



تمرین کامپیوتری شماره ۲

ساختمان داده - پاییز ۱۴۰۱

دانشکده مهندسی برق و کامپیوتر

طراحان تمرین: **پارسا موید**

مهلت تحویل: ۱۴۰۱/۰۱/۱۵ (۱۲ شب)

مدرس: دکتر هشام فیلی

مقدمه

این تمرین کامپیوتری برای آشنایی با داده ساختارهای ساده و نحوه استفاده از آنها است. در قسمت اول به شما یک قالب از سه داده ساختار `stack` ، `queue` ، `linkedList` داده می شود و انتظار می رود که با توجه به مطالب گفته شده در رابطه با هر تابع، آنها را کامل کنید.

مسئله‌ی اول: دستگرمی (۲۵ نمره)

- محدودیت زمان ۱ ثانیه
- محدودیت حافظه ۲۵۶ مگابایت

در این بخش قالبی در اختیار شما قرار می‌گیرد که توضیحات آن در ادامه آمده‌است.

توضیح ارورها :

در هر تابع، حالت‌هایی وجود دارد که موجب رخ دادن ارور می‌شود (مانند پاپ کردن از هیپ خالی). در صورت رخ دادن آنها، صرفاً آنها را به صورت زیر هندل کنید:

```
raise Exception('error_text')
```

تمام این ارور ها عبارت اند از (بقیه ارور ها بررسی نمی‌شوند) :

```
raise Exception('invalid index') -> ایندکس وارد شده عدد نباشد یا تایپ آن درست نباشد
```

```
raise Exception('out of range index') -> ایندکس وارد شده در محدوده سبایز نباشد
```

```
raise Exception('empty') -> از هیپ یا درخت خالی مقداری خارج شود
```

توضیح توابع:

```
@for_all_methods(fix_str_arg)
```

```
@for_all_methods(print_raised_exception)
```

```
class Queue :
```

```
    def __init__(self) -> کانستراکتور صف
```

```
    def getSize(self) -> تعداد عناصر ذخیره شده در صف را بر می‌گرداند
```

```
    def enqueue(self, value) -> عنصری را به صف اضافه می‌کند
```

```

def dequeue(self) -> عنصری از صف کم می کند و آنرا برمی گرداند

def isEmpty(self) -> از جنس بولین بوده و درستی خالی بودن صف را برمی گرداند

def getInOneLine(self) -> تمام عناصر را در یک خط مانند 1 2 3 بر می گرداند

@for_all_methods(fix_str_arg)
@for_all_methods(print_raised_exception)

class Stack :

    def __init__(self, size=10) -> کانستراکتور پشته

    def isEmpty(self) -> از جنس بولین بوده و درستی خالی بودن پشته را برمی گرداند

    def push(self, value) -> عنصری را به پشته اضافه می کند

    def pop(self) -> عنصری از پشته کم می کند و آنرا برمی گرداند

    def put(self, value_) -> عنصری از پشته کم و عنصر دیگری را جایگزین می کند

    def peek(self) -> عنصر بالای پشته را صرفاً می گرداند یعنی حذف نمی کند

    def expand(self) -> سایز تعریف شده (با تعداد عناصر ذخیره شده فرق دارد) را دو برابر می کند

    def getInOneLine(self) -> تمام عناصر را در یک خط مانند 1 2 3 بر می گرداند

    def getSize(self) -> تعداد عناصر ذخیره شده در صف را بر می گرداند

    def getCapacity(self) -> سایز تعریف شده را برمی گرداند

class Node():

    def __init__(self, val) -> کانستراکتور نود که برای نوشتن کانستراکتور لینکد لیست لازم است

```

```
class LinkedList():
```

```
    def __init__(self) -> کانستراکتور لینکد لیست
```

```
    def insertFront(self, new_data) -> در ابتدای لیست عنصری را وارد می کند
```

```
    def insertEnd(self, new_data) -> در انتهای لیست عنصری را وارد می کند
```

```
    def reverse(self) -> لیست را بر عکس می کند
```

توضیح در مورد قالب

قالب شامل چند کلاس و تابع می باشد که کافی است توابع مشخص شده در بالا را کامل کنید و نیازی به یادگیری مابقی قالب نیست.

ورودی

با توجه به قالب داده شده ابتدا یک یا چند آبجکت از نوع پشته یا صف یا لینکدلیست ایجاد می شود. سپس توابع مشخص شده برای هر کدام صدا زده می شوند که همگی در قالب آمده است و توضیح مربوط به هر کدام در pdf تمرین آمده است.

نمونه ی ورودی و خروجی 1

Input:

```
make queue q1
call q1.dequeue(1)
call q1.enqueue(1)
call q1.enqueue(2)
call q1.enqueue(3)
call q1.isEmpty()
call q1.getSize()
call q1.getlnOneLine()
call q1.dequeue()
call q1.dequeue()
call q1.dequeue()
call q1.isEmpty()
call q1.getSize()
```

Output:

empty
False
3
1 2 3
1
2
3
True
0

نمونه‌ی ورودی و خروجی 2

Input:

```
make stack s1  
call s1.push(1)  
call s1.push(2)  
call s1.push(3)  
call s1.isEmpty()  
call s1.getSize()  
call s1.getInOneLine()  
call s1.pop()  
call s1.pop()  
call s1.pop()  
call s1.isEmpty()  
call s1.getSize()
```

Output:

False
3
1 2 3
3
2
1
True
0

مسئله‌ی دوم: سهراب سگمنت بلد نیست (۲۵ نمره)

- محدودیت زمان ۱ ثانیه
- محدودیت حافظه ۲۵۶ مگابایت

علی یک آرایه n عضوی دارد که مدام ماکزیمم یک بازه از آن را از سهراب می‌پرسد. سهراب که از حساب کردن این همه ماکزیمم خسته شده، به علی می‌گوید "من سگمنت بلد نیستم". علی هم تصمیم می‌گیرد سوال‌هایش را ساده تر کند و تضمین می‌کند طول هر بازه‌ای که از سهراب می‌پرسد دقیقاً k باشد. سهراب می‌داند این سوال را قبلاً حل کرده‌است. اما راه حل آن یادش نمی‌آید. پس از شما کمک می‌خواهد تا جواب سوال‌های علی را بدهید.

