

به نام خداوند علم و دانش



دانشکده مهندسی کامپیوتر

برنامه نویسی پیشرفته (پایتون)

تمرین اول (پایتون مقدماتی)

دکتر مرضیه داودآبادی

زمستان 1403

طراحان تمرین: آقایان امین شیروانی و محمدصادق همتی

- در صورت وجود هرگونه ابهام به طراح پیام دهید.
 - باتوجه به وجود تاخیر 10 روزه، امکان تاخیر تحت هیچ شرایطی امکان پذیر نیست.
 - انجام تمرین ها تک نفره می باشد.
 - زبان برنامه نویسی پایتون است.
 - موارد ارسال شده به صورت آنلاین تحویل گرفته خواهند شد.
 - ددلاین تمرین: 18 اسفند ساعت 23:59
 - برای دیدن تست کیس های نمونه متنوع به کوئرای درس بپیوندید.
 - [/https://quera.org/course/add_to_course/course/20444](https://quera.org/course/add_to_course/course/20444)
 - Password: ap4032
 - لینک تلگرام طراحان:
 - <https://telegram.me/OversealMSH> | آقای محمدصادق همتی
 - <https://telegram.me/aminkte> | آقای امین شیروانی
-
-

01. اعداد خاص (شیروانی)

امین و صادق دو دوست قدیمی هستند، اما وقتی پای اعداد خاص وسط می آید، این دو تبدیل به رقبای سرسخت می شوند!

امین تعصب شدیدی روی اعداد اول دارد و معتقد است که فقط این اعداد ارزش بررسی دارند. از طرف دیگر، صادق عاشق اعداد فیبوناچی است و می گوید که هیچ عددی به اندازه ی فیبوناچی ها جذاب و پر رمز و راز نیست!

حالا این دو تصمیم گرفته اند که با هم همکاری (یا شاید رقابت!) کنند و لیستی از اعدادی که هم اول باشند و هم در دنباله فیبوناچی قرار داشته باشند تهیه کنند. اما داستان اینجا تمام نمی شود! علی که همیشه در حال نظارت بر این دو نفر است، یک چالش جدید مطرح کرده: اگر عدد m که او انتخاب کرده، در موقعیت فردی این لیست باشد (مثلاً عضو اول، سوم، پنجم و ...)، پس Yes چاپ شود. در غیر این صورت، علی دست هایش را به نشانه ی ناامیدی بالا می برد و No اعلام می کند!

ورودی

دو عدد صحیح n و m که تعداد حداکثری برای بررسی اعداد و عدد مورد نظر علی را مشخص می کنند.

خروجی

لیستی از اعداد خاص امین و صادق، سپس بررسی اینکه m در جایگاه فردی قرار دارد یا نه. اگر عدد m همزمان نه اول بود و نه فییبوناچی عبارت $invalid$ چاپ شود.

02. تقسیم سکه (شیروانی)

امین n سکه دارد و x دوست صمیمی (یا شاید صمیمی سابق!) که می خواهد بین آن ها این سکه ها را تقسیم کند. مشکل اینجا است که اگر دوستانش بفهمند که کسی بیشتر یا کمتر گرفته، ممکن است رفاقت شان تا اطلاع ثانوی لغو شود! پس امین تصمیم گرفته که تمام حالت های ممکن برای تقسیم این سکه ها را بررسی کند تا ببیند آیا می تواند راهی پیدا کند که همه راضی باشند. برنامه ای بنویسید که دو عدد n و x را از ورودی دریافت کرده و تمام روش های ممکن برای توزیع این سکه ها را پیدا کند. (البته اینکه دوستان امین در نهایت راضی باشند، دیگر مشکل ما نیست!)

ورودی

دو عدد صحیح n و x که تعداد سکه ها و تعداد دوستان امین را مشخص می کنند.

خروجی

همه روش های ممکن برای تقسیم این سکه ها، به طوری که مجموع سکه های داده شده همیشه n باشد. توجه کنید که حالت های تقسیم را از کوچک به بزرگ چاپ کنید.

