Question 1.

Write a function that takes an array of integers and returns that array rotated by N positions using Python or Ruby or JavaScript.

```
For example, if N=2, given the input array given the input array [1, 2,
3, 4, 5, 6]
the function should return [5, 6, 1, 2, 3, 4]
```

In [1]:
```
def rotate_list(arr,n=2):
    """Function rotates the given array using slicing
    n is optional
    @param arr: list of numbers
    @param n: position to rotate the array"""
    return arr[-n:]+arr[:len(arr)-n]

rotate_list([1,2,3,4,5,6],2)
```

Out[1]:
```
[5, 6, 1, 2, 3, 4]
```

Question 2. JavaScript

```
In JavaScript , answer if the following expressions result in true or f
alse (and explain your
answer)
```

- “0” == 0 //true
- "" == 0 //true
- "" == "0" //false

Question 3. Python, Ruby or JavaScript

Most languages have a built in sort method that will sort an array of strings alphabetically.
Demonstrate how to sort an array of strings by the length of each string, shortest strings first.

Hint: clean, small code wins.

In [2]:
```
def sort_strings(list_of_strings,reverse=False):
    '''
    Function to sort the list of the strings based on their lengths
    @param list_of_strings: list of strings
    @param reverse: True sorts the list in reverse order and false in non reverse
    @returns: sorted list of strings
    '''
    return sorted(list_of_strings,key=lambda x: len(x),reverse=reverse)

print('normal sort:', sort_strings(['sam','sa','satyam','saty']))
print('reverse sort: ',sort_strings(['sam','sa','satyam','saty'],True))
```

```
normal sort: ['sa', 'sam', 'saty', 'satyam']
reverse sort:  ['satyam', 'saty', 'sam', 'sa']
```

Question 4. JavaScript

```
function weird(x) {

        var tmp = 3;

        return function(y) {

            return x + y + ++tmp;

        }

}

var funny = weird(2);

var final_answer = funny(10);
```

What is the value of final_answer at the end of this snippet? Please explain your answer

explanation

> this is a *closure*
>
> when `weird(2)` is called it returns inner function with x=2 & tmp = 4 as it is increemented
>
> next when `funny(10)` is called, sets y = 10 and returns sum of all these as 16

Question 5. Python, Ruby or JavaScript

Consider the following pattern:

```
A → D M → P X → A

a → d m → p x → a
```

Now, write a program to solve the following message using Python, JavaScript or Ruby.

```
Vrphwklqj phdqlqjixo
```

Hint: The answer is something meaningful.

In [3]:
```
def decode_string(string):
    '''
    function to decode string using the ascii value
    @param string: string to decode
    @returns: decoded string
    '''
    def inner_map(char):
        return dict(zip(map(chr,list(range(ord(char)+3,ord(char)+26
))+list(range(ord(char),ord(char)+3))),
                        map(chr,range(ord(char),ord(char)+26))))
    return ''.join([dict(**inner_map('A'),**inner_map('a')).get(i,i
) for i in string])

print(decode_string('Vrphwklqj phdqlqjixo'))
```

```
Something meaningful
```

Question 6. Python, Ruby or JavaScript

Use the following paragraphs (in italics below) as input to the program you wrote for Question 5.

It should output a meaningful question. Write a program to solve that question.

> Zulwh d surjudp (lq Sbwkrq, MdydVfulsw ru Uxeb) wr srsxodwh dqg wkhq vruw d udqgrpob glvwulexwhg olvw ri 1 ploolrq lqwhjhuv, hdfk lqwhjhu kdylqj d ydoxh >= 1 dqg <= 100 zlwkrxw xvlqj dqb exlowlq/hawhuqdo oleudub/ixqfwlrq iru vruwlqj. Brxu surjudp vkrxog fduhixoob frqvlghu wkh lqsxw dqg frph xs zlwk wkh prvw hiilflhqw vruwlqj vroxwlrq brx fdq wklqn ri. Surylgh wkh vsdfh dqg wlph frpsohalwb ri brxu dojrulwkp

