



## تمرین کامپیوتری شماره ۲

ساختمان داده - پاییز ۱۴۰۳

دانشکده مهندسی برق و کامپیوتر

طراحان تمرین: آرین باستانی، میثاق

مدرس: دکتر هشام فیلی، مهندس

محقق، علی عطاءاللهی

مصطفی امیری

### 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.

## مسئله اول: دستگرمی (۲۰ نمره)

● طراح: میثاق محقق، علی عطاءاللهی

توضیح توابع:

```
class Queue:
```

```
def __init__(self) -> کانستراکتور صف  
  
def enqueue(self, value) -> عنصری را به صف اضافه می کند  
  
def dequeue(self) -> عنصری از صف کم می کند و آن را برمی گرداند  
  
def size(self) -> تعداد عناصر ذخیره شده در صف را برمی گرداند  
  
def empty(self) -> از جنس بولین بوده و درستی خالی بودن صف را برمی گرداند  
  
def one_line_str(self) -> تمام عناصر را در یک خط مانند 1 2 3 برمی گرداند
```

```
class Stack:
```

```
def __init__(self, capacity=10) -> کانستراکتور پشته  
  
def push(self, value) -> عنصری را به پشته اضافه می کند  
  
def pop(self) -> عنصری از پشته کم می کند و آن را برمی گرداند  
  
def put(self, value) -> عنصری از پشته کم و عنصر دیگری را جابگزین می کند  
  
def peek(self) -> عنصر بالای پشته را صرفاً برمی گرداند و حذف نمی کند  
  
def expand(self) -> ظرفیت تعریف شده (با تعداد عناصر ذخیره شده فرق دارد) را دو برابر می کند  
  
def capacity(self) -> ظرفیت پشته را برمی گرداند  
  
def size(self) -> تعداد عناصر ذخیره شده در صف را برمی گرداند  
  
def empty(self) -> از جنس بولین بوده و درستی خالی بودن پشته را برمی گرداند
```

```

def one_line_str(self) -> تمام عناصر را در یک خط مانند 1 2 3 برمی گرداند

class Node:

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

class LinkedList:

    def __init__(self) -> کانستراکتور لینکد لیست

    def insert_front(self, value) -> در ابتدای لیست عنصری را وارد می کند

    def insert_back(self, value) -> در انتهای لیست عنصری را وارد می کند

    def reverse(self) -> لیست را برعکس می کند

    def one_line_str(self) -> تمام عناصر را در یک خط مانند 1 2 3 برمی گرداند

```

## توضیح کد قالب

کد قالب داده شده در کنار صورت پروژه، شامل چند کلاس و تابع می باشد که کافی است توابع مشخص شده در بالا را در آن کامل کنید و نیازی به یادگیری مابقی قالب نیست. توجه کنید که در توابع بالا از `print` استفاده نکنید و مثلاً برای توابع `one_line_str`، باید یک مقدار استرینگ را ریترن کنید.

## ورودی

با توجه به قالب داده شده، ابتدا یک یا چند آبجکت از نوع پشته و صف و لینکد لیست ایجاد می شود. سپس توابع مشخص شده برای هر کدام از آبجکت ها صدا زده می شوند و در صورت داشتن خروجی چاپ می شوند. همه توابع استفاده شده در قالب آمده است و توضیح مربوط به هر کدام در بالا آورده شده است.

توجه کنید که گرفتن و پردازش ورودی و چاپ خروجی در قالب انجام شده است و شما فقط توابع کلاس های پشته، صف و لینکد لیست را کامل می کنید.

## نمونه ورودی و خروجی ۱

### INPUT:

```
make queue q1
call q1.enqueue(1)
call q1.enqueue(2)
call q1.enqueue(3)
call q1.empty()
call q1.size()
call q1.one_line_str()
call q1.dequeue()
call q1.dequeue()
call q1.dequeue()
call q1.empty()
call q1.size()
```

### OUTPUT:

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

## نمونه ورودی و خروجی ۲

### INPUT:

```
make stack s1
call s1.push(1)
call s1.push(2)
call s1.push(3)
call s1.empty()
call s1.size()
call s1.one_line_str()
call s1.pop()
call s1.pop()
call s1.pop()
call s1.empty()
call s1.size()
```

### OUTPUT:

```
False
3
1 2 3
3
```

```
2
1
True
0
```

## مسئله دوم: مرد ری فکتور (۲۰ نمره)

● طراح: میثاق محقق، علی عطاءاللهی

رئیس سازمان مسعود از کد پروژه ایراد گرفته و گفته که متن کد در صفحه گوشی همراه او جا نمی‌شود! این مشکل به خاطر indentation زیاد و تو رفتگی‌های کد هست که اگر از فلسفه دیدن کد در گوشی بگذریم، حرف بدی هم نیست! برای همین مسعود تصمیم گرفت که کدش یک لایه indent بیشتر نداشته باشد! برای این کار مسعود می‌خواهد هر scope جدیدی که داریم (که منجر به تورفتگی جدیدی می‌شود) را با یک فراخوانی تابع جایگزین کند.

