Developer

- Use the langage, tools and libraries you are familiar with and you think are appropriate.
- Best would be to post your answers on github

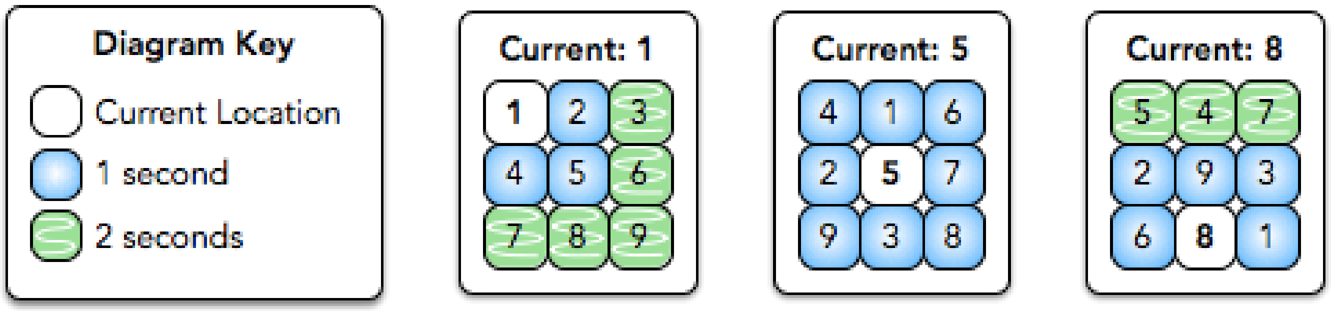
Exercise 1: Keyboard algorithm

You work at a secret organization where you must type a string of numbers into a console using a **3x3** numeric keypad.

Use the following rules to calculate the total amount of time it takes to type a string:

- It takes 0 seconds to move your finger to the first key, and it takes 0 seconds to press the key where your finger is located any number of times
- You can move your finger from one location to any adjacent key in one second.
- Moving to a non-adjacent key is done as a series of moves to adjacent keys

For example:



This diagram depicts the minimum amount of time it takes to move from the current location to all other locations on the keypad.

Function description

Write the function entryTime. The function must return an integer denoting the minimum amount of time it takes to type the string s.

entryTime has the following parameters :
s : the string to type on the keyboard
keypad: a string of 9 digits where each group of 3 digits represents a row on the keypad of the day, in order.

Use the language of your choice.

Function constraints

- $L \leq |\mathbf{s}| \leq 10^5$
- $|\mathbf{keypad}| = 9$
- **keypad**[i] ∈ [1-9]

Sample

Sample input

423692
923857614

Sample Output

8

Explanation

The keypad looks like this:

9	2	3
8	5	7
6	1	4

We calculate the time it takes to type **s** = 423692 as follows:

- 4: We start at this number so it takes 0 seconds.
- 2: It takes 2 seconds to move from 4 → 2
- 3: It takes 1 second to move from 2 → 3
- 6: It takes 2 seconds to move from 3 → 6
- 9: It takes 2 seconds to move from 6 → 9
- 2: It tales 1 second to move from 9 → 2

The total time is **2 + 1 + 2 + 2 + 1 = 8**

Bonus : you may use an online website to post your solution.
Suggestion : https://js.do/ (allows you to use babel ES6)

Exercise 2: CSS

Enhanced Flexible Grid

You need to create a *1x4 flexible grid*, that is, there is 1 row and 4 columns. The width of each column is *25% of the window size*. This width must be maintained even if the page is resized. Also make sure that each cell of the grid can contain another 1x4 flexible grid. The border of the grid must be *1px black*.

Write the HTML/CSS that performs the following operations based on the value of window size:

- If the window size is less than *720px*, then the 1x4 flexible grid becomes a *2x2* grid, that is, the 3rd and 4th columns slide down onto the 2nd row.
- If the window size is less than *360px*, then the 1x4 flexible grid becomes *4x1* grid, that is, each column slides under the one before it. The 2nd column slides under the 1st, the 3rd slides under the 2nd, and the 4th slides under the 3rd.

Bonus : you may use online website to post your solution.

