمیه‌سادات فاطمی
صفحه‌آرا : راضیه‌سادات فرهانیان
طراح جلد : الهه شرفی
تصویرسازان : محمدحسین صفدریان
سال چاپ : ۱۴۰۱
نوبت چاپ : هفتم
شمارگان : ۲۰۰۰ جلد
قیمت : ۶۸۰۰۰ تومان
شماره شابک : ۹۷۸-۶۰۰-۴۹۶-۲۳۳-۹



تهران، نمایان انقلاب، میدان فردوسی، ابتزای کوچه براتی، پلاک ۱۶ و ۱۴

تلفن دفتر مرکزی: ۶۶۷۴۴۳۸۴-۵

کلیه حقوق این اثر برای ناشر محفوظ است.

هیچ شخص حقیقی یا حقوقی حق برداشت تمام یا قسمتی از اثر را به صورت چاپ، فتوکپی، جزوه و مجازی ندارد.

متخلفان به موجب بند ۵ از ماده ۲ قانون حمایت از ناشران تحت پیگرد قانونی قرار می‌گیرند.



بالب است
برای

	فصل ۰ یادآوری	درس نامه ۵	تمرین ۲۴
---	-------------------------	------------	----------

درس نامه ۲۷	فصل ۱ تابع	
تمرین ۴۲		

	فصل ۲ لیست	درس نامه ۴۵	تمرین ۵۸
---	----------------------	-------------	----------

درس نامه ۶۱	فصل ۳ کار با رشته و فایل	
تمرین ۸۰		

	فصل ۴ برنامه نویسی الگوریتمی	درس نامه ۸۳	تمرین ۱۰۱
---	--	-------------	-----------

قبل از شروع به مطالعه کتاب، این قسمت را بخوانید:

وقتی شروع به خواندن این کتاب کنید با بخش‌های مختلفی مواجه می‌شوید که غالباً یک لاک‌پشت متفاوت در اول هرکدام وجود دارد. برای هرکدام از این بخش‌ها از شما انتظار داریم کار متفاوتی انجام دهید. این قسمت‌ها بر اساس تئوری‌های نوین آموزش و تجارب موفق تدریس برای آموزش دانش‌آموزان مستعد طراحی شده است. این بخش‌ها شامل:

درخت دانش: در صفحه دوم هر فصل، نمودار دایره‌ای شکلی کشیده شده که به ما کمک می‌کند بفهمیم در آن فصل مطالب علمی چطوری تقسیم‌بندی شده و ارتباط آن‌ها با هم چیست. درواقع این بخش نقشه‌ای است برای گم نشدن در موضوعات علمی.

اهداف رفتاری: زیر هر درخت دانش، چند جمله نوشته شده که از اول کار معلوم کند که این فصل را می‌خوانیم که چه بشود. خوب است در آخر فصل هم برگردیم و ببینیم که می‌توانیم کارهایی را که در این بخش گفته انجام دهیم یا نه.

پاسخگو باش: در این قسمت باید پاسخگو باشیم. پاسخگوی سؤالی که پرسیده شده و انتظار می‌رود بعد از خواندن درس تا آن قسمت، بتوانیم باکمی فکر کردن به آن جواب دهیم.

فسفر بسوزان: شاید لازم باشد مقدار بیشتری از مغز خودمان استفاده کنیم و قدری از فسفرهای ذخیره‌شده را بسوزانیم! سؤالاتی که در بخش فسفر بسوزان مطرح می‌شود فقط با خواندن مطالب درسی قابل پاسخگویی نیست و باید کمی بیش از معمول درباره آن‌ها فکر کنیم.

جالب است بدانی: برای افرادی که دوست دارند بیشتر از سطح استاندارد با موضوعات آشنا شوند این قسمت توصیه می‌شود. در این قسمت مطالبی آورده شده که خواندن و یادگرفتن آن الزامی نیست، ولی آن قدر جذاب است که نشود به راحتی بی‌خیال خواندن آن شد.

لغت‌نامه: ما دانش‌آموزان مستعد و متفاوت (!) دوست داریم بتوانیم علاوه بر مطالب درسی، جست‌وجویی هم بکنیم و ببینیم در دنیا درباره موضوع درسی ما چه چیزی وجود دارد، برای همین در پایان هر فصل لغات مهم فصل با معادل انگلیسی آن آورده شده است.

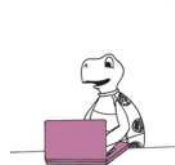
تمرین‌ها: در آخر هر فصل تمرین‌های مرتبط با آن آورده شده است. از آنجایی که مؤلفان کتاب از دبیران باسابقه هستند پس تعداد تمرین‌ها، وقت لازم برای انجام آن‌ها، تعداد سؤالات سخت و آسان و نوع سؤالات با برنامه و محاسبه تعیین شده است پس خیالتان راحت باشد که همه تمرین‌ها را در طول سال می‌شود انجام داد. تمرین‌ها براساس موضوعات هر فصل بخش‌بندی شده؛ بنابراین لازم نیست برای تمرین منتظر پایان فصل باشید، در پایان هر مبحث می‌توانید به بخش تمرین‌ها مراجعه کنید و تمرین‌های همان مبحث را حل کنید.

