

به نام خدا

تمرین اول درس برنامه‌نویسی پیشرفته

نیم‌سال اول ۱۳۹۹-۱۴۰۰

۰. فایل مربوط به توضیحات نحوه ارسال تمرین‌ها را که در مودل قرار دارد، مطالعه کنید.

۱. تمامی فایل‌های کد را به همراه فایل متنی که در قالب pdf است (مورد سوم را بخوانید) به صورت یک فایل آرشیو zip (zip != rar) که به قالب زیر نام‌گذاری شده است، بارگذاری نمایید.

StudentNumber_FirstName_LastName.zip

9031066_Ehsan_Edalat.zip

۲. در سوال‌هایی که ورودی و خروجی مطلوب آن‌ها مشخص شده است، برنامه‌ی شما به صورت ماشینی تصحیح می‌شود. بنابراین رعایت نحوه ورودی گرفتن و نمایش خروجی اهمیت بسیاری دارد. دقیقاً همان‌طور که از شما خواسته شده است ورودی‌ها را خوانده و خروجی‌ها را تولید کنید.

۳. پاسخ سوالات تشریحی را به صورت تایپ‌شده و در قالب یک فایل pdf (برای کل تمرین) تحویل دهید.

۴. در صورت مشاهده هرگونه تقلبی، طبق موارد گفته شده در قوانین درس برخورد خواهد شد.

۵. در صورت وجود هرگونه ابهام می‌توانید از طریق ربات تلگرامی [AP_Admin_bot](#) با تدریس‌یاران در

ارتباط باشید.

مهلت تحویل: تا جمعه ۱۱ مهر ۱۳۹۹ ساعت ۲۳:۵۵ شب

صفحه

فهرست سوالات

- سوال اول.....۳
- سوال دوم.....۳
- سوال سوم.....۴
- سوال چهارم.....۴
- سوال پنجم.....۵
- سوال ششم.....۷
- سوال هفتم - امتیازی.....۷

سوال اول

در درس مبانی برنامه‌نویسی با نحوه‌ی برنامه‌نویسی ساخت‌یافته^۱ آشنا شده‌اید. در درس برنامه‌نویسی پیشرفته، مفاهیم برنامه‌نویسی با دیدگاه شی‌گرایی^۲ را می‌آموزید؛ حال در این تمرین، قصد داریم به آشنایی مقدماتی این مفاهیم بپردازیم. در این رویکرد برنامه‌نویسی، مفهومی به نام شی^۳ تعریف می‌شود و برای مفهوم شی، مفاهیم حالت^۴، رفتار^۵ و هویت^۶ بیان می‌شود. از شما می‌خواهیم تا این مفاهیم را با ذکر مثالی توضیح دهید.

سوال دوم

در این تمرین قصد داریم با تفاوت برنامه‌نویسی ساخت‌یافته و شی‌گرا آشنا شویم. برنامه‌نویسی شی‌گرا و ساخت‌یافته را به اختصار شرح داده؛ سپس مزایا و معایب شی‌گرایی را نسبت به برنامه‌نویسی ساخت‌یافته نام ببرید.

¹ Structured

² Object-Oriented

³ Object

⁴ State

⁵ Behavior

⁶ Identity

سوال سوم

شما در زبان برنامه‌نویسی C با string و پیاده‌سازی این مفهوم آشنا شدید. در زبان جاوا، ما کلاسی به نام String داریم، تفاوت پیاده‌سازی string در C و جاوا را بیان کنید؛ همچنین این کلاس شامل توابع زیر است (این توابع تمام توابع این کلاس نیستند و بهتر هست نگاهی به توابع دیگر نیز بیندازید):

```
* char charAt(int index)
* String replace(CharSequence old, CharSequence new)
* boolean equals(Object another)
* String[] split(String regex, int limit)
* String concat(String str)
* static String format(String format, Object... args)
* int indexOf(int ch)
```

درباره‌ی این توابع و ورودی و خروجی آن‌ها توضیح دهید.

سوال چهارم

(پیاده‌سازی)

در این سوال قصد داریم، یک عدد را به رشته تبدیل کنیم، شرایط این تبدیل به شرح زیر است:

* اگر عدد بر ۳ بخش‌پذیر باشد، خروجی باید شامل رشته «Pthree» باشد.