Suggestions :

- https://codepen.io/

- https://jsfiddle.net/

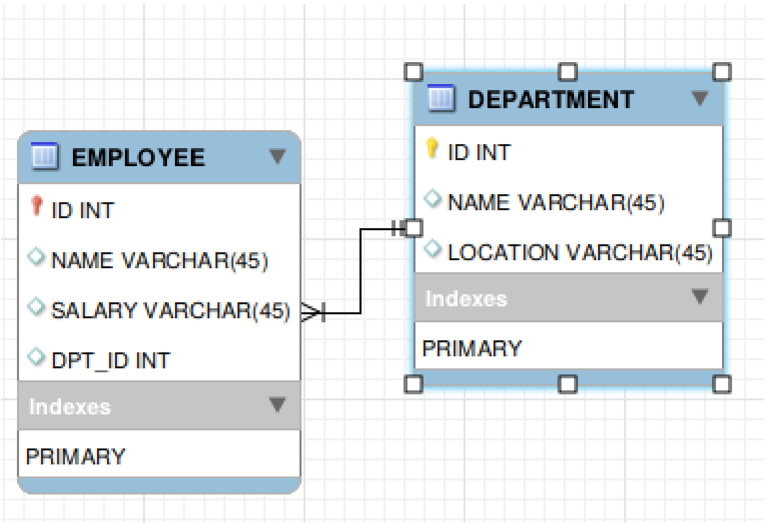
Exercise 3: DATABASE

EMPLOYEES PER DEPARTMENT

A company stores employee and department information in two data tables : *Employee and Department.* **Write a query** to print the respective department name and number of employees in each department of all departments in the Department table (even ones with no current employees).
Sort the results by descending order of the number of employees; if two or more departments have the same number of employees, then sort those departments alphabetically by department name.

Each row of the resulting output must contain the following respective attributes for a department:
1. The name of the department.
2. The number of employees in the department.

DEPARTMENT.NAME COUNT_OF_EMPLOYEES_IN_THE_DEPARTMENT



Schema

Table content examples :

EMPLOYEE

ID	NAME	SALARY	DEPT_ID
1	Candice	4685	1
2	Julia	2559	2
3	Bob	4405	4
4	Scarlet	2350	1
5	Ileana	1151	4

DEPARTMENT

ID	NAME	LOCATION
1	Executive	Sydney
2	Production	Sydney
3	Resources	Cape Town
4	Technical	Texas
5	Management	Paris

Exercise 4: HTTP GET

Wikipedia article

Use an HTTP GET method to retrieve information from wikipedia.

Query:

https://en.wikipedia.org/w/api.php?action=parse&section=0&prop=text&format=json&page=[topic] to get the *topic* Wikipedia article. Print the total number of times that the string *[topic]* appears in the article's *text* field. Note that the search is case-sensitive.

The query response from the website is a JSON object described below:

parse: A JSON object representing the article's parsed web page. It has the following three fields:

1. *title*: The article's title, as specified by the argument passed as *topic*.
2. *pageid*: The article's Page ID.
3. *text*: A JSON object that contains the Wikipedia article as an HTML dump.

Function Description

Create the function *getTopicCount*.
The function must print an integer, the number of times the search term *topic* appears in the returned *text* field.

getTopicCount has the following parameter(s):
***topic**:* a string to query

Sample Case

Sample Input

pizza

*Sample Output **

102

*Note that because this question is dynamically getting the data from Wikipedia, the actual number of occurrences may have changed. 102 is just used as an illustration.

Explanation

The query is
https://en.wikipedia.org/w/api.php?action=parse&section=0&prop=text&format=json&page=pizza

From the response, we get the number of occurences in the text.

Exercise 5: Min max

Function Description

<u>Create the *max_diff function*</u>.
This function is given an array of integers and should return the maximum difference between two elements of this array, given the lowest integer must be place before the largest. Those two elements can happen to not be consecutive.

max_diff has the following parameter(s):
***int_array**:* an array of integers.

Sample Case

Sample Input

[100, 10, 5, 10, 1, 8, 16, 21]

*Sample Output **

20

Explanation

100 is the first element and is larger than any of the f- ollowing elements.
21 is the largest and the biggest.
The lowest element before 21 is 1

21 -1 = 20