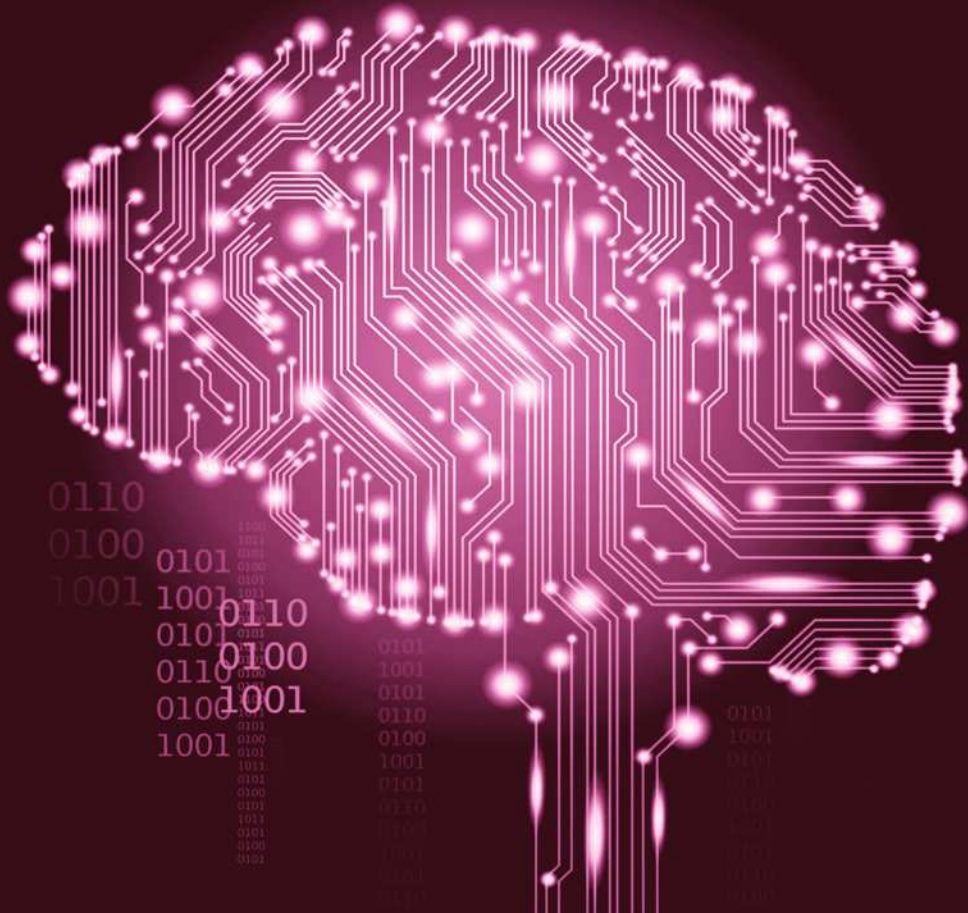
دست‌به‌کد شو: برنامه‌نویسی درسی کاربردی است که در حین آموزش آن لازم است شما هم دست به کد بشوید. در بخش دست به کد شو از شما خواسته شده تا سعی کنید خودتان برنامه را بنویسید. حواستان باشد این بخش، قسمت مهمی از روند درسی است و نمی‌شود بدون دست‌به‌کد شدن برنامه‌نویسی یاد گرفت.

اشتباهات رایج: همان‌طور که از اسمش مشخص است، در این قسمت اشتباهاتی که ممکن است برای هرکسی پیش آید را برای شما توضیح داده‌ایم تا شما دیگر آن‌ها را تکرار نکنید! می‌توان گفت، اگر قرار باشد در بخش‌های دیگر راه برنامه‌نویسی را یاد بگیرید، در این قسمت با چاه‌های آن آشنا می‌شوید.

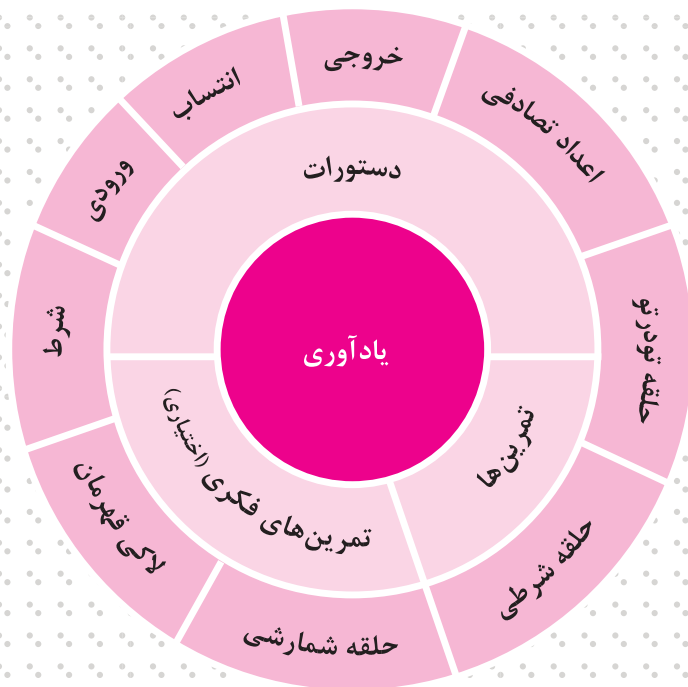
چه می‌کنه: در این قسمت، یک برنامه کامل برای شما نوشته‌ایم و از شما انتظار داریم بگویید این برنامه برای چه هدفی نوشته شده، چه کاری انجام می‌دهد و برای ورودی‌های مختلف، چه خروجی تولید می‌کند.

در ضمن شما هم می‌توانید برای ما مطالب و مسئله ارسال کنید! مطالب و مسئله‌هایی که خودتان از آن‌ها لذت برده‌اید به آدرس: ketab.helli@gmail.com





فصل صفر یادآوری



اگر این فصل را به خوبی مطالعه کنی و کارهای خواسته شده را به دقت انجام دهی:

- نکاتی را که در گذشته از کتاب پایتون (1) فراموش کرده‌ای یا خوب متوجه نشده‌ای، به خوبی یاد می‌گیری.
- می‌توانی از تمرین‌های یادآوری به عنوان محک خوبی برای آزمودن مهارت برنامه‌نویسی خود استفاده کنی.
- نشانه تسلط بر مطالب این فصل، حل کم‌زحمت تمرین‌های یادآوری است، چنانچه در حل این تمرین‌ها دچار مشکل بودی حتماً به بخش مربوطه از کتاب پایتون (1) مراجعه کن.
- کسانی هم که ادعای برنامه‌نویسی دارند یا تمرین‌های یادآوری را خیلی ساده می‌پندارند، می‌توانند درستی ادعای خود را با انجام تمرین‌های فکری بسنجند.