In [4]:
```
#decoding the question
print(decode_string('''Zulwh d surjudp (lq Sbwkrq, MdydVfulsw ru Ux
eb) wr srsxodwh dqg wkhq vruw d udqgrpob glvwulexwhg olvw ri 1 ploo
lrq lqwhjhuv, hdfk lqwhjhu kdylqj d ydoxh >= 1 dqg <= 100 zlwkrxw x
vlqj dqb exlowlq/hawhuqdo oleudub/ixqfwlrq iru vruwlqj. Brxu surjud
p vkrxog fduhixoob frqvlghu wkh lqsxw dqg frph xs zlwk wkh prvw hii
lflhqw vruwlqj vroxwlrq brx fdq wklqn ri. Surylgh wkh vsdfh dqg wlp
h frpsohalwb ri brxu dojrulwkp'''))
```

```
Write a program (in Python, JavaScript or Ruby) to populate and th
en sort a randomly distributed list of 1 million integers, each in
teger having a value >= 1 and <= 100 without using any builtin/ext
ernal library/function for sorting. Your program should carefully
consider the input and come up with the most efficient sorting sol
ution you can think of. Provide the space and time complexity of y
our algorithm
```

In [5]:
```
import random
records = [random.randint(1,100) for i in range(1000000)]
```

In [6]:
```
def counting_sort(arr, exp1):

    n = len(arr)

    # The output array elements that will have sorted arr
    output = [0] * (n)

    # initialize count array as 0
    count = [0] * (10)

    # Store count of occurrences in count[]
    for i in range(0, n):
        index = (arr[i]//exp1)
        count[ (index)%10 ] += 1

    # Change count[i] so that count[i] now contains actual
    #  position of this digit in output array
    for i in range(1,10):
        count[i] += count[i-1]

    # Build the output array
    i = n-1
    while i>=0:
        index = (arr[i]//exp1)
        output[ count[ (index)%10 ] - 1] = arr[i]
        count[ (index)%10 ] -= 1
        i -= 1

    # Copying the output array to arr[],
    # so that arr now contains sorted numbers
    i = 0
    for i in range(0,len(arr)):
        arr[i] = output[i]

def radix_sort(arr):

    # Find the maximum number to know number of digits
    max1 = max(arr)

    # Do counting sort for every digit. Note that instead
    # of passing digit number, exp is passed. exp is 10^i
    # where i is current digit number
    exp = 1
    while max1//exp > 0:
        counting_sort(arr,exp)
        exp *= 10
```

In [7]:
```
%%time
radix_sort(records)
```

```
CPU times: user 1.98 s, sys: 13.7 ms, total: 2 s
Wall time: 2.01 s
```

Explanation:

> I have used radix sort to sort out the records
>
> it comprises of radix sort and counting sort, giving it the complexity of O(nLog(n))
>
> This algorithm proved to be better than merge sort and heap on 1e6 records sorting them in 1.9 secs.
>
> Radix sort helped me get better time complexity and performance due to its sorting by most significant places in every iteration and follow by counting sort for count and placing the elements

---

Question 7. Python, Ruby or JavaScript

Write a simple program that reads a line from the keyboard and outputs the same line where every word is reversed. A word is defined as a continuous sequence of alphanumeric characters or hyphen (‘-’). For instance, if the input is

> **“We are at Ignite Solutions! Their email-id is careers@ignitesol.com”**

the output should be

> **“eW era ta etingI snoituloS! riehT di-liame si sreerac@losetingi.moc”**

HINT: Read the problem and the example carefully before starting.

In [8]:
```
import re

def reverse_string(string):
    '''
        function to reverse the words in the string
        @params string: string of the size n
        @returns: string with reversed words
    '''
    def rev_word(word):
        '''
        inner function to handle reverse word sequence.
        @params word: word to reverse
        @returns: reversed word
        '''
        parts = re.findall('[A-Za-z0-9/-]+',word)
        for i in parts: word = word.replace(i,i[::-1])
        return word
    return ' ' .join([rev_word(i) for i in string.split()])
```

In [9]:
```
reverse_string('We are at Ignite Solutions! Their email-id is careers@ignitesol.com')
```

Out[9]:
```
'eW era ta etingI snoituloS! riehT di-liame si sreerac@losetingi.moc'
```