ورودی

در خط اول دو عدد n, k می‌آید که نشان‌دهنده‌ی تعداد اعضای آرایه و طول بازه پرسش است.

سپس در خط بعد، n عدد می‌آید که عدد i ام a_i است.

در خط سوم q می‌آید که تعداد پرسش‌ها است.

در q خط بعدی، در هر خط یک عدد $k \leq r \leq n$ می‌آید که نشان‌دهنده راست بازه است.

خروجی

برای هر پرسش یک عدد خروجی دهید که نشان‌دهنده ماکزیمم بازه $[r - k + 1, 1, r]$ باشد. توجه: بازه بسته است.

محدودیت‌ها

$$1 \leq n < 2 * 10^5$$

$$1 \leq a_i < 10^5$$

نمونه‌ی ورودی و خروجی ۱

Input:

9 3

1 3 1 1 1 2 5 2 3

5

5

6

9

5

7

Output:

1

2

5

1

5

نمونه‌ی ورودی و خروجی ۲

Input:

9 4

6 5 2 3 2 3 8 4 9

6

4

5

6

7

8

9

Output:

6

5

3

8

8

9

مسئله‌ی سوم: پیرمرد مهربان (۲۵ نمره)

- محدودیت زمان ۱ ثانیه
- محدودیت حافظه ۲۵۶ مگابایت

پیرمرد آب‌نبات فروش برای عید پیشنهاد جذابی گذاشته‌است. او برای هر x چوب که به او پس دهید یک آب‌نبات جایزه می‌دهد (از آنجایی که پیرمرد قصد ورشکسته‌شدن ندارد x از یک بیشتر است). قیمت هر آب‌نبات p است و علی در جیب خود q تومان پول دارد. او می‌خواهد بداند که با این پول حداکثر چقدر می‌تواند آب‌نبات بگیرد. به علی در حل این پرسش کمک کنید و تعداد آب‌نبات‌هایی که می‌تواند بخورد را به او بگویید (البته ما به علی توصیه می‌کنیم از پیرمرد ها آب‌نبات نگیرد!).

ورودی

در تنها خط ورودی به ترتیب سه عدد q و p و x می‌آید که به ترتیب پول علی، قیمت هر آب‌نبات و تعداد چوب آب‌نبات‌هایی که باید پس داده‌شود تا آب‌نبات جدید بگیرد.

خروجی

یک عدد خروجی دهید که تعداد آب‌نبات‌هایی است که علی می‌تواند بگیرد.

محدودیت‌ها

$$x, p, q \leq 10^9$$

نمونه‌ی ورودی و خروجی ۱

Input:

10 2 5

Output:

6

نمونه‌ی ورودی و خروجی ۲

Input:

10 2 2

Output:

9

مسئله‌ی چهارم: یک سوال پرانتزگذاری دیگر (۲۵ نمره)

- محدودیت زمان ۱ ثانیه
- محدودیت حافظه ۲۵۶ مگابایت

محمد به تازگی با مسئله پرانتزگذاری آشنا شده‌است و دنبال یک سوال چالشی‌تر می‌گردد. در این سوال یک رشته شامل پرانتز، براکت و کروشه به شما داده می‌شود و شما باید بلندترین پسوند از این رشته که یک عبارت ریاضی درست را تشکیل می‌دهد خروجی دهید.

یک عبارت ریاضی درست به این صورت تعریف می‌شود:

$[], \{\}, ()$ یک عبارت ریاضی درست هستند.

رشته S یک عبارت ریاضی درست است اگر بتوان آن را به یکی از چهار صورت زیر نوشت:

(T) , $[T]$, $\{T\}$, $T.Q$

که T , Q خودشان عبارت ریاضی درست باشند.

توجه: عملگر «.» همان concatenation یا چسباندن دو رشته است.

ورودی

در یک خط رشته ای شامل کاراکتر های $[\]$, $\{ \}$, $()$ داده می شود. که طول آن حداکثر ده به توان ۵ است.

خروجی

بلندترین پسوند از رشته که یک پرانتزگذاری معتبر است را خروجی دهید.

نمونه ی ورودی و خروجی ۱

Input:

{O[(OO)}{[]}{((O))}[]

Output:

12

نمونه ی ورودی و خروجی ۲

Input:

{((([OO])))}{[OO{}]}(

Output:

14

نکات تکمیلی

- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

- استفاده از کدهای آماده برای پیاده‌سازی این مباحث (جستجو شده در اینترنت و ...)، مجاز نمی‌باشد. در صورت کشف، مانند تقلب برخورد می‌شود.
- در تمامی سوالات به جز مواردی که در ادامه گفته می‌شود نباید از کتابخانه های آماده استفاده شود.
 - در سوال اول از کتابخانه sys استفاده شده که برای آپلود استفاده از آن مشکلی ندارد.
 - در سوال ۲ و ۳ و ۴ اجازه استفاده از کتابخانه deque را دارید.
- در صورتی که تست‌های تمامی سوالات پاس بشوند و نمره آنها کامل شود، ۱۰ نمره امتیازی اعمال می‌شود (نمره ۱۰۰ ، ۱۱۰ خواهدشد).