Q1.Write a Python program to get the string from the given string where all the occurrence of its first char has been changed to '$,' except first char itself?

Sample String: 'prospect'

Expected Result : 'pros$ect'

In [0]:
```
def change_occurences(str):
  return str[0] + str[1:].replace(str[0], '$')

print(change_occurences('prospect'))
```

```
pros$ect
```

Q2.Write a Python program to get the single string from the two given strings, and separated by the space and swap the first two characters of each string?

Sample String : 'abc', 'xyz'.

Expected Result: 'xyc abz'

In [0]:
```
def concat_swap(str1, str2):
  str1, str2 = str2[0] + str1[1:], str1[0] + str2[1:]
  return "{0} {1}".format(str1, str2)

concat_swap('abc', 'xyz')
```

Out[20]:
```
'xbc ayz'
```

Q3. Write thePython program to add 'ing' at the end of thegiven string (length of the string shouldbe at least 3). Ifgiven string already ends with 'ing,'then add 'ly' instead. Ifstring length of the given string is less than 3, leave it unchanged?

Sample string:'abc'

Expected result:'abcing'

Sample string:'string'

Expected result: 'stringly'

```
In [84]: def append_ing_lg(str):
           if len(str) < 3:
             return str
           elif "ing" in str[-3:]:
             return str + "ly"
           else:
             return str + "ing"

         print( append_ing_lg('abc'))
         print( append_ing_lg('string'))
```

```
abcing
stringly
```

Q4. Write the Python program to find the first appearance of the substring 'not' and 'poor' from the given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'.Return the resulting string.

Sample string: 'The lyrics are not that poor!''The lyrics arepoor!'

Expected Result: 'The lyrics are good!''The lyrics are poor!'

```
In [0]: def fun(str):
            try:
                a, b = str.index('not'), str.index('poor')
                if a < b:
                   return str[0:a] + "good" + str[b+4:]
                else:
                  return str
            except:
                return str

        fun("'The lyrics are not that poor!''The lyrics are poor!'")
```

```
Out[59]: "'The lyrics are good!''The lyrics are poor!'"
```

Q5.Write the Python program to remove the characters which have odd index values of a given string.

```
In [0]: def remove_oddindex(str):
            return "".join([str[i] for i in range(len(str)) if i%2 != 0])

        remove_oddindex("MachineLearning")
```

```
Out[73]: 'ahnLann'
```

Q6. Write the Python function to get a string made of 4 copies of the last two characters of the specified string (length must be at least 2).

Sample functionandresult :

insert-end('Python') -> abababab

insert-end('Exercises') -> jkjkjkjk

```
In [0]: def insert_end(str):
            if len(str) >=2:
                return str[-2:]*4
            return str

        insert_end('vijayakumar')
```

```
Out[78]: 'arararar'
```

Q7. Write the python function to get a string made of its first three characters of a specified string. If the length of the string is less than 3 then return the original string.

Sample function and result :

first-three('ipy') -> ipy

first-three('python') -> pyt

```
In [0]: def first_three(str):
            if len(str) > 2:
                return str[:3]
            return str

        print(first_three('ipy'))
        print(first_three('python'))
```

```
ipy
pyt
```

Q8. Write the python program to print the following floating numbers upto 2 decimal places?

```
In [0]: def two_decimal(number):
          return float('{0:.2f}'.format(number))

        two_decimal(2.337)
```

```
Out[97]: 2.34
```

Q9. Write the Python program to format a number with a percentage?

```
In [0]: def format_number(number):
            return "{0}%".format(number)

        format_number(100)

Out[109]: '100%'
```

Q10. Write thePython program to count occurrences of a substring in a String?

```
In [0]: def count_occurrences(str, substr):
            try:
                substr_occur_count = 0
                substr_found = True
                while substr_found:
                    substr_found = False
                    idx = str.index(substr)
                    if idx > -1:
                        str = str[idx + len(substr):]
                        substr_occur_count += 1
                        substr_found = True
                return substr_occur_count
            except:
                return substr_occur_count

        count_occurrences("where there's a will there's a way", "there's")

Out[132]: 2
```

Q11. Write the Python program to count repeated characters in a string.