03. عیول دنباله بازی (همتی)

ورودی ها:

برنامه از ورودی استاندارد دریافت می کند که شامل موارد زیر است:

- نوع دنباله (arithmetic, geometric, fibonacci, factorial)
 - تعداد اعضای دنباله (n)
 - مقدار اولیه (a)
 - در دنباله های حسابی و هندسی، مقدار d یا r نیز دریافت می شود.
- ** عدد اعشاری نداریم!**

خروجی ها

برنامه دنباله مورد نظر را محاسبه کرده و به عنوان خروجی نمایش می دهد.

مدیریت خطاها

برنامه ورودی ها را بررسی کرده و در صورت نامعتبر بودن ورودی، پیام های مناسبی نمایش می دهد:

اگر ورودی ها کمتر از ۳ مقدار داشته باشند:

Invalid input: Not enough parameters provided.

اگر مقدار n خارج از بازه $[1, 1000]$ باشد:

Invalid input: 'n' must be between 1 and 1000.

اگر مقدار a خارج از بازه $[-10^6, 10^6]$ باشد:

Invalid input: 'a' is out of range.

اگر دنباله از نوع arithmetic باشد ولی مقدار d داده نشده باشد

Invalid input: missing common difference for arithmetic sequence.

اگر دنباله از نوع geometric باشد ولی مقدار r داده نشده باشد

Invalid input: missing ratio for geometric sequence.

اگر نوع دنباله وارد شده نامعتبر باشد

Invalid sequence type.

04. کارت اونور آبی (همتی)

ورودی ها:

برنامه شماره کارت را به عنوان ورودی دریافت می کند و بررسی می کند که آیا معتبر است یا خیر.



خروجی ها

برنامه نوع کارت را بر اساس شماره ورودی مشخص می کند و یکی از موارد زیر را نمایش می دهد:

VISA •

MASTERCARD •

AMEX •

INVALID • (در صورت نامعتبر بودن کارت)

برنامه دنباله مورد نظر را محاسبه کرده و به عنوان خروجی نمایش می دهد.

مدیریت خطاها

برنامه ورودی ها را بررسی کرده و در صورت نامعتبر بودن ورودی، پیام های مناسبی نمایش

می دهد:

اگر ورودی شامل کاراکترهای غیر عددی باشد

INVALID

اگر طول شماره کارت معتبر نباشد (۱۳، ۱۵ یا ۱۶ رقم نباشد):

INVALID

اگر شماره کارت الگوریتم Luhn را پاس نکند:

INVALID

الگوریتم بررسی شماره کارت:

1. بررسی می‌شود که شماره کارت فقط شامل اعداد باشد.
2. بررسی می‌شود که طول شماره کارت یکی از مقادیر مجاز باشد.
3. برای بررسی صحت کارت
 - از رقم یکی مانده به آخر شروع کرده و هر عدد دوم را در ۲ ضرب کنید.
 - اگر حاصل ضرب دو رقمی شد، مجموع ارقام آن را محاسبه کنید.
 - تمامی اعداد را (شامل اعدادی که ضرب نشده‌اند) با هم جمع کنید.
 - اگر مجموع بر ۱۰ بخش پذیر بود، شماره کارت معتبر است.
4. بررسی نوع کارت بر اساس شماره وارد شده:
 - اگر شماره کارت ۱۵ رقمی باشد و با 34 یا 37 شروع شود ← AMEX
 - اگر شماره کارت ۱۶ رقمی باشد و با 51 تا 55 شروع شود ← MASTERCARD
 - اگر شماره کارت ۱۳ یا ۱۶ رقمی باشد و با 4 شروع شود ← VISA
 - در غیر این صورت ← INVALID

05. شاباش حریصانه (همتی)

در روزی روزگاری، در یک جشن عروسی شاد و پرهیاهو، مهمانان مشغول شاباش دادن به داماد بودند. هر کسی سکه‌ای به هوا می‌انداخت و داماد با ذوق و شوق آن‌ها را جمع می‌کرد. اما یک مشکل بزرگ وجود داشت! داماد نمی‌دانست چطور با کمترین تعداد سکه، مبلغی که از مهمانان دریافت کرده را به عروس بدهد. سکه‌ها ۲۵ و ۱۰ و ۵ و ۱ سنتی بودند.

