



مبانی برنامه‌سازی پایتون

نیم‌سال دوم ۹۲-۹۳

مدرس: بی‌نظیر بیگلری

دانشکده‌ی مهندسی کامپیوتر

مهلت ارسال: ۲۱ اسفند

حلقه رشته‌ها و توابع

تمرین یک

به موارد زیر توجه کنید:

- حتما در ارسال فایل برنامه‌ها به نام فایل و فرمت ورودی و خروجی‌ها توجه شود. هیچ‌گونه کوتاهی و بی‌دقتی 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.
- فایل‌ها در سامانه جاج بارگذاری خواهند شد که آدرس و نحوه‌ی 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 تمرین‌ها 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.
- کامنت‌گذاری و نام‌گذاری مناسب متغیرها الزامی‌ست و بخشی از نمره‌ی شما را در تحویل حضوری مشخص می‌کند.

گروه تمرین

سؤال ۱. فیبوناچی (۱۵ امتیاز) (*fibonacci.py*)

برنامه‌ای بنویسید که جمله‌های دنباله‌ی فیبوناچی را گام به گام (تا گام خواسته شده) چاپ کند تعداد جملات فیبوناچی توسط کاربر به عنوان ورودی به برنامه داده می‌شود و سپس برنامه‌ی شما جملات فیبوناچی را تا مرحله‌ی خواسته شده چاپ می‌کند.

نمونه ورودی

```
2
```

نمونه خروجی

```
1
1
```

نمونه ورودی

```
5
```

نمونه خروجی

```
1
1
2
3
5
```

سؤال ۲. توابع رشته‌ای (۲۰ امتیاز) (*str.py*)

برنامه‌ای بنویسید که از کاربر یک رشته را بگیرد سپس به نسبت کاراکتری که کاربر وارد میکند، کار خاصی انجام دهد. به ازای کاراکترهای زیر باید عملیاتی که جلوی آنها قرار دارد را انجام دهد.

- *: برنامه رشته وارد شده را معکوس کرده و چاپ میکند
- /: برنامه تمام رشته‌ی وارد شده را به صورت حروف بزرگ در آورده و چاپ میکند
- ?: برنامه رشته‌ی فعلی را رها میکند و رشته‌ی جدیدی از کاربر میگیرد و ادامه میدهد
- +: برنامه حروف رشته را به صورت یکی در میان بزرگ و کوچک میکند (اولین حرف بزرگ می‌شود و اگر کاراکتر غیرحرفی در میان رشته بود از روی آن عبور میکند و ادامه میدهد)
- #: به برنامه خاتمه میدهد

توجه کنید که کار گرفتن رشته و سپس انجام عملیات مورد نظر کاربر تا زمانی ادامه میابد که کاربر کاراکتر # را وارد کند. در ضمن حل حتما از دستورات *break*, *continue* استفاده شود. در صورت عدم استفاده از این دستورات نمره‌ای تعلق نخواهد گرفت. راهنمایی: از `while(True)` استفاده کنید.

استفاده از توابع رشته‌ای پایتون مجاز است و برای سادگی برنامه‌نویسی توصیه می‌گردد. به عنوان ورودی ابتدا رشته و سپس کاراکتر وارد میشود و این کار به طور پیاپی ادامه میابد. در خروجی برنامه به نسبت کاراکتر ورودی پیام‌های مناسب چاپ میشود. فرض کنید کاربر ابتدا رشته *hello* را وارد کرده :

نمونه ورودی:

```
hello
*
/
-
hey
+
#
```

نمونه خروجی:

```
olleh
HELL0
HeY
```

نمونه ورودی:

```
w36stis5v
/
+
#
```

نمونه خروجی:

```
W36STIS5V
W36sTiS5v
```

سؤال ۳. برعکس (۳۰ امتیاز) (*rev.py*)

یک تابع با نام `reverse` بنویسید که به عنوان ورودی یک کلمه را در قالب رشته بگیرد و معکوس آن را به عنوان خروجی برگرداند. سپس با کمک این تابع تابع دیگری به نام `reverse_sentence` بنویسید که جمله را به صورت معکوس برگرداند. دقت کنید که کلمات با فاصله از هم جدا می شوند و در صورت وجود چند فاصله یا علامت نگارشی پشت سر هم در رشته خروجی فقط یک فاصله بیاید. دقت کنید کلمات ممکن است با فاصله یا هر یک از علائم نگارشی `!.,` از هم جدا شوند. رشته‌هایی که به شما داده می‌شوند تنها شامل حروف بزرگ و کوچک و فاصله و علائم نگارشی ذکر شده در بالا هستند. استفاده از توابع رشته‌ای پایتون مانند `reverse` یا `slicing extended` مجاز نیست. برنامه‌ی شما می‌بایست توابع زیر را پیاده‌سازی کند و در تابع `reverse_sentence` باید تابع `reverse` صدا زده شود و از آن استفاده شود.

```
reverse (word):  
  
    { your code}  
  
    return reversed_word  
  
reverse_sentence (str):  
  
    { your code }  
  
    return reversed_str
```

