مبانی برنامه‌سازی پایتون

نیم‌سال دوم ۹۲-۹۳

دانشکده‌ی مهندسی کامپیوتر — مدرس: بی‌نظیر بیگلری

تمرین دو — بازگشتی، لیست و محاسبات اعشاری — مهلت ارسال: ۲۵ فروردین

به موارد زیر توجه کنید:

- حتما در ارسال فایل برنامه‌ها به نام فایل و فرمت ورودی و خروجی‌ها توجه شود. هیچ‌گونه کوتاهی و بی دقتی پذیرفته نخواهد بود.
  دقت کنید ورودی و خروجی مطابق با صورت گفته شده در تمرین باشد در غیر این صورت به دلیل تصحیح خودکار تمارین نمره‌ای به شما تعلق نخواهد گرفت.
- فایل ها در سامانه جاج بارگذاری خواهند شد که آدرس و نحوه ی کارکرد با آن در $piazza$ اعلام خواهد شد.
- با هرگونه تقلب برخورد جدی خواهد شد. فرد خاطی ۱۰۰− نمره به ازای همان تمرین دریافت خواهد کرد و در صورت تکرار برخوردهای شدیدتری را در پی خواهد داشت.
- هرگونه سوال مربوط به تمرین ها را با موضوع مناسب در صفحه درس در سایت $piazza$ مطرح کنید.
- با هرگونه تلاش در جهت دسترسی غیرقانونی به سرور جاج برخورد جدی خواهد شد.
- کامنت گذاری و نام‌گذاری مناسب متغیرها الزامی‌ست و بخشی از نمره‌ی شما را در تحویل حضوری مشخص می‌کند.

گروه تمرین

سؤال ۱. ب.م.م(۱۵امتیاز) $(gcd.py)$

برنامه‌ای بنویسید که دو عدد از ورودی بگیرید و با استفاده از یک تابع بازگشتی بزرگترین مقسوم علیه مشترک آن دو عدد را محاسبه و چاپ کند.

نمونه ورودی

```
14
21
```

نمونه خروجی

```
7
```

سؤال ۲. ماتریس معادل (۳۰ امتیاز) $(matrix.py)$

فرض کنید ماتریس A، فقط K درایه غیر صفردارد. ماتریس B با ابعاد $K \times ۳$ را ماتریس معادل A گوییم هرگاه هر سطر آن به ترتیب شامل شماره سطر شماره ستون و مقدار یکی از عناصر غیرصفر A باشد یعنی سطرهای آن به ترتیب شامل شماره سطر شماره ستون و مقادیر عناصر غیرصفر A باشد. در زیر چند مثال آمده است
:

$$A = \begin{bmatrix} ۰ & ۵۶ \end{bmatrix} \Rightarrow B = \begin{bmatrix} ۱ & ۲ & ۵۶ \end{bmatrix}$$

$$A = \begin{bmatrix} ۰ & ۱۷ & ۰ \\ ۲۴ & ۰ & ۱۲ \end{bmatrix} \Rightarrow B = \begin{bmatrix} ۱ & ۲ & ۱۷ \\ ۲ & ۱ & ۲۴ \\ ۲ & ۳ & ۱۲ \end{bmatrix}$$

$$A = \begin{bmatrix} ۰ & ۰ & ۸ & ۰ & ۷ \\ ۰ & ۰ & ۰ & ۲۲ & ۰ \\ ۰ & ۱۴ & ۰ & ۰ & ۱۱ \\ ۱۳ & ۰ & ۰ & ۰ & ۰ \end{bmatrix} \Rightarrow B = \begin{bmatrix} ۱ & ۳ & ۸ \\ ۱ & ۵ & ۷ \\ ۲ & ۴ & ۲۲ \\ ۳ & ۲ & ۱۴ \\ ۳ & ۵ & ۱۱ \\ ۴ & ۱ & ۱۳ \end{bmatrix}$$