لهراف رفتاری

دستور خروجی

در فصل سوم کتاب پایتون (۱) با دستور پرینت (print) آشنا شدیم. در آنجا دیدیم که منظور ما از چاپ کردن، نوشتن روی صفحه نمایش است و نه چاپگر!

قالب دستور:

print (یک عبارت)

دستور پرینت یک عبارت را روی صفحه نمایشگر می‌نویسد.

چاپ رشته

آنچه بین دو گیومه می‌آید که می‌تواند زنجیره یا رشته‌ای از حروف، اعداد و علامت قابل‌نوشتنی باشد را رشته می‌گوییم. رشته برای پایتون قابل‌فهم نیست و به معنای آن فکر نمی‌کند، مثلاً برای چاپ پیغام خوش‌آمدگویی به کاربر می‌توانیم متن موردنظر را داخل گیومه در داخل پرانتز دستور پرینت، در اول برنامه بیاوریم. انگلیسی یا فارسی بودن متن پیغام هم برای پایتون اهمیتی ندارد، مانند:

print ("به برنامه من خوش آمدید")

یکی از نشانه‌های یک برنامه کاربر پسند (User Friendly)، استفاده مناسب برنامه‌نویس از دستور پرینت در چاپ پیام‌های مناسب و به‌جا است. این پیام‌ها می‌تواند برای راهنمایی کاربر و افزایش سادگی کار با برنامه بسیار مفید باشد، همچنین می‌تواند در زمان توسعه و ارتقاء برنامه به نسخه‌های بالاتر (در آینده)، با بالابردن خوانایی متن برنامه، حسابی به کار برنامه‌نویس بیاید.



بالب (است) برای

چاپ اعداد یا نتیجه محاسبه یک عبارت ریاضی

هر عبارت محاسباتی را می‌توان با پرینت محاسبه و چاپ کرد. این را هم دیدیم، برای آنکه اعداد و یا عبارت ریاضی برای پایتون قابل‌فهم باشد نباید آن را در گیومه بگذاریم. پایتون بین عملگرهای ریاضی تفاوت قائل می‌شود و بعضی از آن‌ها را بر بقیه اولویت می‌دهد (به قول خودمان پارتی‌بازی می‌کند). اولویت عملگرهای ریاضی در پایتون در جدول روبه‌رو آمده است:

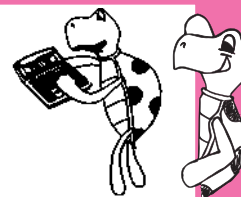
اولویت	ریاضی	پایتون
۱	توان	**
۲	×	*
۲	÷	/
۳	+	+
۳	-	-

همان‌طور که در جدول می‌بینید توان دارای بالاترین اولویت است. ضرب و تقسیم در رده بعدی قرار دارند (به عبارت دیگر ضرب و تقسیم هم اولویت هستند) و در نهایت جمع و تفریق قرار دارند (جمع و تفریق هم اولویت مساوی دارند). در این میان یک استثنا وجود دارد و آن هم وجود پرانتز در عبارت محاسباتی است. پرانتز می‌تواند اولویت‌ها را به هم بزند، برای مثال کاری کند که عمل جمع قبل از توان محاسبه شود، به عبارت دیگر پرانتز دارای بالاترین اولویت است.



print(3**(2+1))

خروجی دستور زیر چیست؟



بله می‌کنه!

بله درست حدس زدید جواب صحیح ۲۷ است، چون اولویت پرانتز از همه چیز بالاتر است اول عمل جمع و بعد از آن، عمل توان انجام می‌شود.

په می‌کنند؟



خروجی دستور زیر چیست؟

```
print(3*6+(11-2)/3)
```

اگر درست محاسبه کرده باشید باید به جواب ۲۱ رسیده باشید، چون اولویت پرانتز از همه چیز بالاتر است اول عمل تفریق انجام می‌شود. بین عمل ضرب و تقسیم که هم اولویت هستند، آنکه سمت چپ‌تر است (در اینجا عمل ضرب) اول انجام می‌شود سپس عمل تقسیم و در نهایت عمل جمع انجام می‌شود.

په می‌کنند؟



خروجی دستور زیر چیست؟

```
print(3**2-((8+1)/3+1))
```

در اینجا جواب صحیح عدد ۵ است، اگر داخل یک پرانتز، پرانتزی دیگر باشد پرانتز داخلی اولویت بالاتری دارد، به عبارت دیگر در مثال بالا ابتدا عمل جمع اول (۱+۸) انجام می‌شود بعد نوبت به عمل تقسیم داخل پرانتز می‌رسد و در نهایت عمل جمع دوم داخل پرانتز انجام می‌شود و بدین ترتیب حاصل محاسبه داخل پرانتز بزرگ‌تر ۴ خواهد شد. پس از اتمام محاسبه عبارت داخل پرانتز، عمل توان و در نهایت عمل تفریق انجام می‌شود.

اگر خارج قسمت صحیح و باقی‌مانده تقسیم را لازم داشته باشیم، برای محاسبه خارج قسمت صحیح از علامت // به جای / استفاده می‌کنیم، همچنین از علامت % برای محاسبه باقی‌مانده تقسیم استفاده می‌کنیم.

پاینون	ریاضی
//	خارج قسمت یا تقسیم صحیح
%	باقی‌مانده



کنکاش کن

اولویت عملگرهای // و % را نسبت به بقیه عملگرها پیدا کن.