### ورودی

در ورودی یک نمونه از کد مسعود که در زبان غیررسمی Painthon زده شده قرار می‌گیرد. سینتکس این زبان به صورت زیر است. همیشه در انتهای هر بلاک `# end_scope` قرار می‌گیرد و همه if-ها برای سادگی شرط True دارند. برای درک بهتر این زبان به ورودی و خروجی‌های نمونه رجوع کنید.

```
# Function Definition
def function_name():
    # content
    # end_scope
```

```
# If Statement
if True:
    # content
    # end_scope
```

```
# Simple Statements
statement_text
```

## خروجی

در خروجی، نسخه ری فکتور شده کد مسعود می آید که به ازای هر if statement یک تابع داریم و خود if-ها با فراخوانی تابع جایگزین شده اند.

## توضیح کد قالب

کد قالب در کنار صورت پروژه به شما داده شده است. این کد قالب ورودی را خوانده و آن را پردازش می کند. شما فقط کافیست که تابع print\_result را پیاده سازی کنید و نیازی نیست به هیچ جای دیگر کد دست بزنید. کارکرد کد به این صورت است که کلاس Runner ورودی را خوانده و سپس بنا بر نوع statement (که می تواند تعریف تابع، گزاره شرطی، statement ساده و یا end scope باشد) متد متناظر کلاس Handler را صدا می زند تا در آنجا پردازش بشود.

در نهایت همه statement-ها به عنوان یک Node ذخیره می شوند که هر Node می تواند چندین فرزند Node داشته باشد. مثلاً فرزندان یک IfStatement شامل محتوای داخل آن گزاره شرطی است. SimpleStatement-ها فرزندی ندارند. شما در تابع print\_result یک ورودی program\_node می گیرید که فرزندان آن FunctionStatement هستند. حال شما باید با استفاده از یک صف، کد ری فکتور شده را در این تابع پرینت کنید.

## نمونه ورودی و خروجی ۱

```
INPUT:
def main():
    first_statement
    if True:
        inner_statement
        # end_scope
    # end_scope
```

```

OUTPUT:
def main():
    first_statement
    condition_1()

def condition_1():
    inner_statement

```

همانطور که می‌بینیم، گزاره شرطی به یک تابع تبدیل شده و از indentation زیاد کد جلوگیری شده است.

فیلد node.name همه گزاره‌های شرطی در کد قالب، مقدار 'condition' را دارند که ما اینجا باید با استفاده از یک شمارنده مقدار 1\_ را به انتهای نام تابع اضافه کنیم تا نام آنها یکسان نباشد.

## نمونه ورودی و خروجی ۲

```

INPUT:
def myfunc():
    statement1
    statement2
    statement3

    if True:
        i_am_first_if

        if True:
            this_is_inner_if
            this_is_inner_if2
            if True:
                very_nested
                # end_scope
            # end_scope

        i_am_first_if_end
        # end_scope

    if True:
        another_if
        # end_scope

    statement100
    # end_scope

```

```

OUTPUT:
def myfunc():
    statement1
    statement2

```

```

statement3
condition_1()
condition_2()
statement100

def condition_1():
    i_am_first_if
    condition_3()
    i_am_first_if_end

def condition_2():
    another_if

def condition_3():
    this_is_inner_if
    this_is_inner_if2
    condition_4()

def condition_4():
    very_nested

```

در این مثال نحوه هندل شدن شرط‌های nested را می‌بینیم که مانند قبل است. از آنجا که برای پیاده‌سازی از یک صف استفاده می‌کنید نحوه شماره‌گذاری توابع نیز به شکلی که در خروجی مشخص است در خواهد آمد.

نکته: در صورت وجود بیش از یک تابع، کد مربوط به ری‌فکتور کردن یک تابع را در چرخه به ازای هر تابع به ترتیب اجرا کنید.

## مسئله سوم: نبرد فاضلابی (۲۰ نمره)

● طراح: آرین باستانی

تعدادی موش در لوله فاضلاب به صورت صف قرار گرفته و هرکدام به جهت راست یا چپ حرکت می‌کنند. در صورت برخورد دو موش با یکدیگر، آنها با هم می‌جنگند و موش با قدرت کمتر می‌میرد (اگر قدرت دو موش برابر بود، هردو می‌میرند). پس از اتمام این نبردها، موش‌های باقی‌مانده به سمت مخفی‌گاه استاد اسپلینتر حرکت می‌کنند و در نهایت با او می‌جنگند.