دقت کنید که ابتدا تمام عناصر سطر اول A را به ترتیب وارد B کردیم. سپس به سطرهای بعدی A رفتیم پس به صورت سطری ماتریس A را بررسی کنید. برنامه‌ای بنویسید که ابعاد (ابتدا تعداد سطرها و خط بعدی تعداد ستون‌ها) و اعضای ماتریس A را به صورت سطر به سطر و در قالب یک لیست بگیرد(ابتدا تمام اعضای سطر یک سپس سطر دو و ...)، سپس ماتریس معادل آن را به شکل ماتریسی چاپ کند.

نمونه ورودی:

```
2
3
0 17 0 24 0 12
```

نمونه خروجی:

```
1 2 17
2 1 24
2 3 12
```

سؤال ۳. محاسبه‌ی ریشه (۳۰ امتیاز)$(root.py)$

روش نصف کردن اولین و ساده ترین روش برای پیدا کردن صفرهای تابع است ، که البته معایب و محدودیتهایی دارد.این روش برای توابعی قابل اجراست که حول ریشه خود اکیدا یکنوا باشند. به عبارت دیگر این روش تنها برای پیدا کردن ریشه های ساده قابل استفاده است ، و قادر به یافتن ریشه های مضاعف نیست. در ضمن سرعت همگرایی آن بسیار کند است و به همین دلیل اغلب برای محاسبه صفرهای توابع چند جمله ای (معادلات ساده) استفاده می شود.
در این روش یک رابطه f به همراه یک بازه به صورت $[a, b]$ داده می‌شود و وسط بازه (c)(میانگین دو سر بازه) یافت می‌شود .اکنون دو بازه‌ی جدید داریم برای ادامه‌ی محاسبه باید یکی از این دو بازه که ریشه در آن قرار دارد را انتخاب کنیم. مقادیر $f(c)$ و $f(a)$ و $f(b)$ را به دست می آوریم اگر $f(a).f(b) < ۰$ آنگاه بازه‌ی جدید $[a, c]$ خواهد بود در غیر این صورت بازه‌ی $[c, b]$ انتخاب می‌گرددو عملیات روی بازه‌ی جدید ادامه‌می‌یابد.این کار تا زمانی که خطای دوسر بازه بیشتر از خطای داده شده در ورودی است ادامه مییابد زمانی که به عملیات متوقف شود مقدار c در بازه‌ی آخر را به عنوان ریشه باز می‌گردانیم.

محاسه‌ی خطا از این روش خواهد بود:

$$error = \frac{|a_i - b_i|}{۲}$$

در این سوال می‌خواهیم این روش را برای توابع چند جمله‌ای پیاده‌سازی کنیم.شما می‌بایست با دریافت یک معادله‌ی چند جمله‌ای ، یک بازه ، یک حداکثر خطا و یک p به عنوان تعداد ارقام اعشار ریشه‌ی معادله‌ی داده شده در آن بازه را محاسبه کنید و جواب نهایی را با p رقم اعشار چاپ کنید.
فرض کنید تمامی توابع داده شده به شما در بازه‌ی مورد نظر جواب دارند.
برای این سوال شما می‌بایست این دقیقا این تابع را پیاده‌سازی کنید:

```
root_finder(equation, interval, error)
```

در صورت نیاز به سایر متغیرها از $global$ استفاده کنید.
برای محاسبه‌ی ریشه با تعداد ارقام خواسته شده از ماژول $decimal$ و توابع آن مانند $getcontext().prec$ استفاده کنید.