دست به کار شو

برنامه‌ای بنویس که حاصل عبارت زیر را محاسبه و چاپ کند، با این شرط که در برنامه از کمترین تعداد پرانتز استفاده شود.

$$\frac{11}{2+7} + \frac{3^4}{8-9} \\ 2+9$$

از آنجا که اولویت توان و تقسیم از عملگرهای جمع و تفریق بالاتر است، برای محاسبه صحیح، لازم است عبارات محاسباتی مربوط به مخرج کسرها داخل پرانتز بیاید؛ بنابراین دست کم به چهار پرانتز، به جز پرانتز دستور پرینت نیاز است:

```
print((11/(2+7)+3**4/(8-9))/(2+9))
```

بین دو رشته متنی فقط دو عملگر معنا دارد، یکی عملگر + که نتیجه آن چسباندن دو متن به یکدیگر می‌شود و دومی ضرب یک عدد در متن که نتیجه آن تکرار متن به تعداد عدد است.



خروجی دستور زیر چیست؟

```
print(3*("Comp"+"uter"))
```

جواب:

```
ComputerComputerComputer
```

دستور انتساب

علامت تساوی در زبان‌های برنامه‌نویسی، برخلاف ریاضی، به معنی تساوی نیست، بلکه یک دستور است! به این دستور در برنامه‌نویسی دستور انتساب (یا نسبت‌دادن) یا جایگزینی می‌گویند. پایتون با دیدن این دستور، ابتدا مقدار سمت راست تساوی را محاسبه می‌کند سپس آن مقدار را در متغیر سمت چپ تساوی ذخیره می‌کند. دقت داشته باشید که سمت چپ تساوی همیشه باید نام یک متغیر (یکی از خانه‌های حافظه) وجود داشته باشد. وقتی مقداری را درون متغیری قرار می‌دهیم، به عبارتی (به قول علمای رایانه) متغیر را مقداردهی کرده‌ایم.

قالب دستور:

یک مقدار = نام یک متغیر

دستور مساوی، یک مقدار را در یکی از خانه‌های حافظه (متغیر) ذخیره می‌کند؛ البته جای یک مقدار ممکن است نام متغیر دیگر و یا یک عبارت محاسباتی باشد که حاصل آن پس از محاسبه داخل متغیر قرار می‌گیرد. برای مثال:

```
A=50-7**2+3
```

که در نهایت حاصل عبارت سمت راست تساوی؛ یعنی عدد ۴ داخل متغیر A قرار می‌گیرد. مهم‌ترین ویژگی متغیرها این است که می‌توان مقدار آن‌ها را در طول برنامه عوض کرد. اصلاً به خاطر همین است که به آن‌ها متغیر می‌گویند.

دستور ورودی

مثل بقیه برنامه‌ها، رایانه باید بتواند گاهی مقدار متغیر را از کاربر بپرسد. روش‌های زیادی برای پرسیدن از کاربر وجود دارد. ساده‌ترین آن‌ها در پایتون، دستور input است.

قالب دستور:

نام یک متغیر = input()

دستور input یک مقدار از کاربر می‌گیرد.

رشته یا عدد؟

همان‌طور که قبلاً هم دیدید، هر چه به دستور input بدهیم، از نظر پایتون رشته است. برای گرفتن عدد از کاربر می‌توانیم از دستور int استفاده کنیم.

قالب دستور:

int (یک رشته)

دستور `int` یک رشته (که از عدد صحیح تشکیل شده) را به عدد صحیح تبدیل می‌کند.

```
s = input()
n = int(s)
```

یا به صورت خلاصه می‌توان بنویسیم:

```
n = int(input())
```

دستور `float` که به معنای شناور یا اعشاری است می‌تواند متن را تبدیل به عدد اعشاری کند.

قالب دستور:

```
float ( یک رشته )
```

دستور `float` یک رشته (که از عدد تشکیل شده) را به عدد اعشاری تبدیل می‌کند.

```
x = float(input())
```

اگر در دستور `input` بین دو پرانتز یک رشته وارد شود، نتیجه ترکیبی است از یک `print` (چاپ شدن رشته درون پرانتز) به علاوه اجرای دستور `input` که گرفتن مقدار از ورودی است.

```
s = input("Lotfan yek adad:")
r = float(s)
```

همچنین می‌توانیم دو خط بالا را به صورت زیر با هم ترکیب کنیم:

```
r = float(input("Lotfan yek adad:"))
```

از حالا تا انتهای فصل، سعی کنید ابتدا خودتان برنامه قسمت‌های "دست‌به‌کد شو" را نوشته و روی رایانه اجرا کنید سپس راه‌حل کتاب را با راه‌حل خودتان مقایسه کنید.

اشتباه رایج

نام‌گذاری اشتباه!

نام یک متغیر نمی‌تواند عدد باشد و یا با عدد شروع شود.



دست‌به‌کد شو

برنامه‌ای بنویس که شعاع دایره را از کاربر بپرسد سپس محیط و مساحت دایره را چاپ کند.

یادت نره قرار شد اول خودت برنامه را بنویسی و روی رایانه اجرا کنی و بعد جوابت را با جواب کتاب مقایسه کنی.

```
r = float(input(" لطفا شعاع دایره: "))
print("محیط", 3.14 * r * 2)
print("مساحت", 3.14 **r * 2)
```

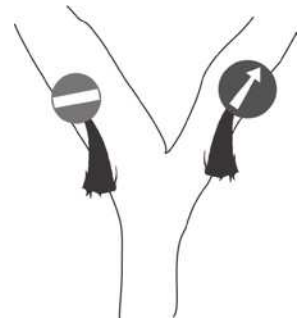


دست‌به‌کد شو

برنامه‌ای بنویس که یک عدد سه رقمی از کاربر بپرسد سپس مجموع یکان و دهگان و صدگان آن را محاسبه کرده و چاپ کند.

```
n = int(input(" لطفا یک عدد سه رقمی وارد کنید: "))
Yekan = n % 10
Dahgan = (n // 10) % 10
Sadgan = n // 100
print(" مجموع ارقام: ", Yekan + Dahgan + Sadgan)
```

در فصل‌های بعدی کتاب (۱) با دستور ایف (if) آشنا شدیم. دستور ایف همانند یک دوراهی، به پایتون دستور می‌دهد که شرط را بررسی کند و تنها در صورت درست بودن شرط، دستورهای زیر آن را (که با فرورفتگی مشخص شده‌اند) اجرا کند، همچنین با استفاده از دستور ال‌اس (else) می‌توان گفت در صورت اشتباه بودن شرط، دستورهای دیگری را اجرا کند.