نمونه ورودی:

```
Hi, Python!
```

نمونه خروجی:

```
nohtyP iH
```

نمونه ورودی:

```
This is a sentence
```

نمونه خروجی:

```
ecnetnes a si sihT
```

سؤال ۴. متد رشته‌ای (۳۰ امتیاز) (*simpfind.py*)

تابع `find` از توابع پرکاربرد رشته‌هاست. این تابع یک رشته و دو ورودی اختیاری شروع و پایان می‌گیرد و اگر رشته ورودی بین زیر رشته‌ی شروع شده از عدد شروع و پایان یافته تا عدد پایان در رشته اصلی که `find` روی آن صدا زده می‌شود وجود داشت نمایه (`index`) اولین حرف اولین وقوع رشته ورودی در رشته اصلی برگردانده می‌شود. اگر رشته ورودی در رشته اصلی وجود نداشت عدد `-۱` برگردانده می‌شود. بزرگی و کوچکی حروف مهم است. ما می‌خواهیم نسخه‌ی ساده‌تری از `find` را پیاده کنیم. بدین منظور تابعی بنویسید که دو رشته به عنوان ورودی می‌گیرد و رشته دوم را در رشته اول جستجو می‌کند. (در واقع برای سادگی کار ورودی‌های شروع و پایان را در نظر نگرفته ایم). به تعریف تابع و مثال‌های زیر دقت کنید.

```
find (string1, string2)
```

تابع به شکل بالا تعریف می‌گردد.

نمونه ورودی:

```
find("Viva metal", "metal")
```

نمونه خروجی:

5

نمونه ورودی:

```
find("Hi Python!", "python")
```

نمونه خروجی:

-1

سؤال ۵. هنرمند (۲۵ امتیاز اضافه) (art.py)

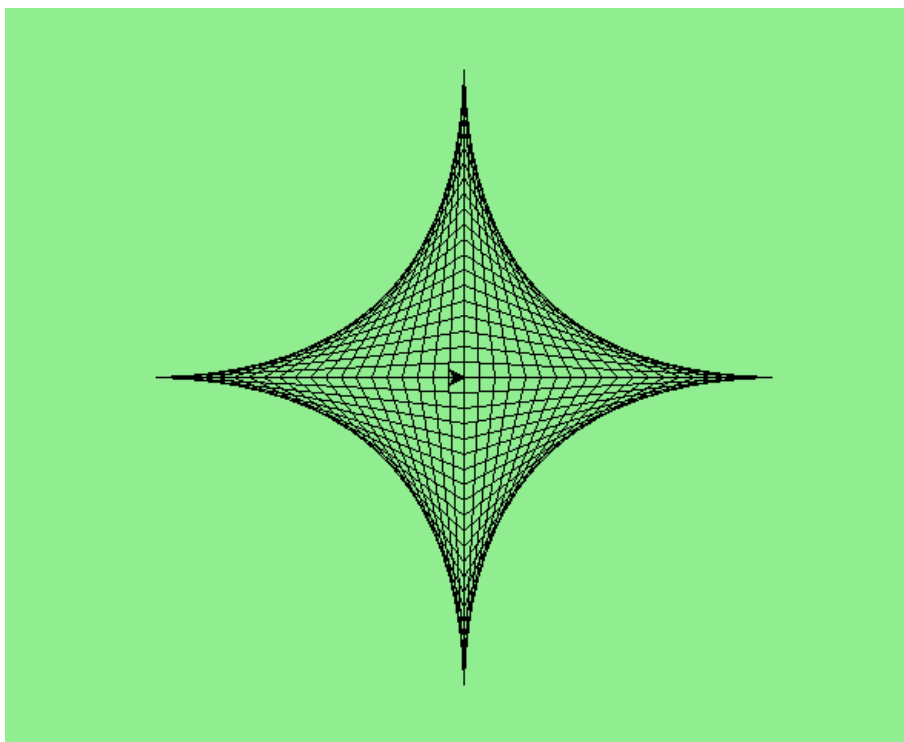
می‌خواهیم شکل زیر را توسط لاک پشت دوست داشتنی پایتون طراحی کنیم. برای این کار ابتدا تابعی با نام *design* تعریف میکنیم. ورودی این تابع یک عدد طبیعی است به این ترتیب که این عدد نشان دهنده تعداد قسمتهای جدا شده روی هر محور است. سپس به کمک این تابع برنامه‌ای بنویسید که از کاربر عدد مورد نظر را دریافت کند و با استفاده از تابع ذکر شده این شکل را ترسیم کند. به نمونه‌ی زیر دقت کنید.

دقت کنید این سوال توسط جاج تصحیح نخواهد شد ولی حتما در زمان مشخص باید در جاج بارگذاری شود. نمره‌ی این سوال در تحویل حضوری مشخص می‌گردد.

نمونه ورودی:

20

نمونه خروجی:



شکل ۱: خروجی مورد نظر