



تمرین کامپیوتری شماره ۲

ساختمان داده - بهار ۱۴۰۳

دانشکده مهندسی برق و کامپیوتر

مهلت تحویل: ۱۴۰۳/۰۲/۰۴ (۱۲ شب) طراحان تمرین: فاطمه کرمی محمدی،

مدرس: دکتر هشام فیلی

میثاق محقق، سروش صحرانی

مقدمه

این تمرین کامپیوتری برای آشنایی با داده ساختارهای ساده و نحوه‌ی استفاده از آنها است. در قسمت اول به شما یک قالب از سه داده ساختار `stack`، `queue`، `linkedList` داده می‌شود و انتظار می‌رود 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
- طراح: میثاق محقق

توضیح توابع:

```
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 می‌کند و آنرا برمی‌گرداند  
  
def isEmpty(self) -> از جنس بولین بوده و درستی خالی بودن صف را برمی‌گرداند  
  
def getInOneLine(self) -> تمام عناصر را در یک خط مانند 1 2 3 بر می‌گرداند
```

```
class Stack:
```

```
def __init__(self, size=10) -> کانستراکتور پشته  
  
def isEmpty(self) -> از جنس بولین بوده و درستی خالی بودن پشته را برمی‌گرداند  
  
def push(self, value) -> عنصری را به پشته اضافه می‌کند  
  
def pop(self) -> عنصری از پشته کم می‌کند و آنرا برمی‌گرداند  
  
def put(self, value_) -> عنصری از پشته کم و عنصر دیگری را جایگزین می‌کند  
  
def peek(self) -> عنصر بالای پشته را صرفاً برمی‌گرداند یعنی حذف نمی‌کند  
  
def expand(self) -> سایز تعریف شده (با تعداد عناصر ذخیره شده فرق دارد) را دو برابر می‌کند
```

```

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 لینکد لیست لازم است

class LinkedList:

    def __init__(self) -> کانستراکتور لینکد لیست

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

    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 لینکد لیست ایجاد می‌شود. سپس توابع مشخص شده برای هر کدام صدا زده می‌شوند که همگی در قالب آمده است و توضیح مربوط به هر کدام در pdf تمرین آمده است.

نمونه‌ی ورودی و خروجی ۱

INPUT:

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

OUTPUT:

```
False
3
1 2 3
1
2
3
True
0
```

نمونه‌ی ورودی و خروجی ۲

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: کارفرمای بی‌ملاحظه (۲۵ نمره)

- محدودیت زمان: ۳ ثانیه
- محدودیت حافظه: ۱۵۰۰ مگابایت
- طراح: فاطمه کرمی محمدی

علی کارمند شرکت دیوار است و یک کارفرمای بدجنس دارد که می‌خواهد علی را گیج کند. کارفرمای علی به او تعدادی تسک می‌دهد که علی باید آن‌ها را بر حسب اولییتی که کارفرمایش مشخص کرده انجام دهد. از آنجایی که علی کارمند باهوشی است از یک صف استفاده می‌کند که کارهای او را بر حسب اولویت ترتیب‌بندی کند. کارفرمای علی گاهی به او کار جدیدی می‌دهد که باید با فوریت زیادی انجام شود پس علی آن کار را در ابتدای صف خود می‌گذارد. گاهی هم کارفرما کاری به علی می‌گوید که خیلی مهم نیست و می‌تواند در اولویت آخر باشد پس علی آن کار را در انتهای صف خود می‌گذارد. اما کارفرمای بدجنس ممکن است ناگهان به علی بگوید که اولویت تمام کارهایت را برعکس کن یعنی کاری که بیشترین اولویت را داشته از الان به بعد کمترین اولویت را دارد و کاری که اصلاً مهم نبوده از الان بیشترین اولویت را دارد. همچنین علی فقط زمانی کار انجام می‌دهد که کارفرما به او دستور دهد. کارفرما ممکن است بگوید کاری که اولویت بیشتری دارد را انجام دهد یا ممکن است (چون قصد اذیت دارد) بگوید کاری که کمترین اهمیت را دارد انجام دهد.

ورودی

در خط اول تعداد درخواست‌های کارفرما (Q) می‌آید. هر درخواست به یکی از شکل‌های زیر است:

back: کار با کمترین اهمیت را انجام بده.

front: کار با بیشترین اهمیت را انجام بده.

reverse: اولویت همه کارها را برعکس کن.

push_back N: کار N با کمترین اهمیت را به صف اضافه کن.

push_front N: کار N با بیشترین اهمیت را به صف اضافه کن.

در Q خط بعدی درخواست‌های کارفرما می‌آید.

$$1 \leq Q \leq 10^5$$

$$1 \leq N \leq 100$$

خروجی

به ازای هر کاری که انجام می شود آن کار از صف حذف شده و در یک خط چاپ می شود. در صورتی که دستور back یا front داده شد ولی کاری برای انجام دادن نبود "No job" چاپ می شود.

نمونه‌ی ورودی و خروجی ۱

INPUT :

```
12
push_front 12
back
reverse
push_back 16
push_back 92
push_front 46
back
reverse
front
push_back 60
reverse
front
```

OUTPUT :

```
12
92
16
60
```

نمونه‌ی ورودی و خروجی ۲

INPUT:

```
15
push_front 30
front
back
reverse
back
reverse
push_front 55
push_back 78
push_back 23
front
front
reverse
push_back 60
reverse
front
```

OUTPUT:

```
30
No job
No job
55
78
60
```


مسئله‌ی سوم: اردک‌های گرسنه (۲۵ نمره)

- محدودیت زمان: ۰.۵ ثانیه
- محدودیت حافظه: ۶۴ مگابایت
- طراح: فاطمه کرمی محمدی

بیلی، میلی، گیلی و جیلی چهار اردک هستند که هر کدام به تازگی یک بچه اردک به دنیا آوردند. آن‌ها برای اینکه بتوانند شکم بچه‌های خود را سیر کنند باید طوری وارد مرغداری خانم لیلی شوند که نگهبان مرغداری فکر کند آن‌ها مرغ هستند و به آن‌ها اجازه عبور دهد.

هر گروه مرغی از تعدادی مرغ و بچه‌ی هر مرغ تشکیل می‌شود که اگر هر مرغ را با حرف M نشان دهیم بچه آن را با حرف m نشان می‌دهیم. حال رشته‌های مناسب عبور مرغی برای گروه مرغ‌ها به این شکل است که یا رشته خالی است، یا از اتصال دو رشته مناسب عبور مرغی تشکیل شده و یا از یک رشته مناسب عبور مرغی که قبل از آن یک مرغ بالغ (M)، و بعد از آن، بچه آن مرغ بالغ (m) آمده تشکیل شده است. مثلاً رشته‌های MmNnTt و ABba رشته‌های مناسبی برای عبور مرغی هستند (رشته اول از اتصال سه رشته مناسب عبور مرغی تشکیل شده و رشته دوم یک رشته مناسب عبور مرغی است (Bb) که قبل از آن مرغ A و بعد از آن بچه او یعنی a آمده) اما رشته‌های xX و JUju رشته‌های مناسبی برای عبور مرغی نیستند (در رشته اول مرغ کودک قبل از مادر خود آمده و در رشته دوم جوجه j قبل از جوجه u آمده).

حال این چهار اردک (B, M, G, J) و کودکانشان (b, m, g, j) در یک صف قرار می‌گیرند. شما باید مشخص کنید زیررشته‌های مشخص شده از این صف، قابل عبور از در مرغداری هستند یا خیر.

ورودی

در خط اول ورودی رشته حاوی ترتیب قرارگیری اردک‌ها به طول L می‌آید که حاوی حروف B, M, G, J و b, m, g, j است. در خط بعد عدد Q می‌آید که تعداد زیررشته‌های رشته اصلی است که باید چک شود. در خط i ام از Q خط بعدی دو عدد صحیح li و ri با فاصله از هم می‌آید که شماره اولین و آخرین حرف زیررشته‌ای است که باید چک شود.

$$1 \leq L \leq 300$$

$$1 \leq Q \leq 100$$

$$1 \leq li \leq ri \leq L$$

خروجی

رشته‌ای حاوی اعداد 0 و 1 به طول Q که رقم i ام آن 1 است در صورتی که زیررشته حاوی حروف li تا ri رشته مناسب عبور مرگی باشد و در غیر این صورت رقم i ام 0 است.

نمونه‌ی ورودی و خروجی

INPUT:

mbBgjGJjgBbmMm

4

2 3

6 9

1 14

13 14

OUTPUT:

0101

توضیح: زیر رشته اول bB است که مرغ کودک قبل از مادر خود آمده پس مناسب نیست. زیر رشته دوم GJjg است که J، رشته مناسبی است که قبل آن G و پس از آن g آمده است. زیر رشته سوم در اصل کل رشته است که جوجه m در ابتدا آمده پس رد می‌شود. زیر رشته چهارم نیز Bb است که مانند زیر رشته اول قابل قبول است.

مسئله‌ی چهارم: مدرسان کمپانی (۲۵ نمره)

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- طراح: سروش صحرایی

مدرسان کمپانی که تشنه پول است می‌خواهد یک تبلیغ خیلی خیلی خیلی گنده در خیابان آزادی نصب کند. خیابان آزادی متشکل از تعدادی ساختمان است که ارتفاع i امین آن‌ها برابر h_i است. بئر مدرسان کمپانی باید به شکل یک مستطیل باشد که اضلاع آن در موازات محورها قرار دارند و کل مساحت بئر روی ساختمان‌ها قرار دارد. به مدرسان کمپانی کمک کنید مساحت بزرگ‌ترین بئر ممکن را پیدا کنند.

ورودی

در خط اول تنها عدد n به شما داده می‌شود. در خط بعد n عدد که معرف ارتفاع ساختمان‌ها هستند به شما داده می‌شود.

$$1 \leq n \leq 2 * 10^5$$
$$0 \leq h_i \leq 10^9$$

خروجی

یک عدد چاپ کنید که برابر بزرگ‌ترین مساحت ممکن برای بئر است.

نمونه‌ی ورودی و خروجی

INPUT:

8
4 1 5 3 3 2 4 1

OUTPUT:

10

توضیح: کفایت یک بئر با ارتفاع ۲ و طول ۵ بین ساختمان‌های سوم تا هفتم نصب کنیم.

نکات تکمیلی

- هدف این تمرین یادگیری شماسست. 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 خواهد شد).