قالب دستور:

```
if شرط :
    دستورات در صورت برقرار بودن شرط
    ...
else:
    دستورات در صورت برقرار نبودن شرط
    ...
```

عملگرهای مقایسه‌ای

یک شرط، مقایسه بین دو مقدار است که برای آن از عملگرهای مقایسه‌ای استفاده می‌کنیم.

در شرط باید دقت شود که مقایسه بین دو مقدار هم‌جنس انجام شود؛ یعنی فقط عدد با عدد و رشته با رشته مقایسه شود.

دنیای پایتون	دنیای ریاضی
==	=
!=	≠
>	>
<	<
>=	≥
<=	≤

شرط مرکب

زمانی که نیاز به بررسی چند شرط (شرط مرکب یا ترکیبی) داریم، می‌توانیم از چند دستور if که درون یکدیگر قرار گرفته‌اند استفاده کنیم؛ ولی برای خواناتر شدن برنامه بهتر است از عملگرهای منطقی درون یک if استفاده کنیم.

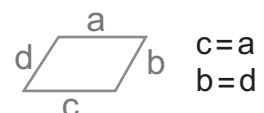
همچنین با استفاده از دستور if و متغیر کمکی، یاد گرفتیم که تعداد چیزی را بشماریم یا بزرگ‌ترین و کوچک‌ترین عدد را بین چند عدد پیدا کنیم. از این روش‌ها در فصل‌های بعدی نیز استفاده کردیم. دستور if یکی از دستورهای اصلی هر زبان برنامه‌نویسی است، چنانچه می‌توان گفت هیچ برنامه رایانه‌ای بدون if وجود ندارد. در هر برنامه‌ای لازم می‌شود بررسی کنیم شرایط چگونه است تا متناسب با آن به رایانه بگوییم چه کارهایی باید انجام دهد. این کار معمولاً با دستور if انجام می‌شود. بسیار مهم است به این دستور تسلط کافی داشته باشید و بدانید در کجا به آن نیاز دارید.

پایتون	محاوره‌ای
and	و
or	یا

برنامه‌ای بنویس که اندازه چهارضلع یک چهارضلعی را به ترتیب بگیرد سپس بگوید آیا این چهارضلعی متوازی‌الاضلاع هست یا نه.



برای حل این سؤال، ابتدا باید بدانید چگونه از روی اندازه اضلاع یک چهارضلعی تشخیص دهید که متوازی‌الاضلاع است یا خیر. در صورتی یک چهارضلعی متوازی‌الاضلاع است که اضلاع روبه‌روی آن برابر باشند، در غیر این صورت متوازی‌الاضلاع نیست.



بنابراین، باتوجه به اینکه اعداد واردشده به ترتیب هستند، کافی است بررسی کنید اعداد اول و سوم واردشده است، همچنین اعداد دوم و چهارم وارد شده با هم برابر باشند.

```
a = float(input(' ضلع اول: '))
b = float(input(' ضلع دوم: '))
c = float(input(' ضلع سوم: '))
d = float(input(' ضلع چهارم: '))
if a==c and b==d:
    print(' متوازی الاضلاع است ')
else:
    print(' متوازی الاضلاع نیست ')
```



دست به کار شو

برنامه‌ای بنویس که یک عدد حداکثر ۵ رقمی از کاربر بگیرد سپس تعداد ارقام آن را چاپ کند.

اگر فصل حلقه شرطی (while) کتاب پایتون (۱) را یادتان باشد، در آن راه‌حل کلی این مسئله برای هر تعداد رقم داده شده بود. در اینجا بدون استفاده از حلقه برای حداکثر ۵ رقم، از راه‌حل دیگری استفاده می‌کنیم.

می‌دانیم اعداد یک رقمی بین ۰ و ۹، اعداد دو رقمی بین ۱۰ و ۹۹، اعداد سه رقمی بین ۱۰۰ و ۹۹۹ و ... می‌باشند؛ بنابراین می‌توان با بررسی بازه‌ای که عدد در آن قرار دارد، تعداد ارقام آن را فهمید.

```
a = int(input(" عدد: "))
if a >= 0 and a <= 9:
    print(1)
if a >= 10 and a <= 99:
    print(2)
if a >= 100 and a <= 999:
    print(3)
if a >= 1000 and a <= 9999:
    print(4)
if a >= 10000 and a <= 99999:
    print(5)
```

در فصل پنجم کتاب پایتون (۱) با لاک‌آشنا شدیم. در این فصل آموختیم که در پایتون کتابخانه‌های زیادی وجود دارد که می‌توان از آن‌ها برای موارد مختلفی استفاده کرد، برای استفاده از این کتابخانه‌ها باید در ابتدای برنامه خود، آن‌ها را به پایتون اضافه کنیم. به صورت زیر:

قالب دستور:

```
import نام کتابخانه
```

دقت کنید که این دستور را باید در ابتدای برنامه خود بنویسیم، همچنین با ایمپورت کردن یک کتابخانه، تمامی دستوره‌های آن، به همان برنامه اضافه می‌شود؛ بنابراین برای استفاده از دستورات یک کتابخانه باید آن را در همان برنامه ایمپورت کنیم. یکی از کتابخانه‌هایی که در کتاب پایتون (۱) نیز به آن اشاره شد، کتابخانه لاک‌ی (turtle) است که برای رسم اشکال و نقاشی‌های ساده از آن استفاده می‌شود. دقت داشته باشیم که هرگز نباید نام برنامه خود را همانند نام یک کتابخانه بگذاریم. در جدول زیر بخشی از دستورات لاک‌ی، کار آن و ساده‌نویسی (مخفف) آمده است:

نام دستور	توضیحات	ساده‌نویسی دستور
turtle.forward(اندازه)	با این دستور لاک‌ی، یک خط صاف رو به جلو، به اندازه‌ای که داده شده است، رسم می‌کند.	turtle.fd (اندازه)
turtle.backward(اندازه)	با این دستور لاک‌ی، یک خط صاف رو به عقب (بدون تغییر جهت نوک پیکان لاک‌ی) به اندازه‌ای که داده شده است، رسم می‌کند.	turtle.bk (اندازه)
turtle.right(اندازه)	با این دستور لاک‌ی، به اندازه زاویه‌ای که داده شده است، در جهت راست نوک پیکان لاک‌ی می‌چرخد.	turtle.rt (اندازه)
turtle.left(اندازه)	با این دستور لاک‌ی، به اندازه زاویه‌ای که داده شده است، در جهت چپ نوک پیکان لاک‌ی، می‌چرخد.	turtle.lt (اندازه)
turtle.penup(اندازه)	با این دستور لاک‌ی مدادش را از روی صفحه بر می‌دارد و در صورت حرکت کردن روی صفحه، اثری از خودش روی صفحه به جا نمی‌گذارد.	turtle.pu ()
turtle.pendown(اندازه)	با این دستور لاک‌ی مدادش را روی صفحه می‌گذارد و در صورت حرکت کردن روی صفحه، از خودش اثر به جا می‌گذارد.	turtle.pd()

برنامه‌ای بنویس که اندازه ضلع یک شش‌ضلعی منتظم را بگیرد و آن را رسم کند.



دست به کار شو

برای حل این سؤال، ابتدا باید بدانیم که اندازه زاویه بین اضلاع یک شش‌ضلعی منتظم چقدر است، همان‌طور که می‌دانیم اندازه زوایای داخلی یک n ضلعی منتظم به صورت زیر به دست می‌آید:

$$\frac{(n-2)*180}{n}$$

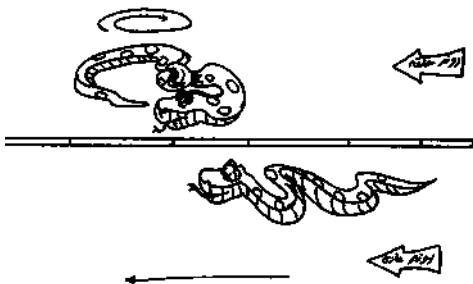
```
import turtle
andaze = int(input('اندازه ضلع: '))
zaviye = ((6 - 2)*180/6 )
turtle.forward( andaze )
turtle.right( zaviye )
turtle.forward( andaze )
turtle.right( zaviye )
turtle.forward( andaze )
turtle.right( zaviye )
turtle.forward( andaze )
turtle.right( zaviye )
turtle.forward( andaze )
turtle.right( zaviye )
turtle.forward( andaze )
turtle.right( zaviye )
turtle.forward( andaze )
turtle.right( zaviye )
```

از اینجا تا آخر کتاب در بخش‌های "چه می‌کنه؟"، اول سعی کن بدون استفاده از رایانه، خروجی برنامه را حدس بزنی، بعد برنامه را در محیط پایتون بنویس و خروجی آن را با حدس خودت مقایسه کن.



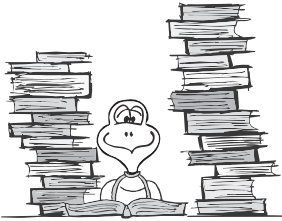
خروجی برنامه زیر چیست؟

```
import turtle
turtle.forward(50)
turtle.right(72)
turtle.forward(50)
turtle.right(72)
turtle.forward(50)
turtle.right(72)
turtle.forward(50)
turtle.right(72)
turtle.forward(50)
turtle.right(72)
```



فصل شش کتاب پایتون (۱) به دستور قدرتمند حلقه فور (for) اختصاص داشت. یکی از مهم‌ترین برتری‌های رایانه نسبت به انسان سرعت محاسبه بالای آن است که با استفاده از دستور for می‌توان به خوبی از این قابلیت استفاده کرد. برای این کار کافی است در پرانتز این دستور بنویسیم که می‌خواهیم دستورهای زیر آن چندبار تکرار شوند، و دستورهایی که قرار است تکرار شوند را با یک فرورفتگی مشخص کنیم.

لغت نامه



واژه علمی	ترجمه	واژه علمی	ترجمه
User Friendly	کاربر پسند	left	چرخش به چپ
import	وارد کن	right	چرخش به راست
turtle	لاک پشت (لاکی)	penup	قلم بالا
random	تصادفی	pendown	قلم پایین
forward	حرکت به جلو	range	محدوده
backward	حرکت به عقب	float	شناور (اعشاری)

جمع بندی کن



..... می‌تواند اولویت‌ها را به هم بزند و برای مثال کاری کند که عمل جمع قبل از توان محاسبه شود.
برای محاسبه خارج قسمت صحیح از علامت به جای / استفاده می‌کنیم، همچنین از علامت برای محاسبه باقی‌مانده تقسیم استفاده می‌کنیم.

تعداد تکرار در حلقه مشخص نیست. در واقع این حلقه تا زمانی تکرار می‌شود که شرط جلوی دستور درست باشد.
جدول برای آزمون درستی برنامه به‌خصوص در مواردی که به رایانه دسترسی ندارید (مانند سر آزمون) بسیار راهگشا است. ستون‌های این جدول متغیرهای برنامه، همچنین برنامه است.