برای محاسبه‌ی مقدار یک تابع میتوانید از تابع $eval()$ استفاده کنید به شکل زیر:

```
>>> c = "a^2+2*a+1"
>>> eval(c)
5
```

نحوه‌ی دادن ورودی به شکل زیر است:
در سطر اول ضرایب چند جمله‌ای مورد نظر به شما داده خواهد شد بین ضرایب یک فاصله خواهد بود و ضرایب از بیشترین درجه تا کمترین درجه خواهد بود.
در سطر بعدی بازه‌ی مورد نظر داده خواهد شد .(ابتدا کران پایین و سپس کران بالا)
در سطر سوم مقدار خطا را دریافت میکنید.
در سطر چهارم و آخر نیز تعداد ارقام ارعشار به شما داده میشود.
معادله‌ی $۳x^۲ - x - ۱$ را در نظر بگیرید و به مثال زیر دقت کنید:

```
3 -1 -1
0 1
0.01
10
```

خروجی:

```
0.7685546875
```

سؤال ۴. فتوشاپ متنی (۲۵ امتیاز اضافه)(امتیازی) $(photoshop.py)$

می‌خواهیم یک فتوشاپ بسیار ساده بنویسیم که چند فیلتر ساده $blending$ را روی عکس‌ها پیاده کند. یک عکس در واقع ماتریس دو بعدی از پیکسل‌هاست. برای سادگی فرض می‌کنیم که هر پیکسل را می‌توان با یک عدد (رنگ) نشان داد که هر چه عدد کوچکتر باشد یعنی رنگ تیره تر است. پس برای نمایش یک عکس کافی است کاراکتر مربوط به هر خانه‌ی ماتریش را نمایش دهیم. رنگ‌ها و عددشان عبارت‌اند از:

```
5:  [space]
4:  .
3:  ,
2:  :
1:  *
0:  #
```

می‌خواهیم از ترکیب دو عکس فیلترهای $blending$ مختلف بسازیم. عکسی که می‌خواهیم تغییرش دهیم را $bottom$، عکسی که روی عکس ما می‌آید و آن را تغییر دهد را top و عکس خروجی را $result$ می‌نامیم. در هر فیلتر تمام خانه‌های دو عکس طبق فرمول هر فیلتر روی هم اعمال می‌شوند و خانه نظیر عکس نهایی را می‌سازند. ۳ فیلتر داریم که عبارتند از:

- add: رنگهای top و $bottom$ جمع می‌شوند. $result = top + bottom$
- $darken$: نقاط تیره تر از بین دو عکس انتخاب شده و عکس نهایی را می‌سازند. $result = min(top, bottom)$
- $multiply$: با استفاده از top عکس را تیره تر می‌کند. $result = (topbottom)//۵$
- $init$: عکس جدید را جایگزین قبلی میکند.

حال شما برنامه‌ای بنویسید که عکس و فیلتر را از ورودی بگیرد و روی عکس شما اعمال کند. و هر زمان که دستور $print$ را دید تصویر را چاپ کند. برنامه شما با دیدن دستور $finish$ پایان می یابد. نکات مهم:

- تمام تصاویر $۴x۵$ هستند.

- قبل از گرفتن ورودی تصویر شما یک تصویر خالی است. (همه نقاط عکس مقدار ۵ را داشته باشند)
- عکس حاصل از یک فیلتر عکس اصلی شما خواهد بود در فیلترهای بعدی.
- برای تبدیل اعداد یک رشته به لیستی از اعداد از تابع $split$ استفاده کنید. بین کاراکترها در هر سطر یک $space$ چاپ کنید.
- هر کجا که عدد حاصل در $result$ بیشتر از ۵ شد آنرا ۵ و هرکجا که کمتر از ۰ شد آنرا ۰ بگیرید.

نمونه ورودی:

```
init
. . . . ,
, , * :
    # # #
. . . . .
print
```

نمونه خروجی:

```
. . . . ,
, , * :
    # # #
. . . . .
```

نمونه ورودی:

```
init
  . , .
. , : , .
, : # : ,
. , , , .
multiply
* : : : *
. . : . .
,       ,
. . # . .
print
```

نمونه خروجی:

```
* * * * *
, : # : ,
* : # : *
, : # : ,
```

موفق و پیروز باشید!