



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی صنایع

به نام خدا

نام استاد: دکتر بهروز کریمی

برنامه ریزی و کنترل موجودی ۱

تدریسار: پدرام پیرو اصفیا

لطفا قبل از بارگزاری جوابها در سامانه، به نکات زیر توجه بفرمایید:

- تمامی جوابها را در فایل ژوپیتر داده شده بنویسید.
- تمامی کدهای هر سوال در تابع مشخص شده برای هر سوال بنویسید و خارج از آن چیزی ننویسید.
- برای برگرداندن جواب درون تابع از دستور return استفاده نمایید و جواب را با دستور print چاپ نکنید، چرا که در فرآیند ارزیابی دچار اشکال خواهد شد.

۱. تابعی به اسم answer_one بنویسید، که در این تابع به شما جمع و ضرب دو عدد طبیعی به عنوان ورودی داده شده و اعدادی که در این شرایط صادق هستند، بعد از مرتب شدن به ترتیب صعودی، باید به عنوان خروجی در یک لیست برگردانده شوند. همچنین ورودی اول تابع، حاصل جمع و ورودی دوم حاصل ضرب است.

مثال:

Input: 7,12

Calling the function: answer_one(7,12)

Output: [3,4]

۲. تابعی به اسم answer_two بنویسید، در این تابع یک ماتریس مربعی که سطرهای آن کنار هم قرار گرفته است درون یک لیست به شما داده شده و چیزی که شما باید محاسبه کنید، حاصل جمع مقادیر روی قطر اصلی است. این تابع یک ورودی به فرمت لیست دارد.

به طور مثال ماتریس مربعی $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ به صورت [1,2,3,4,5,6,7,8,9] ورودی داده خواهد شد و باید عدد ۱۵ خروجی داده شود.

مثال:

Input: [1,2,3,4,5,6,7,8,9]

Calling the function: answer_two([1,2,3,4,5,6,7,8,9])

Output: 15

۳. تابعی به اسم answer_three بنویسید، در این تابع یک عدد رندوم با استفاده از کتابخانه‌ی random بین دو عدد داده شده به عنوان ورودی، خروجی می‌دهید که: (۱) عدد رندوم اول باشد، (۲) فرد باشد.

مثال:

Input: 1,100

باید یک عدد رندوم بین یک تا ۱۰۰ برگردانده شود که هم فرد باشد هم اول.

Calling the function: answer_three(1,100)

Output: 11

خروجی‌های دیگری نیز میتواند انتخاب شود که ماهیتش تصادفی است.

راهنمایی: برای تولید عدد رندوم بین دو عدد دلخواه داده شده (که عدد اول بازه، بسته و عدد دوم باز است)، از این [طریق](#) عمل کنید.

۴. تابعی به اسم answer_four بنویسید، در این تابع یک عدد به شما داده شده و خروجی شما باید ریشه‌ی دوم این عدد باشد. ولی حق استفاده از به توان ۰.۵ رساندن یا استفاده از توابع آماده در پایتون را ندارید. به جای آن از الگوریتمی که در زیر توضیح داده میشود استفاده نمایید.

الگوریتم به صورت زیر کار میکند:

$$x_0 = 1$$

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{\text{number}}{x_n} \right)$$

$$\sqrt{\text{number}} = \lim_{n \rightarrow \infty} x_n$$

برای شفاف شدن الگوریتم، فرض کنید میخواهید ریشه دوم عدد ۵ را حساب نمایید:

$$x_0 = 1$$

$$x_1 = \frac{1}{2} \left(1 + \frac{5}{1} \right) = 3$$

$$x_2 = \frac{1}{2} \left(3 + \frac{5}{3} \right) = \frac{7}{3}$$

$$x_3 = \frac{1}{2} \left(\frac{7}{3} + \frac{15}{7} \right) = \frac{47}{21} \dots$$

برای رسیدن به جواب دقیق باید در تئوری بینهایت بار این کار را تکرار کرد، اما شما در کد خود، صرفا به اندازه ۱۰۰ تکرار این کار را انجام دهید و خروجی را با دقت ۳ رقم اعشار و گرد شده برگردانید. دقت کنید که اگر عدد ورودی حتما بزرگتر از ۰ است.

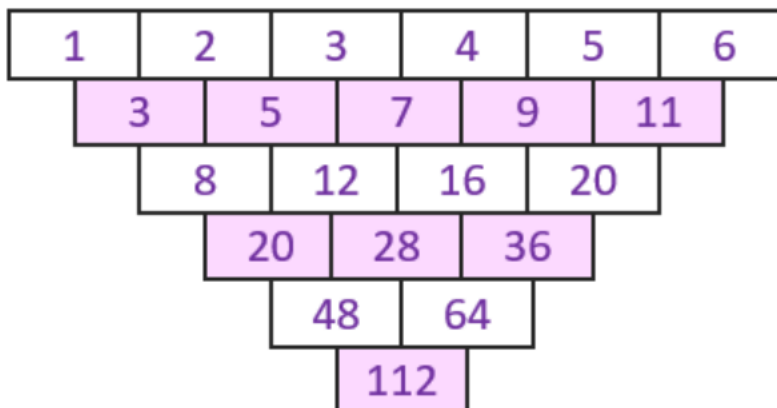
مثال:

Input: 5

Calling the function: answer_four(5)

Output: 2.236

۵. تابعی به اسم answer_five بنویسید، در این تابع شما باید پازل زیر را حل کنید:



بدین شکل که ردیف بالا، به عنوان ورودی تابع به شما داده میشود، و هر کدام از باکس های زیر دو باکس بالایی، حاصل جمع دو باکس بالایی است، به طور مثال:

1	+	2	=	3
---	---	---	---	---

مثال:

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

Calling the function: answer_five([1,2,3,4,5,6])

Output: 112

۶. (این سوال امتیازی است) تابعی به اسم answer_six بنویسید، در این تابع شما باید شناسایی کنید که آیا یک شماره کارت که به شما داده میشود، معتبر است یا خیر، ورودی تابع شماره کارت است و خروجی کد شما اگر شماره کارت معتبر بود، رشته‌ی valid (string) و اگر نبود invalid خواهد بود.



الگوریتم استفاده شده به شرح زیر است:

شماره کارت بدین شکل است:

4000123456789010 →

4	0	0	0	1	2	3	4	5	6	7	8	9	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

گام اول باید تمام اعدادی که در جایگاه فرد قرار دارند را ضرب در دو کنیم:

8	0	0	0	2	2	6	4	10	6	14	8	18	0	2	0
---	---	---	---	---	---	---	---	----	---	----	---	----	---	---	---

گام بعد اگر اعداد حاصل شده از ضرب کردن، بزرگتر از ۹ شدند، ارقام آن عدد باید با هم جمع شوند:

8	0	0	0	2	2	6	4	1	6	5	8	9	0	2	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

در نهایت تمامی اعداد با هم جمع میشوند:

$$8+0+0+0+2+2+6+4+1+6+5+8+9+0+2+0=53$$

اگر عدد جمع شده (در مثال ما ۵۳) رقم آخرش ۰ باشد، شماره کارت معتبر است، در غیر این صورت نامعتبر میباشد. در مثال ما شماره کارت نامعتبر است.

مثال:

Input: 4000123456789010

Calling the function: answer_six(4000123456789010)

Output: "invalid"

راهنمایی: برای تبدیل یک عدد به لیستی از ارقامش، از [این لینک](#) استفاده کنید.