ناگهان، پیرمرد دانایی که همیشه گوشه‌ی مجلس می‌نشست و چای می‌نوشید، جلو آمد و گفت:

❖ پسر! اگر می‌خواهی سریع‌تر و راحت‌تر حساب و کتاب کنی، از روش حریصانه استفاده

کن.

داماد که همیشه در ریاضی ضعیف بود، لبخندی زد و از شما که مهمان ویژه او بودید خواست که برنامه‌ای برایش بنویسند تا این کار را به‌صورت خودکار انجام دهد! برنامه‌ای که مبلغ ورودی را دریافت کند، بررسی کند که عدد صحیح است و سپس با کمترین تعداد سکه، آن را پرداخت کند.

و این‌گونه بود که داماد، نه‌تنها عروسی‌اش را به‌خوبی برگزار کرد، بلکه برنامه‌ای برای

مدیریت شاباش‌های حریصانه ساخت و در تمام شهر مشهور شد!

برنامه مقدار پول ورودی را (به سنت) دریافت کرده و کمترین تعداد سکه‌های مورد نیاز برای پرداخت آن را محاسبه می‌کند. سکه‌های موجود شامل ۱ سنتی، ۵ سنتی، ۱۰ سنتی و ۲۵ سنتی هستند. برنامه باید بررسی کند که ورودی معتبر است و در غیر این صورت، پیام خطا نمایش دهد.

ویژگی‌های برنامه

دریافت مقدار پول از کاربر

بررسی معتبر بودن ورودی (عددی و غیرمنفی بودن)

استفاده از کمترین تعداد سکه ممکن برای پرداخت مقدار ورودی

نمایش تعداد سکه‌های مورد نیاز

محدودیت‌ها و شرایط اعتبارسنجی ورودی‌ها

اعتبارسنجی مقدار ورودی

- ورودی باید عدد صحیح باشد. در غیر این صورت، پیام **error!** نمایش داده شود.
- مقدار ورودی نباید خالی باشد.
- مقدار ورودی نباید منفی باشد.

الگوریتم حل مسئله

1. مقدار ورودی را از کاربر دریافت کنید.

۲. بررسی کنید که مقدار ورودی عدد صحیح مثبت باشد.

۳. مقدار ورودی را به سکه‌های ۲۵، ۱۰، ۵ و ۱ سنتی تقسیم کنید، به طوری

که کمترین تعداد سکه ممکن استفاده شود.

۴. مقدار باقی‌مانده را محاسبه کرده و مراحل را تکرار کنید تا مقدار ورودی به

صفر برسد.

۵. تعداد سکه‌های مورد استفاده را نمایش دهید.

ورودی

1 | 87

خروجی

1 | 6

توضیح: 87 سنت به این صورت پرداخت می‌شود:

۳ سکه ۲۵ سنتی

۱ سکه ۱۰ سنتی

۰ سکه ۵ سنتی

۲ سکه ۱ سنتی

مجموع: ۶ سکه

06. اولین گام مدار منطقی (همتی)

سوشیانت که به تازگی درس مدار منطقی را برداشته، می‌ترسد که نمره کمی بگیرد. او از استاد می‌پرسد که چگونه می‌تواند نمره اضافه داشته باشد. استاد به او می‌گوید که یک برنامه تبدیل مبنا بنویسد تا اعداد را از مبنای ۲ تا ۱۶ به یکدیگر تبدیل کند. مثلاً مبنای ۲ را به ۱۶ تبدیل کند یا مبنای ۷ را به ۳ تبدیل کند و...

دریافت ورودی‌ها

برنامه ابتدا ورودی‌ها را دریافت کرده و به صورت یک لیست از رشته‌ها ذخیره می‌کند. سپس بررسی می‌شود که:

- تعداد ورودی‌ها دقیقاً ۳ باشد.
- دو ورودی اول باید اعداد صحیح باشند.

بررسی مقادیر ورودی

1. مقدار اولین ورودی (مبنای ورودی) باید بین ۲ تا ۱۶ باشد.
2. مقدار دومین ورودی (مبنای خروجی) باید بین ۲ تا ۱۶ باشد.
3. مقدار سومین ورودی (عدد مورد نظر) باید متناسب با مبنای ورودی معتبر باشد.

