



تمرین کامپیوتری شماره ۲

ساختمان داده - پاییز ۱۴۰۲

دانشکده مهندسی برق و کامپیوتر

طراح تمرین: سیده زینب پیش‌بین،

مهلت تحویل: ۱۴۰۲/۹/۳ (۱۲ شب)

مدرس: دکتر هشام فیلی

میثاق محقق

مقدمه

این تمرین کامپیوتری برای آشنایی با داده‌ساختارهای ساده و نحوه‌ی استفاده از آنها است. در قسمت اول به شما یک قالب از سه داده‌ساختار `stack`، `queue`، `linkedList` داده می‌شود و انتظار می‌رود که با توجه به مطالب گفته‌شده 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 می گرداند  
  
    def enqueue(self, value) -> عنصری را به صف اضافه می کند  
  
    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  
  
  
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 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 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.

نمونه‌ی ورودی و خروجی 1

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 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: عدد مایکل انژلو (۲۵ نمره)

- محدودیت زمان: 1 ثانیه
- محدودیت حافظه: 256 مگابایت
- طراح: سیده زینب پیش‌بین



مایکل انژلو یک عدد به صورت $x + 0.5$ در ذهن خود دارد، به طوری که x عددی صحیح بین 0 تا N است. رافائل سعی دارد عدد مایکل انژلو را حدس بزند. او مجاز است سوال خود را تنها به فرم "آیا عدد y از عدد تو بزرگ‌تر است یا کوچک‌تر" به صورتی که y عددی صحیح از 1 تا N است، بپرسد؛ و در پاسخ مایکی به صورت "بز" (خلاصه بزرگ‌تر) یا "کو" (خلاصه کوچک‌تر) پاسخ می‌دهد. رافائل تصمیم گرفته از یک استراتژی برای یافتن عدد مایکی استفاده کند. پیش از هر گونه حدس، او یک جایگشت از 1 تا N را در نظر می‌گیرد. سپس اعداد این جایگشت را به ترتیب به عنوان حدس خود مطرح می‌کند. با این حال، رافائل از حدس‌هایی که پاسخ آن را می‌داند، عبور می‌کند. ثابت می‌شود با این استراتژی، رافائل بدون توجه به جایگشت مدنظرش، به x یکسانی می‌رسد. اگر تمام پاسخ‌های مایکل انژلو را به هم بچسبانیم و رشته S را بسازیم، تعداد دفعاتی که مایکی گفته است "بزکو" برابر تعداد زیر رشته‌های به طول 4 برابر "بزکو" است. مایکل انژلو استراتژی رافائل را می‌داند. علاوه بر این، با استفاده از قدرت‌های نینجایی‌اش، جایگشت مدنظر رافائل را نیز می‌داند. با این حال، مایکل انژلو هنوز عدد مورد نظر خود را انتخاب نکرده است. به مایکل انژلو کمک کنید تا مطلع شود به ازای هر x چه تعداد "بزکو" خواهد گفت.

ورودی

خط اول عدد N است. ($1 \leq N \leq 2 \cdot 10^5$)
خط دوم شامل N عدد می باشد، که جایگشت مدنظر رافائل است.

خروجی

به ازای هر x از 0 تا N ، تعداد دفعاتی که مایکل انژلو "بزکو" خواهد گفت را چاپ کنید.

نمونه‌ی ورودی و خروجی

INPUT :

5

5 1 2 4 3

OUTPUT :

0

1

1

2

1

0

مسئله‌ی سوم: نقاشی داناتلو (۲۵ نمره)

- محدودیت زمان: 1 ثانیه
- محدودیت حافظه: 256 مگابایت
- طراح: سیده زینب پیش‌بین



داناتلو فرهیخته‌ترین عضو لاک‌پشت‌های نینجا است. او علاقه زیادی به علم و هنر دارد، و در اوقات فراغت خود وقت بسیاری را به نقاشی کردن اختصاص می‌دهد.

از آنجایی که ذهن داناتلو فرای لاک‌پشت‌های عادی است، تصمیم گرفت نقاشی‌هایش نیز همانند ذهن خودش منحصر به فرد باشند. از همین رو تصمیم گرفت نقاشی‌هایش را به صورت خطی ترسیم کند.

او در ابتدا بوم خطی خود را به N خانه تقسیم می‌کند و در هر مرحله قسمتی از آن را با یکی از N رنگ مجموعه‌اش رنگ‌آمیزی می‌کند. او در هر مرحله چند بازه از بوم را رنگ‌آمیزی می‌کند، به طوری که هر بازه تک رنگ است و همچنین در هر مرحله از هر رنگ حداکثر یک بازه رنگ‌آمیزی می‌شود.

داناتلو ارزش زیادی برای وقت خود قائل است و اگر زمان هر مرحله یک دقیقه باشد، از شما می‌خواهد کمترین زمانی که صرف رسم طرح موردنظرش خواهد کرد، چند دقیقه خواهد بود؟

ورودی

در خط اول ورودی N خواهد بود و N خط بعدی، هر کدام عددی در بازه 0 تا N ، نشان دهنده رنگ خانه‌ها (0 به معنی خانه خالی و رنگ نشده) در طرح داناتلو است. ($1 \leq N \leq 10^5$)

خروجی

کمترین زمانی که داناتلو صرف رسم طرح موردنظرش خواهد کرد.

نمونه‌ی ورودی و خروجی

INPUT:

7
0
1
4
5
1
3
3

OUTPUT:

2