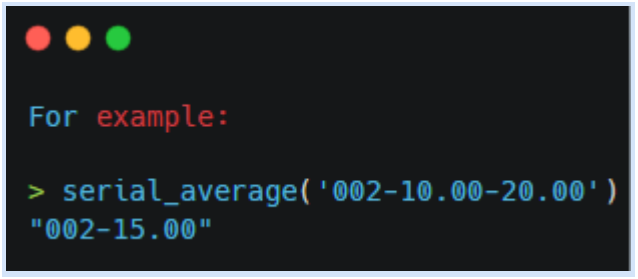
NOTE: You might need to install [ruby gems]

Follow :

- ***OOP approach***
- ***Try to use module if and when necessary***
- ***Have unit testing***

1)
In this challenge, your task is to code a serial_average method which is described below:

It takes a fixed width string in format: SSS-XX.XX-YY.YY. SSS is a three digit serial number, XX.XX and YY.YY are two digit numbers including up to two decimal digits.
It returns a string containing the answer in format SSS-ZZ.ZZ where SSS is the serial number of that input string, and ZZ.ZZ is the average of XX.XX and YY.YY.
All numbers are rounded off to two decimal places.



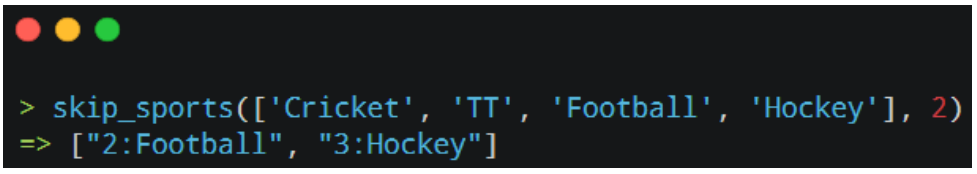
2) For this challenge, your task is to complete the functions using ruby syntax.
Assume the array to be ***arr = [9, 5, 1, 2, 3, 4, 0, -1]***

```ruby
def element_at(arr, index)
    # return the element of the Array variable `arr` at the position `index`
    # arr.at(index) # or
    # arr[index]
end

def inclusive_range(arr, start_pos, end_pos)
    # return the elements of the Array variable `arr` between the start_pos and end_pos (both inclusive)
end

def non_inclusive_range(arr, start_pos, end_pos)
    # return the elements of the Array variable `arr`, start_pos inclusive and end_pos exclusive
end

def start_and_length(arr, start_pos, length)
    # return `length` elements of the Array variable `arr` starting from `start_pos`
end
```

3) In this challenge, create a hash . Now to this hash,You have to add

- A key-value pair [543121, 100] to the hash.
- Retain all key-value pairs where keys are Integers (clue : is_a? Integer)
- Delete all key-value pairs where keys are even-valued.

4) In this challenge, your task is to complete the skip_sports method that takes an sports array and a skip integer and returns an array of all elements except first skip number of items as shown in the example below.



5) In this challenge,perform the following tasks

1. save the given the content to a JSON file:

```
{
  "name": "Nicole Smith",
  "age": 25,
  "salary": 25552.67,
}
```

2. Read from this newly created JSON file and change the name to first name and last name as 2 separate key value pairs.
3. Save the contents to a new JSON file in the same folder having a name as follows:

   <Your name>_<timestamp>.json, timestamp should be in UTC format.

6) In this challenge, we will try to encode and decode a string to base 64 format, and have it written to a file. Perform the following tasks,

- Create a file having the following contents

```
Originally, John Doe was a sham name used to indicate any plaintiff in an
action of ejectment (a legal action to regain property) in civil court [ "
the string that will be base encoded "]. Richard Roe was the counterpart, to
indicate the defendant. These fake names were used in delicate legal matters,
a practice that was abolished in English law in 1852.
```

- Read the content from the file
- Base64 encode the string containing `the string that will be base encoded.` line.
- Write the extracted and encoded string to a new file.
- The file name should be <your name>.txt