استاد اسپلینتر از شما می‌خواهد تا به او کمک کنید که بفهمد با چه موش‌هایی خواهد جنگید تا بتواند خود و لاک‌پشت‌های نینجا را برای این نبرد آماده کند. صف موش‌های داخل لوله، به صورت یک لیست به شما داده می‌شود و هر



موش با یک عدد نشان داده می‌شود (عدد منفی نشان‌دهنده‌ی حرکت به سمت چپ و عدد مثبت نشان‌دهنده‌ی حرکت به راست می‌باشد و همچنین مقدار عدد برابر مقدار قدرت موش است).

## ورودی

ورودی به صورت یک خط که نشان‌دهنده‌ی لیست موش‌ها (Rats) با طول  $L$  داده می‌شود. لیست با '[' شروع شده و با ']' به اتمام می‌رسد. همچنین اعضای لیست با ',' از هم جدا می‌شوند.

$$2 \leq L \leq 10^4$$

$$-10^3 \leq Rats[i] \leq 10^3$$

$$Rats[i] \neq 0$$

## خروجی

همانند ورودی، یک لیست از موش‌ها در یک خط باید چاپ شود (لیست با '[' شروع شده و با ']' به اتمام می‌رسد و همچنین اعضای لیست با ',' از هم جدا می‌شوند). هر موش نیز باید مشابه ورودی، با یک عدد نشان داده شود.

### نمونه ورودی و خروجی ۱

INPUT:  
[7, 8, -7]

OUTPUT:  
[7, 8]

توضیح: موش 8 با موش 7- برخورد کرده و موش 7- می‌میرد.

### نمونه ورودی و خروجی ۲

INPUT:  
[5, -5]

OUTPUT:  
[ ]

توضیح: موش 5 با موش 5- برخورد کرده و چون قدرت هر دو برابر است، هر دو می‌میرند.

### نمونه ورودی و خروجی ۳

INPUT:  
[12, 5, -10]

OUTPUT:  
[12]

توضیح: موش 5 با موش 10- برخورد کرده و موش 5 می‌میرد و سپس موش 12 با موش 10- برخورد می‌کند و 10- می‌میرد.

## مسئله چهارم: کتاب شبانه (۲۰ نمره)

● طراح: آرین باستانی

دیر وقت است اما علی خوابش نمی‌برد، او تصمیم می‌گیرد که از کتابخانه کتابی بردارد و بخواند. در یک طبقه از کتابخانه کتاب‌های علی قرار گرفته است و او یک برچسب عددی به معنای شدت جذابیت کتاب روی هر کدام چسبانده است. از آنجایی که علی نمی‌خواهد بقیه را بیدار کند چراغ را روشن نمی‌کند و با چراغ قوه از سمت چپ ردیف کتاب‌هایش شروع به جستجو می‌کند و با توجه به نور چراغ قوه، در هر لحظه می‌تواند  $k$  کتاب را ببیند و از آنجایی که می‌خواهد در اتاقش با وجود روشنایی و آرامش بیشتر کتاب خود را انتخاب کند، در هر بار دیدن کتاب‌ها، از بین  $k$  کتابی که می‌بیند بهترین کتاب را برمی‌دارد تا به اتاق برود (فرض کنید به ازای هر کتاب، بی‌نهایت کتاب دیگر با همان امتیاز پشت آن قرار دارد و با برداشتن یک کتاب، کتاب دیگری با همان امتیاز جایگزین آن می‌شود) و سپس یک قدم به راست حرکت می‌کند تا بتواند کتاب بعدی را نیز ببیند (سمت چپی‌ترین کتاب از دید او حذف می‌شود).

لیست کتاب‌هایی که علی به اتاق خود می‌برد را با رعایت ترتیب پیدا کنید.

### ورودی

در خط اول ورودی، کتاب‌های علی در کتابخانه به صورت یک لیست (Books) با طول  $L$  داده می‌شود (لیست با '[' شروع شده و با ']' به اتمام می‌رسد. همچنین اعضای لیست با ',' از هم جدا می‌شوند).

در خط دوم ورودی،  $k$  داده می‌شود.

$$1 \leq L \leq 10^5$$

$$-10^4 \leq Books[i] \leq 10^4$$

$$1 \leq k \leq L$$

## خروجی

برابر است با لیست کتاب‌هایی که علی به اتاقش می‌برد. این لیست در یک خط چاپ می‌شود (لیست با '[' شروع شده و با ']' به اتمام می‌رسد. همچنین اعضای لیست با ',' از هم جدا می‌شوند).