یکی از نشانه‌های یک برنامه (User Friendly)، استفاده مناسب برنامه‌نویس از دستور در چاپ پیام‌های مناسب و به‌جا است.
برای آنکه اعداد و یا عبارت ریاضی برای پایتون قابل فهم باشد نباید آن را در بگذاریم.

بین دو رشته متنی فقط دو عملگر معنا دارد. یکی عملگر که نتیجه آن چسباندن دو متن به یکدیگر می‌شود و دومی در متن که نتیجه آن تکرار متن به تعداد عدد است.
وقتی مقداری را درون متغیری قرار می‌دهیم، به عبارتی متغیر را کرده‌ایم.
دستور یک رشته (که از عدد تشکیل شده) را به عدد اعشاری تبدیل می‌کند.
نام یک متغیر نمی‌تواند باشد و یا با عدد شود.
در شرط باید دقت شود که مقایسه بین دو مقدار انجام شود؛ یعنی عدد فقط با و رشته فقط با مقایسه شود.



سطح مبتدی

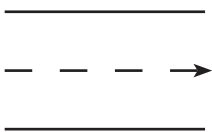
۱. ب.ب.ک چهار عدد از کاربر بگیرد و میانگین آن‌ها را بر روی صفحه بنویسد.
۲. برنامه‌ای برای محاسبه حاصل عبارت زیر بنویسید به طوری که از کمترین تعداد پرانتز در آن استفاده شود.

$$\frac{(2+3)^7}{17} + \frac{4*(6-2)}{18-11}$$

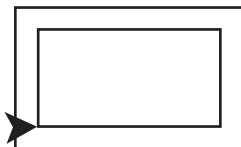
$$\frac{\quad}{1*9-3*5}$$
۳. ب.ب.ک دو عدد از کاربر گرفته در خانه‌های حافظه A و B قرار دهد سپس محتوای این دو متغیر را با هم جابه‌جا کند. (این مسئله چند راه حل دارد).
۴. ب.ب.ک سه عدد از کاربر بگیرد و نمایش دهد که آیا تشکیل مثلث می‌دهند یا خیر.
۵. ب.ب.ک n را از کاربر بگیرد و چاپ کند که آیا تنها بر دو عدد از سه عدد ۲ و ۳ و ۵ بخش پذیر است یا خیر.
۶. ب.ب.ک یک عدد سه رقمی از کاربر بگیرد و نمایش دهد که آیا رقمی تکراری دارد یا خیر.
۷. ب.ب.ک سه عدد از کاربر بگیرد و آن‌ها را به ترتیب از کوچک به بزرگ نمایش دهد.
۸. ب.ب.ک پنج عدد از کاربر بگیرد و کوچک‌ترین آن‌ها را نمایش دهد. (با استفاده از تنها چهار if یک شرطی بدون else و ۵ متغیر).
۹. ب.ب.ک یک عدد سه رقمی از کاربر بگیرد و نمایش دهد که آیا همه ارقامش زوج هستند یا خیر.
۱۰. ب.ب.ک n را از کاربر بگیرد سپس n عدد از کاربر بگیرد و میانگین آن‌ها را نمایش دهد.

سطح متوسط

۱۱. ب.ب.ک اندازه دو ضلع مثلث قائم‌الزاویه را بگیرد و آن را رسم کند.
۱۲. ب.ب.ک لاک‌ی یک خیابان به طول ۲۰۰ را خط‌کشی کند.



۱۳. ب.ب.ک لاک‌ی یک قاب مستطیلی به اضلاع ۱۰۰ و ۲۰۰ بکشد.



۱۴. ب.ب.ک n را از کاربر بگیرد و تمامی اعداد n رقمی که مضرب ۷ هستند را نمایش دهد.
۱۵. ب.ب.ک n را از کاربر بگیرد و نمایش دهد پس از چند مرحله مجموع سری زیر از n بزرگ‌تر می‌شود:

$$A = 1 + 2 + 3 + 4 + 5 + \dots$$

۱۶. کیکاووس تعدادی کارت برای بازی داشته است که روی هر کدام از آنها یک عدد از یک تا n نوشته شده بود. دوست کیکاووس، کیخسرو یکی از کارت‌های کیکاووس را گم کرده است. ب.ب.ک n و شماره کارت‌های موجود را از او بگیرد و شماره کارتی که گم شده است را نمایش دهد.

۱۷. ب.ب.ک n را از کاربر بگیرد سپس n عدد از کاربر بگیرد و در پایان، بزرگ‌ترین و کوچک‌ترین عدد وارد شده را نمایش دهد.

۱۸. ب.ب.ک n را از کاربر بگیرد و بازی هوپ را با کاربر انجام دهد، به‌صورتی که در مضارب n رایانه یا کاربر باید کلمه «هوپ» را بگویند.

۱۹. ب.ب.ک n را از کاربر بگیرد سپس n عدد از کاربر بگیرد و مجموع اعداد مثبت و منفی را جداگانه نمایش دهد.

۲۰. ب.ب.ک تعدادی عدد و علامت را یکی‌درمیان از کاربر بگیرد و وقتی که علامت مساوی وارد شد، نتیجه را نمایش دهد.

۲۱. ب.ب.ک n را از کاربر بگیرد و نمایش دهد که آیا بر مجموع ارقامش بخش‌پذیر هست یا خیر.

۲۲. ب.ب.ک n را از کاربر بگیرد، اگر مجموع ارقام زوج آن از مجموع ارقام فردش کمتر بود، به کاربر «آفرین» بگوید، در غیر این صورت پیام «دوباره تلاش کن» را نمایش دهد.

