# PS/DS questions

## Question 1:

**Level**: **SDE-1** or **SDE-2**

~~Design me a parking lot management system. What classes will you come up with? What methods will each class expose?~~

## Question 2:

**Level**: **SDE-1** or **SDE-2**

You are given an array of integers which was previously sorted but has since been rotated (to left or right) an unknown number of times. Given a number find if it exists in the array.

## Question 3:

**Level**: **SDE-1** or **SDE-2**

How would you design a stack which, in addition, to `push()` and pop(), also has a function `min()` which returns the minimum element? `push()`, `pop()` and `min()` should all operate in `O(1)` time.

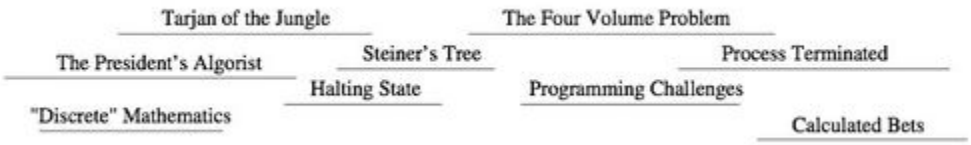
## Question 4:

**Level**: **SDE-1** or **SDE-2** or **SDE-3**

Imagine you are a highly-in-demand actor, who has been presented with offers to star in n different movie projects under development. Each offer comes specified with the first and last day of filming. To take the job, you must commit to being available throughout this entire period. Thus you cannot simultaneously accept two jobs whose intervals overlap.

For an artist such as yourself, the criteria for job acceptance is clear: you want to make as much money as possible. Each of these movies pays the same fee per movie.

Come up with an algorithm to help to choose `r` movies from n movies which maximize profit.




## Question 5:

**Level**: **SDE-1** or **SDE-2** or **SDE-3**

Here is a table showing stock price over the course of some days.

| Day | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Stock price | 20 | 18 | 19 | 24 | 16 | 17 | 17 | 15 | 18 | 19 |

Compute the best days to buy the stock and sell the stock (i.e. the objective is to maximize profits).

## Question 6:

**Level**: **SDE-1** or **SDE-2**

Given pwd (present working directory) and a relative path, print the absolute path of the given relative path.

Hint: pwd will be an absolute path. Need to handle only ".." element.

*Example*:

Input: `pwd /Users/ar/Desktop`

Input: `relative path ../Music`

Output should be `/Users/ar/Music`

---

Question 6:

**Level**: **SDE-1** or **SDE-2**

You are given a list. Elements of the list can be numbers or other such lists. Return a list containing all the numbers in the original list.

Example: Input {1, {2,3}, 4, {5, {6, 7}, 8, {9, 10}}}. Output {1,2,3,4,5,6,7,8,9,10}

---

Question 7:

**Level** : **SDE-1** or **SDE-2**

Given a binary tree, print its "Right profile view". Can you do this using recursion? Using iteration?

---

Question 8:

Level: **SDE-1** or **SDE-2** or **SDE-3**

Let's discuss how garbage collection works in Java. What is mark & sweep? How does JIT affect GC?

---

Question 9:

Level: **SDE-1** or **SDE-2**

In an integer array, all except two elements are repeated even number of times. Find the two numbers that are repeated an odd number of times. Can you do this without extra data structure?

Example:

Input [12, 12, 63, 63, 3,3,4,4,4,5,5,13,13,13]; Output 4 and 13

---

Question 10:

Level: **SDE-2** or **SDE-3**

~~Reverse a stack without using additional data streuture.~~

---

Question 11:

Level: **SDE-1** or **SDE-2**

Given an unsorted linked list, arrange the items such that consecutive pairs have an opposite relationship.

Example:

Input: `10 20 9 18 2 21`

Output: `10 9 20 18 21 2`

i.e. `10 > 9 < 20 > 18 < 21 > 2`

---

Question 12:

Level: **SDE-1** or **SDE-2**

Given a list of number pairs, List all symmetric pairs.
Here symmetric pairs are any pairs such that for {a,b}, {b,a} exists.

Example:
Input: `[{1,3}, {5,7}, {8,9}, {2,20}, {10,19}, {3,1}, {20,2}]`

Output: `[{1,3}, {2,20}]`

---

## Question 13:

Level: **SDE-1** or **SDE-2**

Assume you run a very popular website. You get thousands of requests a second. At any given point of time you want to know the following metrics.

Number of requests that came in last day `getLastDayCount()`

Number of requests that came in last hour `getLastHourCount()`

Number of requests that came in last minute `getLastMinCount()`

Hints:

1. Assume that a method called `handleRequest()` is invoked when a requests comes in.
2. You need not implement HTTP methods or write any logic related to website. This is a simple PS question.
3. Day is past 24 hrs from this moment (i.e. rolling, not calendar date). Ditto for hour and minute.

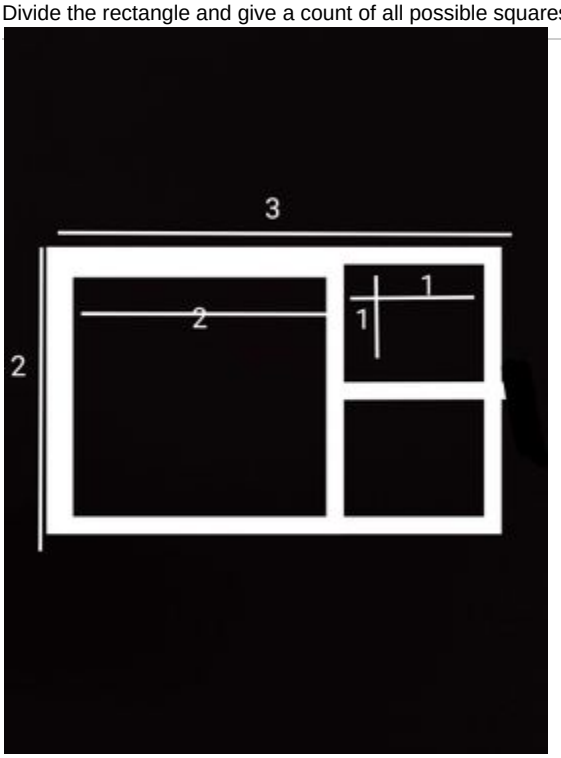
Code up the above three functions i.e. `getLastDayCount(), getLastHourCount(), getLastMinCount()`

---

## Question 14:

Level: **SDE-1** or **SDE-2**

Divide the rectangle and give a count of all possible squares with maximum to the minimum area.




For example, we are given a rectangle with 3 x 2 dimensions. We can divide it with one square of 2 x 2 and two squares of 1 x 1 so for this output will be 3.

---

## Question 15:

Write a function which takes an array and prints the majority element (if it exists), otherwise prints “No Majority Element”. A ***majority element*** in an array A[] of size n is an element that appears more than n/2 times (and hence there is at most one such element). (Ref : Mridul Rai)