تبدیل عدد

بعد از اعتبارسنجی ورودی‌ها، ابتدا عدد ورودی به مبنای ۱۰ تبدیل می‌شود. سپس مقدار به دست آمده مجدداً به مبنای خروجی تبدیل شده و نمایش داده می‌شود.

خطاهای ممکن و محدودیت‌ها

این برنامه دارای برخی محدودیت‌ها و کنترل‌های خطا است که شامل موارد زیر می‌شود:

1. فرمت نادرست ورودی‌ها

• اگر تعداد ورودی‌ها کمتر یا بیشتر از ۳ باشد، پیام **Error: Input base and output**

base must be numbers between 2 and 16! نمایش داده می‌شود.

• اگر دو مقدار اول اعداد صحیح نباشند، پیام **Error: Input base and output**

base must be numbers between 2 and 16! نمایش داده می‌شود.

2. مقادیر نامعتبر برای مبناها

• اگر مقدار مبناها خارج از محدوده ۲ تا ۱۶ باشد، پیام **Error: Base must be**

between 2 and 16! نمایش داده می شود.

• اگر عدد ورودی شامل کاراکترهایی باشد که در مبنای ورودی معتبر نیستند، پیام **Error:**

The number is not valid for the given input base! نمایش داده می شود.

07. کارت ویزیت سوشیانت(همتی)

سوشیانت که به تازگی کسب و کار خود را راه اندازی کرده برای معرفی کسب و کار خود به

دیگران نیاز به یک کارت ویزیت دارد. او به شما که یک برنامه نویس هستید میسپرد تا یک

برنامه طراحی کارت ویزیت بنویسید به این صورت که:

دریافت ورودی ها

برنامه ابتدا ورودی ها را دریافت کرده و به صورت یک لیست از رشته ها ذخیره می کند. سپس

بررسی می شود که:

• تعداد ورودی ها دقیقاً ۴ باشد.

• تمام ورودی ها عدد صحیح باشند.

بررسی مقادیر ورودی

1. مقدار اولین ورودی (انتخاب طرح) باید بین ۱ تا ۳ باشد.
2. مقدار دومین ورودی (عرض) نباید کمتر از ۲۰ باشد.
3. مقدار سومین ورودی (ارتفاع) نباید کمتر از ۱۰ باشد.
4. مقدار چهارمین ورودی (تعداد اطلاعات) نباید از ارتفاع - 2 بیشتر باشد و باید بیشتر از صفر باشد.

دریافت اطلاعات متنی

بعد از اعتبارسنجی ورودی‌ها، به مقدار چهارمین ورودی خط اطلاعات متنی از کاربر دریافت می‌شود. مثلاً اگر مقدار آن برابر ۵ باشد، ۵ خط از کاربر ورودی می‌گیرد.

محاسبه فاصله‌ها

برای نمایش در وسط صفحه به صورت افقی از $\text{center}(\text{width}-2)$ استفاده می‌کنیم. برای عمودی نیز از فاصله از بالا به فرمول $(\text{ارتفاع} - \text{تعداد اطلاعات}) / 2$ است و برای پایین نیز ارتفاع - چهارمین ورودی (تعداد اطلاعات) - فاصله از بالا هست.

طراحی کارت

سه نوع طرح برای کارت قابل انتخاب است:

1. طرح ستاره‌ای (*****)

2. طرح خط‌کشی (**~**+**)

3. طرح موجی (**/*, **~, ***)

هر طرح با استفاده از عرض و ارتفاع مشخص شده رسم می‌شود. متن‌های ورودی به صورت وسط‌چین داخل جعبه قرار می‌گیرند و خطوط خالی مورد نیاز برای تنظیم ارتفاع متناسب با اطلاعات درج می‌شوند.

خطاهای ممکن و محدودیت‌ها

این برنامه دارای برخی محدودیت‌ها و کنترل‌های خطا است که شامل موارد زیر می‌شود:

۱. فرمت نادرست ورودی‌ها

- اگر تعداد ورودی‌ها کمتر یا بیشتر از ۴ باشد، پیام **Invalid input format!** نمایش

داده می‌شود.

- اگر ورودی‌ها شامل عدد صحیح نباشند، پیام **Width or height not number!**

نمایش داده می‌شود.