۲۳. ب.ب.ک یک عدد از کاربر بگیرد و نمایش دهد که عدد قوی هست یا خیر. عدد قوی، عددی است که هر رقم آن از نصف رقم سمت چپ خودش، بیشتر باشد (رقم آخر مورد بررسی قرار نمی‌گیرد).

۲۴. ب.ب.ک n را از کاربر بگیرد و بزرگ‌ترین مربع کاملی که از n کوچک‌تر باشد را نمایش دهد.

۲۵. ب.ب.ک بنویسید که به‌طور تصادفی یک روز از روزهای هفته را انتخاب کرده و نمایش دهد.

سطح پیشرفته

۲۶. ب.ب.ک n را از کاربر گرفته و بگوید آینه‌ای است یا نه. (عددی را آینه‌ای می‌نامیم که از چپ و راست یکسان خوانده شود. مانند: ۳۱۳)

۲۷. ب.ب.ک تمام اعداد سه رقمی که با ارقام ۲ تا ۷ می‌توان نوشت را بنویسید (ارقام تکراری هم می‌تواند در عدد باشد).

۲۸. ب.ب.ک دو عدد گرفته و با استفاده از روش نردبانی ب.م.م (بزرگ‌ترین مقسوم علیه مشترک) آن‌ها را نمایش دهد.

۲۹. ب.ب.ک n و k را از کاربر بگیرد و تعداد اعداد k رقمی‌ای که در n وجود دارد را نمایش دهد.

۳۰. ب.ب.ک n و k را از کاربر بگیرد و بزرگ‌ترین عدد k رقمی‌ای که در n وجود دارد را نمایش دهد.

۳۱. ب.ب.ک n و k را از کاربر بگیرد و بگوید که عدد k چند بار در n تکرار شده است. (هر کدام از n و k می‌توانند هر چند رقمی باشند).

۳۲. ب.ب.ک n را از کاربر بگیرد و مجموع سری‌های زیر را چاپ نماید.

$$1+4+9+16+\dots+n^2 \quad 1+4+9+16+\dots+n^2$$

$$1+\frac{1}{2!}+\frac{1}{3!}+\frac{1}{4!}+\dots+\frac{1}{n!} \quad 1+\frac{1}{2!}+\frac{1}{3!}+\frac{1}{4!}+\dots+\frac{1}{n!}$$

$$1-\frac{1}{3!}+\frac{1}{5!}-\frac{1}{7!}+\dots\pm\frac{1}{(2n-1)!} \quad 1-\frac{1}{3!}+\frac{1}{5!}-\frac{1}{7!}+\dots\pm\frac{1}{(2n-1)!}$$

تمرین‌های فکری (التهاری)



هرگز از بزرگی کارها نهراسید
که کارها هر قدر هم که بزرگ باشند، در مقابل اراده انسان ناچیزند.

امام خمینی (ره)

حل این تمرین‌ها به عزیزی توصیه می‌شود که تمرین‌های یادآوری را پشت سر گذاشته و به دنبال تمرین‌های جدی‌تر برنامه‌نویسی هستند. در حل این مسئله‌ها تنها از دستورات مجاز (دستورات موجود در کتاب پایتون (۱)) حق استفاده دارید.

حدس گلدباخ


در نیمه اول سده هجدهم، ریاضیدان روسی در نامه‌ای به دوست خود، اویلر، مسئله‌ای طرح می‌کند که تاکنون به صورت یک حدس اثبات نشده باقی مانده است. گلدباخ به این مناسبت چنین نوشته است:

«این هم یکی از مسئله‌های من است، یک عدد فرد دلخواه مثلاً ۷۷ را در نظر می‌گیریم، آن را می‌توان به صورت مجموع سه عدد نوشت، به طوری که هر سه اول باشند: $۷۷=۷+۱۷+۵۳$ ، عدد فرد دیگری مانند ۴۶۱ را در نظر بگیرید:

$$۴۶۱=۵+۷+۴۴۹$$

این سه عدد باز هم اول‌اند. اکنون برای من کاملاً روشن است که هر عدد فرد بزرگ‌تر از ۵ را می‌توان به صورت مجموع سه عدد اول نوشت؛ ولی این حکم را چگونه می‌توان ثابت کرد؟ هر آزمایشی درستی حکم را تأیید می‌کند؛ ولی عمر هیچ انسانی امکان آزمایش روی همه عددهای فرد را نمی‌دهد، در اینجا به استدلال کلی نیاز داریم نه به آزمایش. «

اویلر پاسخ داد که این حکم کاملاً درست است؛ ولی او هم نتوانسته است اثبات دقیقی برای آن پیدا کند، در ضمن اویلر حکم دیگری را هم پیشنهاد می‌کند (مسئله اویلر): هر عدد زوج بزرگ‌تر از ۲ را می‌توان به صورت مجموع دو عدد اول نوشت.

حال شما برای کمک به مرحومان گلدباخ و اویلر برنامه‌ای بنویسید که حدس آن‌ها را تحقق بخشد. 

(برای شادروان اویلر یک عدد زوج گرفته و دو عدد اول را که مجموع آن‌ها برابر آن عدد زوج باشد، بنویسد و برای مرحوم گلدباخ یک عدد فرد گرفته و سه عدد اول را که مجموع آن‌ها برابر آن عدد فرد باشد، بنویسد).

تجزیه به عوامل اول

برنامه‌ای بنویسید که یک عدد گرفته، آن را به عوامل اول تجزیه کرده بنویسد.

زیرمجموعه‌ها

برنامه‌ای بنویسید که عدد n را گرفته و تمام زیرمجموعه‌های مجموعه $\{1, 2, \dots, n\}$ را بنویسد. برای مثال، اگر $n=2$ ، داده شود باید بنویسید:

{ }

{ ۱ }

{ ۲ }

{ ۱ و ۲ }

(راهنمایی: می‌توان ثابت کرد تعداد زیرمجموعه‌های یک مجموعه n عضوی 2^n می‌باشد).