In [0]:
```
import collections

str = 'thequickbrownjumpsoverthelazydog'

_dict_repeated_chars = {}
for i in range(len(str)):
    char = str[i]
    if char in _dict_repeated_chars.keys():
        _dict_repeated_chars[char] = _dict_repeated_chars[char] + 1
    else:
        _dict_repeated_chars[char] = 1
else:
    _dict_repeated_chars = sorted(_dict_repeated_chars.items(), key=lambda kv: kv
    _dict_repeated_chars = collections.OrderedDict(_dict_repeated_chars)
    for key in _dict_repeated_chars:
        if _dict_repeated_chars[key] > 1:
            print( key, _dict_repeated_chars[key])
```

```
t 2
h 2
u 2
r 2
e 3
o 3
```

Q12. Write the Python program to print the square and cube symbol in the area of a rectangle and volume of a cylinder?

Sample output-

The area of the rectangle is 1256.66 cm2

The volume of the cylinder is 1254.725 cm3

In [0]:
```
area_of_rectangle = 1256.66
area_of_cylinder = 1254.725
print('The are of rectangle is {0:.2f}cm\u00b2'.format(area_of_rectangle))
print('The volue of the cylinder is {0:.3f}cm\u00b3'.format(area_of_cylinder))
```

```
The are of rectangle is 1256.66cm²
The volue of the cylinder is 1254.725cm³
```

Q13. Write thePython program to check if a string contains all letters of the alphabet?

In [86]:
```
def is_contains_allalphabets(str):
    contains_all_apphabets = True
    for char in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ':
        if char not in str.upper():
            contains_all_apphabets = False
            break
    return contains_all_apphabets

print(is_contains_allalphabets('abcdefghijklmnopqrstuvwxyz'))
print(is_contains_allalphabets('abcdefghijklmnopqrstuvwxz'))
```

```
True
False
```

Q14. Write the Python program to find the second most repeated word in a given string?

In [0]:
```
import collections

_list_of_words = 'where there is a will thee is a way'.split()
_dict_repeated_words = {}
for i in range(len(_list_of_words)):
    word = _list_of_words[i]
    if word in _dict_repeated_words.keys():
        _dict_repeated_words[word] = _dict_repeated_words[word] + 1
    else:
        _dict_repeated_words[word] = 1
else:
    _dict_repeated_words = sorted(_dict_repeated_words.items(), key=lambda kv: kv
    print(_dict_repeated_words[-2])
```

```
('is', 2)
```

Q15. Write the Python program to find the minimum window in the given string,which will contains all the characters of another given strings?

Example 1

Input : string1 = " PRWSOERIUSFK "

string2 = " OSU "

Output: Minimum window is "OERIUS"

In [88]:
```
import collections
def min_window(str1, str2):
    result_char, missing_char = collections.Counter(str2), len(str2)
    i = p = q = 0
    for j, c in enumerate(str1, 1):
        missing_char -= result_char[c] > 0
        result_char[c] -= 1
        if not missing_char:
            while i < q and result_char[str1[i]] < 0:
                result_char[str1[i]] += 1
                i += 1
            if not q or j - i <= q - p:
                p, q = i, j
    return str1[p:q]

str1 = "PRWSOERIUSFK"
str2 = "OSU"
print("Minimum window is", min_window(str1,str2))
```

```
Minimum window is OERIUS
```

### Q16. Write the Python program to find smallest window that contains all characters of the given string?

Original Strings:

asdaewsqgtwwsa

Smallest window that contains all characters of the said string:

Daewsqgt

In [90]:
```
from collections import defaultdict

def find_sub_string(str):
    str_len = len(str)

    # Count all distinct characters.
    dist_count_char = len(set([x for x in str]))

    ctr, start_pos, start_pos_index, min_len = 0, 0, -1, 9999999999
    curr_count = defaultdict(lambda: 0)
    for i in range(str_len):
        curr_count[str[i]] += 1

        if curr_count[str[i]] == 1:
            ctr += 1

        if ctr == dist_count_char:
            while curr_count[str[start_pos]] > 1:
                if curr_count[str[start_pos]] > 1:
                    curr_count[str[start_pos]] -= 1
                start_pos += 1

            len_window = i - start_pos + 1
            if min_len > len_window:
                min_len = len_window
                start_pos_index = start_pos
    return str[start_pos_index: start_pos_index + min_len]

str1 = "asdaewsqgtwwsa"
print("Original Strings:\n",str1)
print("\nSmallest window that contains all characters of the said string:")
print(find_sub_string(str1))
```

```
Original Strings:
 asdaewsqgtwwsa

Smallest window that contains all characters of the said string:
daewsqgt
```

### Q17. Write thePython program to count number of substrings from a given string of lowercase alphabets with exactly k distinct (given) characters?

```
Input a string (lowercase alphabets): wolf

Input k:4

Number of substrings with exactly 4 distinct characters:1
```

### Q18. Write thePython program to count number of non-empty substrings of the given string?

```
Input a string:  w3resource

Number of substrings:55
```

In [92]:
```
def number_of_substrings(str):
    str_len = len(str)
    return int(str_len * (str_len + 1) / 2)

str1 = input("Input a string: ")
print("Number of substrings:")
print(number_of_substrings(str1))
```

```
Input a string: w3resource
Number of substrings:
55
```

Q19.Write thePython program to count number of substrings with same first and last characters of thegiven string?

In [93]:
```
def no_of_substring_with_equalEnds(str1):
    result = 0;
    n = len(str1);
    for i in range(n):
        for j in range(i, n):
            if (str1[i] == str1[j]):
                result = result + 1
    return result
str1 = input("Input a string: ")
print(no_of_substring_with_equalEnds(str1))
```

```
Input a string: vijay
5
```

Q20.Write thePython program to count the number of strings where the string length is 2 or more, and first and last character are same from a given list of strings.

In [0]:
```
def findwordswith_fandlcharsame(list_of_words):
    _words = [word for word in _list_of_words if word[0] == word[-1]]
    return len(_words)

print(findwordswith_fandlcharsame(['abc', 'xyz', 'wxw', '1331']))
```

```
3
```

Q21.Write the Python program to get a list, sorted in increasing orderby the last element in each tuple from the given list of non-empty tuples?

Sample List -[(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]

Expected Result -[(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

In [0]:
```
from collections import defaultdict
_dict = defaultdict(list)
_list_of_tuples = [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]

for tpl in _list_of_tuples:
    _dict[tpl[1]] = tpl
else:
    _sorted_dict = sorted(_dict.items(), key=lambda kv: kv[0])
    _sorted_dict = collections.OrderedDict(_sorted_dict)
    print([value for value in _sorted_dict.values()])
```

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

Q22. Write thePython program to remove duplicates from a list?

In [0]:
```
_list_of_items = [10,20,30,20,10,50,60,40,80,50,40]
_distinct_list = []
for item in _list_of_items:
    if item not in _distinct_list:
        _distinct_list.append(item)
print(_distinct_list)
```

```
[10, 20, 30, 50, 60, 40, 80]
```

Q23. Write the Python program to find the list of words that are longer than n from a given list of words?

In [0]:
```
def filter_long_words(seq, n):
    return [word for word in seq if len(word) > n ]

words = ['Artificial Intelligence','Deep Learning', 'Neural Networks', 'Data Mini
filter_long_words(words, 14)
```

Out[218]:
```
['Artificial Intelligence', 'Neural Networks']
```

Q24. Write the Python program to print a specified list after removing the0th, 4th,and 5th elements?

Sample List -['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

Expected Output -['Green', 'White', 'Black']

In [0]:
```
_list_of_words = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
_filtered_words = [_list_of_words[index] for index in range(len(_list_of_words))
print(_filtered_words)
```

```
['Green', 'White', 'Black']
```

Q25. Write the Python program to generate all permutations of a list in

Python?

