



سؤال ۱. خطایابی

برنامه‌ی زیر یک امتحان ریاضی ساده را پیاده‌سازی کرده است که ده سوال در مورد حاصل ضرب اعداد مختلف را از کاربر سوال می‌کند و در انتها تعداد جواب‌های درست را به او اعلام می‌کند. متأسفانه دانش‌جویی که این برنامه را پیاده‌سازی کرده است دقت لازم را به خرج نداده و برنامه‌اش تعدادی خطا دارد! شما باید خطاهای موجود در این برنامه را اصلاح کنید.

```
from random import randint

for i in range(10)
    n1 = randint(1, 10)
    n2 = randint(1, 10)
    product = n1 * n2

    ans = input('What is {} times {}? '.format(n1, n2))
    if ans = product:
        print('Correct. Well done!')
        correct = correct + 1
    else:
        print('No. The correct answer is {}'.format(product))

print('You got {} out of 10 questions right.'.format(correct))
print 'Well done!'
```

سؤال ۲. چندجمله‌ای

یک چندجمله‌ای را می‌توان به صورت لیستی از ضرایب جملات آن نشان داد، طوری که خانه‌ی i ام این لیست، نشان‌دهنده‌ی ضریب جمله‌ی x^i در چندجمله‌ای است. به طور مثال چندجمله‌ای زیر

$$5x^4 - 3x^3 + 3x^2 + 1$$

توسط یک لیست مانند L به صورت زیر نشان داده می‌شود:

$$L = [1, 0, 3, -3, 5]$$

تابع `polynomial` به صورت زیر به شما داده شده است. این تابع یک لیست از اعداد را دریافت می‌کند و چندجمله‌ای متناظر با آن را به عنوان خروجی برمی‌گرداند.

```
def polynomial(coeffs):
    result = ''
    degree = len(coeffs)

    for i in range(degree):
        result += '{0}x^{1}'.format(coeffs[i], i)
        result += '+'

    return result
```

اما این تابع اشکالاتی دارد که شما باید به شرح زیر آن‌ها را اصلاح کنید:

- ابتدا جملات با درجه‌ی بالاتر نمایش داده شوند.
- جملات با ضرایب صفر نباید نشان داده شوند.
- در جمله‌ی x^0 فقط ضریب آن باید نشان داده شود (یعنی ۵ به جای $5x^0$).
- ضرایب ۱ نباید نشان داده شوند.
- در صورتی که همه‌ی ضرایب صفر باشند، در خروجی ۰ چاپ شود.
- علامت جمع اضافی چاپ نشود.

سؤال ۳. رشته‌ها

در این سؤال سه تابع برای کار با رشته‌ها خواهید نوشت.

الف) تابع `join` به صورت زیر تعریف شده است:

```
def join(L, separator)
```

این تابع رشته‌های موجود در لیست `L` را به وسیله `separator` به هم می‌چسباند و خروجی را به عنوان یک رشته برمی‌گرداند. به عنوان مثال:

```
>>> join(['salam', 'test', 'book', '1234', 'exam'], 'A+A')
'salamA+AtestA+AbookA+A1234A+Aexam'
```

تابع `join` را بدون استفاده از توابع ازپیش‌تعریف‌شده در کلاس `str` پیاده‌سازی کنید.

ب) تابع `split` برعکس تابع `join` عمل می‌کند. بدین صورت که دو رشته را به عنوان ورودی دریافت می‌کند و رشته‌ی اول را به وسیله‌ی جداکننده‌ای که در رشته‌ی دوم تعیین شده است می‌شکند. در خروجی یک لیست از عناصر جداشده توسط این تابع بازگردانده می‌شود.

```
>>> split('this+is+a++test++here++', '+')
['this', 'is', 'a', '', 'test', '', 'here', '', '']
```

ج) به کمک توابع `split` و `join` که در بالا نوشته‌اید، تابع `delete` را به نحوی پیاده‌سازی کنید که دو رشته را دریافت کند و تکرارهای رشته‌ی دوم در رشته‌ی اول را حذف نماید.

```
>>> delete('this is a test!', 's ')
'thiia test!'
```

سؤال ۴. مرتب‌سازی سریع

یک الگوریتم بسیار کارآ برای مرتب‌سازی، الگوریتم مرتب‌سازی سریع یا Quicksort است. این الگوریتم به این صورت عمل می‌کند که ابتدا یکی از عناصر لیست (مثلاً عنصر اول آن) را به نام «محور» انتخاب می‌کند و سپس در طی یک عملیات به نام partition، عناصر لیست را به سه دسته‌ی زیر تقسیم می‌کند:

- عناصر کوچک‌تر از محور
- عناصر برابر محور
- عناصر بزرگ‌تر از محور

نمونه‌ای از عمل‌کرد تابع partition در زیر آمده است. این تابع یک لیست و یک عدد را به عنوان محور دریافت کرده و لیست را بر اساس محور به سه دسته‌ی فوق‌الذکر افراز می‌کند.

```
>>> partition([4, 5, 3, 9, 7, 1], 4)
([3, 1], [4], [5, 9, 7])
```

حال کافی است عناصر دسته‌ی اول و عناصر دسته‌ی سوم را به صورت بازگشتی مرتب کنیم، و از کنار هم قرار دادن این سه دسته، نهایتاً کل لیست مرتب می‌شود. شما قرار است الگوریتم مرتب‌سازی سریع را به نحوی که در بالا توضیح داده شد پیاده‌سازی کنید. برای این کار ابتدا تابع partition را نوشته و سپس با استفاده از آن تابع quicksort را به صورت بازگشتی پیاده‌سازی کنید.

سؤال ۵. هشت وزیر

می‌خواهیم هشت مهره‌ی وزیر را طوری در صفحه‌ی شطرنج قرار دهیم که هیچ دو وزیری یک‌دیگر را تهدید نکنند. از آن جایی که در هر سطر نمی‌تواند بیش‌تر از یک وزیر قرار داشته باشد، هر پاسخ برای این مسئله را می‌توان به صورت یک لیست هشت‌تایی نشان داد که در خانه‌ی اول آن شماره‌ی ستون وزیر موجود در سطر اول، در خانه‌ی دوم آن شماره‌ی ستون وزیر موجود در سطر دوم، و ... قرار دارد. از آن جایی که دو وزیر نمی‌توانند در یک ستون نیز باشند، پس این لیست اعضای تکراری نخواهد داشت. این مسئله را می‌توان به وسیله‌ی الگوریتم تصادفی زیر حل کرد.

۱. یک جای‌گشت تصادفی از اعداد ۱ تا ۸ به عنوان جواب محتمل مسئله تولید کن.

۲. در صورتی که جای‌گشت تولیدشده یک جواب معتبر است (یعنی هیچ دو وزیری یک‌دیگر را تهدید نمی‌کنند)، آن را به عنوان جواب مسئله بازگردان؛ در غیر این صورت مرحله‌ی اول را تکرار کن.

برای پیاده‌سازی این الگوریتم می‌توانید از تابع `shuffle` در ماژول `random` استفاده کنید. نمونه‌ای از این استفاده در زیر آمده است.

```
>>> import random
>>> a = [1, 2, 3, 4]
>>> random.shuffle(a)
>>> print(a)
[2, 4, 3, 1]
```