۲. مقادیر نامعتبر برای طراحی و ابعاد جعبه

- اگر مقدار انتخاب طرح خارج از محدوده ۱ تا ۳ باشد، پیام **Invalid design choice!** نمایش داده می شود.
- اگر طول کمتر از ۲۰ یا height کمتر از ۱۰ باشد، پیام **Invalid width or height!** نمایش داده می شود.
- اگر طول کمتر از ارتفاع باشد، پیام **Height cannot be greater than width!** نمایش داده می شود.
- محدودیت های تعداد خطوط اطلاعاتی
- اگر مقدار تعداد اطلاعات تعداد اطلاعات بیشتر از ارتفاع - ۲ باشد، پیام **Number of information exceeds!** نمایش داده می شود.
- اگر مقدار تعداد اطلاعات کمتر یا مساوی صفر باشد، پیام **Information number must be greater than zero!** نمایش داده می شود.

08. ک.م.م بازگشتی (شیروانی)

چالش ریاضیاتی برای مغزهای آماده پردازش!



برنامه‌نویسان عزیز، یک مأموریت فوق‌العاده برای شما داریم! تصور کنید عددی را به شما می‌دهند و از شما می‌خواهند تمام حالت‌های جایجایی ارقامش را پیدا کنید. اما این تازه اول ماجراست... بعد از اینکه این اعداد را استخراج کردید، باید کمترین مضرب مشترک (ک.م.م) آن‌ها را بدون استفاده از هیچ کتابخانه‌ای حساب کنید! بله، بدون کلک، بدون تقلب، و بدون توابع آماده!

ورودی

در ورودی عدد طبیعی n به شما داده میشود.

خروجی

ک.م.م همه ی جایگشت های ممکن آن عدد را محاسبه کنید.

09. جدول خیام-پاسکال (شیروانی)

جادوی خیام-پاسکال در دنیای برنامه‌نویسی!

شرح مسئله: تصور کنید در میان کتاب‌های قدیمی ریاضی، نوشته‌های خیام و پاسکال را ورق می‌زنید. آن‌ها در قرن‌های مختلف، اما با ذهن‌هایی درخشان، به یک حقیقت شگفت‌انگیز رسیدند: مثلث خیام-پاسکال! و حالا، وظیفه شما به‌عنوان یک برنامه‌نویس این است که این جادو را در دنیای صفر و یک‌ها زنده کنید!



وظیفه شما برنامه‌ای بنویسید که عدد n را از ورودی دریافت کند و مقدار $(a + b)^n$ را به سبک مثلث خیام-پاسکال بسط دهد! اما به جای محاسبه مستقیم، از قوانین زیبای ترکیبیات استفاده کنید.

ورودی

عدد حسابی n در ورودی به شما داده میشود.

خروجی

حاصل جبری $(a + b)^n$ را محاسبه کنید.

10. مهمانی امین (شیروانی)

****مهمانی امین و چالش دوستان مشترک**** 🎉 !

امین بعد از مدت‌ها به یک مهمانی دوستانه دعوت شده بود. جمعی از دوستان قدیمی و جدید دور هم جمع شده بودند، اما یک چیز ذهنش را درگیر کرده بود.

او مدام می‌دید که بعضی‌ها بیشتر با هم حرف می‌زنند و به نظر می‌رسد دوستان مشترکی دارند، اما دقیقاً چقدر؟ چقدر احتمال دارد که دو نفر در این جمع، دوستان مشترک داشته باشند؟

امین که علاقه زیادی به تجزیه و تحلیل روابط دارد، سریع دفترچه یادداشتش را برداشت و

شروع به نوشتن کرد:

- هر نفر را به عنوان یک کلید در نظر می گیریم.
- لیستی از دوستان هر نفر را به عنوان مقدار ذخیره می کنیم.
- حالا باید بفهمیم که هر جفت نفرات در این مهمانی چند دوست مشترک دارند. این کار به عهده شماست.

ورودی

در خط اول عدد n که نشان دهنده ی افراد حاضر در مهمانی است. در n خط بعدی ابتدا شماره ی فرد و سپس شماره افرادی که دوست او هستند وارد میشود.

خروجی

به ازای هر دو فرد باید تعداد دوستان مشترک او را بنویسید