## نمونه ورودی و خروجی

### INPUT:

[2, -1, 3, 12, -4, 10, 2, 34]  
3

### OUTPUT:

[3, 12, 12, 12, 10, 34]

توضیح:

$[2, -1, 3, 12, -4, 10, 2, 34] \rightarrow [2, -1, 3] \rightarrow \max([2, -1, 3]) \rightarrow 3$   
 $[2, -1, 3, 12, -4, 10, 2, 34] \rightarrow [-1, 3, 12] \rightarrow \max([-1, 3, 12]) \rightarrow 12$   
 $[2, -1, 3, 12, -4, 10, 2, 34] \rightarrow [3, 12, -4] \rightarrow \max([3, 12, -4]) \rightarrow 12$   
 $[2, -1, 3, 12, -4, 10, 2, 34] \rightarrow [12, -4, 10] \rightarrow \max([12, -4, 10]) \rightarrow 12$   
 $[2, -1, 3, 12, -4, 10, 2, 34] \rightarrow [-4, 10, 2] \rightarrow \max([-4, 10, 2]) \rightarrow 10$   
 $[2, -1, 3, 12, -4, 10, 2, 34] \rightarrow [10, 2, 34] \rightarrow \max([10, 2, 34]) \rightarrow 34$

## مسئله پنجم: شهر وارونه (۲۰ نمره)

● طراح: آرین باستانی

روزی روزگاری در یک شهر کوچک، خانه‌هایی وجود داشتند که هرکدام یک شماره به خود اختصاص داده بودند و با راه‌هایی به خانه‌های دیگر می‌رسیدند. مردم این شهر همیشه با نظمی خاص در کنار هم زندگی می‌کردند؛ به ترتیب از خانه‌ی اول تا خانه‌ی آخر، یکی یکی به هم متصل بودند.

روزی، پادشاه شهر، «آقای معکوس»، تصمیم گرفت که نظم این شهر را کمی تغییر دهد. او می‌خواست خانه‌های مردم را گروه‌بندی کند و ترتیب خانه‌های هر گروه را معکوس کند. به این ترتیب مردم می‌توانستند همسایه‌های جدیدی پیدا کنند و تجربه‌ای تازه از شهرشان داشته باشند.

آقای معکوس برای اجرای این نقشه، هر دفعه که تصمیم به عمل گرفت، مقدار عدد  $x$  را انتخاب کرده و خانه‌ها را به گروه‌هایی با اندازه  $x$  می‌شکند و خواستار او معکوس شدن ترتیب هر گروه است. همچنین اگر پس از گروه‌بندی، چند خانه باقی ماندند که تعداد آنها از  $x$  کمتر بود، او خواستار به هم زدن ترتیب آنها نیست.

آقای معکوس شهر را (خانه‌ها و راه‌های بین آنها) و مقدار عدد  $x$  به شما می‌دهد و شما باید شهر معکوس شده را حساب کرده و به او بدهید (تغییر مقدار داخل گره‌ها مجاز نیست و فقط می‌توانید خود گره‌ها را تغییر دهید).

### ورودی

ورودی تنها در یک خط داده می‌شود که اول، لیست خانه‌ها ( $n$  تا خانه) داده می‌شود (لیست با '[' شروع شده و با ']' به اتمام می‌رسد و با ',' از هم جدا می‌شوند) و سپس مقدار  $x$  به صورت " $x = \text{some number}$ " داده می‌شود (خانه‌ها و عدد  $x$  با " و " از هم جدا می‌شوند).

$$1 \leq x \leq n \leq 5 \times 10^3$$

$$0 \leq \text{Node.value} \leq 10^3$$

## خروجی

در یک خط باید چاپ شود که برابر با لیست خانه‌ها پس از معکوس کردن است (لیست با '[' شروع شده و با ']' به اتمام می‌رسد و اعضای لیست با ',' از هم جدا می‌شوند).

### نمونه ورودی و خروجی 1

#### INPUT:

[1,2,3,4,5,6,7,8], x = 3

#### OUTPUT:

[3,2,1,6,5,4,7,8]

**توضیح:** 8 خانه داریم و آنها به دو گروه {1,2,3} و {4,5,6} دسته بندی می‌شوند. و سپس آنها را معکوس می‌کنیم (خانه‌های 7, 8 داخل هیچ گروهی جای نمی‌گیرند و ترتیبشان به هم نمی‌خورد).

### نمونه ورودی و خروجی 2

#### INPUT:

[1,2,3,4,5,6,7,8], x = 2

#### OUTPUT:

[2,1,4,3,6,5,8,7]

**توضیح:** 8 خانه داریم و آنها به چهار گروه {1,2} و {3,4} و {5,6} و {7,8} دسته بندی می‌شوند و آنها را معکوس می‌کنیم.

## 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 صورتی که تست‌های تمامی سوالات پاس بشوند و نمره آنها کامل شود، ۱۰ نمره امتیازی اعمال می‌شود (نمره ۱۰۰، ۱۱۰ خواهد شد).