* اگر عدد بر ۵ بخش‌پذیر باشد، خروجی باید شامل رشته «Pfive» باشد.

* اگر عدد بر ۷ بخش‌پذیر باشد، خروجی باید شامل رشته «Pseven» باشد.

* اگر هیچ کدام از موارد بالا نبود، باید عدد ورودی به عنوان خروجی نوشته شود.

مثال ۱:

Input: 28

Output: Pseven

* مقسوم‌علیه‌های عدد ۲۸، اعداد ۱، ۲، ۴، ۷، ۱۴ و ۲۸ هستند.

مثال ۲:

Input: 30

Output: PthreePfive

مثال ۳:

Input: 34

Output: 34

* مقسوم‌علیه‌های عدد ۳۴، شامل اعداد ۱، ۲، ۳ و ۷ نیستند.

سوال پنجم

(پیاده‌سازی)

در زبان برنامه‌نویسی C با تابع `atoi` آشنا شدید، وظیفه‌ی این تابع، تبدیل رشته‌ای از کاراکتر به `integer` است. حال ما از شما می‌خواهیم تا چنین تابعی را در جاوا پیاده‌سازی کنید. این تابع ابتدا تعداد کاراکترهای `space` را تا زمانی که اولین کاراکتر غیر `space` مشاهده شود، حذف می‌کند. سپس، با شروع از این کاراکتر، یک علامت اولیه اختیاری (+ یا -)، به دنبال آن تا آنجا که ممکن است تعدادی عدد می‌گیرد و آنها را به عنوان یک مقدار عددی تفسیر می‌کند.

* این رشته می‌تواند حاوی حروف اضافی بعد از حروف تشکیل دهنده عدد باشد، که این حروف اضافی نادیده گرفته می‌شوند و تأثیری در رفتار عملکرد تبدیل ندارند.

* اگر اولین دنباله از حروف غیر space در متن، یک عدد معتبر نباشد (متن باشد)، یا اگر چنین توالی وجود نداشته باشد (زیرا یا متن خالی است یا فقط شامل space است)، هیچ تغییری انجام نمی‌شود.

* اگر هیچ تبدیل معتبری انجام نشود، مقدار صفر بازگردانده می‌شود.
* اعداد موجود از محدوده‌ی Int32 خارج نمی‌شوند.

مثال ۱:

Input: 132

Output: 132

مثال ۲:

Input: -132

Output: -132

* اولین کاراکتر غیر space برابر با - است.

* space های ابتدای ورودی نادیده گرفته می‌شوند.

مثال ۳:

Input: " -132 blah blah"

Output: -132

* تبدیل در رقم ۲ متوقف می‌شود، چون به حرف رسیده است.

مثال ۴:

Input: blah blah 123

Output: 0

* تبدیل صورت نمی‌گیرد، چون عددی یافت نمی‌شود.

سوال ششم

(پیاده‌سازی)

در این برنامه، باید کاراکترهای تکراری در رشته را پیدا کنیم. برای یافتن کاراکتر تکراری از رشته، وقوع هر یک از کاراکترها را در رشته می‌شماریم. اگر تعداد از ۱ بیشتر باشد، این بدان معنی است که این کاراکتر یک کاراکتر تکراری در رشته است. برای درک بهتر، مثال زیر را مشاهده کنید.

Input: Great responsibility

Output:

r
e
t
s
i

* فرمت خروجی به شکل بالاست. و اگر حرفی تکراری نباشد خروجی خالی خواهد بود.

سوال هفتم - امتیازی

(پیاده‌سازی)

یک تابع بنویسید تا بررسی کند دو رشته داده شده با یکدیگر «یک‌ریشه» هستند یا خیر. «یک‌ریشه»ی یک رشته، رشته دیگری است که شامل همان کاراکترها است، فقط ترتیب کاراکترها می‌تواند متفاوت باشد؛ به عنوان مثال، "abcd" و "dabc" «یک‌ریشه» هستند. مثال‌های زیر نحوه ورودی/خروجی را نمایش می‌دهند (خروجی با حروف بزرگ است).

مثال ۱:

Input: first
second

Output: NO

مثال ۲:

Input: abcd
dabc

Output: YES