In [0]:
```
from itertools import permutations
_list = list(permutations([1,2,3]))
print(_list)
```

```
[(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)]
```

Q26. Write the Python program to convert a pair of values into a sorted unique array?

Type *Markdown* and LaTeX: α^2

Type *Markdown* and LaTeX: α^2

In [0]:
```
_list_of_tuples = [(1, 2), (3, 4), (1, 2), (5, 6), (7, 8), (1, 2), (3, 4), (3, 4)
_unique_items = set()
for v1,v2 in _list_of_tuples:
    _unique_items.add(v1)
    _unique_items.add(v2)
print(list(_unique_items))
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Q27. Write the Python class to convert an integer to a roman numeral?

In [0]:
```
class RomanNumeral:
    def __init__(self):
        pass
    def convert(self, number):
        values = [1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1]
        roman_symbols = ['M', 'CM', 'D', 'CD', 'C', 'XC', 'L', 'XL', 'X', 'IX', '

        i = 0
        roman_number = ''
        while (number > 0):
            for _ in range(number//values[i]):
                roman_number += roman_symbols[i]
                number = number - values[i]
            i += 1
        return roman_number

roman = RomanNumeral()
roman.convert(3523)
```

Out[16]:
```
'MMMDXXIII'
```

Q28 Write thePython class to convert a Roman numeral to an integer?

```
In [0]: class IntegerConversion:
            def __init__(self):
                pass

            def convert(self, roman_numeral):
                values = [1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1]
                roman_symbols = ['M', 'CM', 'D', 'CD', 'C', 'XC', 'L', 'XL', 'X', 'IX', '

                _integer = 0
                while (len (roman_numeral) > 0):
                    i = 2
                    while(i>0):
                        symbol = roman_numeral[:i]
                        is_exist = symbol in roman_symbols
                        i = i-1
                        if is_exist:
                            index = roman_symbols.index(symbol)
                            _integer += values[index]
                            roman_numeral = roman_numeral[i+1:]
                            break
                return _integer

        integer = IntegerConversion()
        integer.convert('MCM')

Out[47]: 1900
```

Q29. Write the Python class to find the validity of the string of the parentheses, '(', ')', '{', '}', '[' and '] and the brackets must be closed in the correct order, example -"()" and "()[]{}" are valid but "[)","({[)]" and "{{{"are invalid

In [0]:
```
class StringExpValidator:
    def __init__(self):
        self.items = []

    def validate(self, exp):

        _valied_exp = 'valied'
        _bracket_symbols = ['(', ')', '{', '}','[', ']']

        for char in exp:
            if char in _bracket_symbols:
              if (_bracket_symbols.index(char) % 2 == 1) and (len(self.items) == (
                  _valied_exp = 'invalied'
                  break

              elif (_bracket_symbols.index(char) % 2 == 1) and (len(self.items) !
                  if  _bracket_symbols[ _bracket_symbols.index(char) - 1 ] not in
                      _valied_exp = 'invalied'
                      break
                  else:
                      self.items.remove(_bracket_symbols[ _bracket_symbols.index(
              else:
                  self.items.append(char)
        if len(self.items) != 0:
            _valied_exp = 'invalied'
        return _valied_exp


expvalidator = StringExpValidator()
print( '[4 + (3 * 2)] = ', expvalidator.validate('[4 + (3 * 2)]') )
print( '[4 + (3 * 2) = ', expvalidator.validate('[4 + (3 * 2)') )
```

```
[4 + (3 * 2)] =  valied
[4 + (3 * 2) =  invalied
```

Q30. Write thePython class to get all possible unique subsets from a set of distinct integers?

Input -[4, 5, 6]

Output -[[], [6], [5], [5, 6], [4], [4, 6], [4, 5], [4, 5, 6]]

In [0]:
```
class py_solution:
    def sub_sets(self, sset):
        return self.subsetsRecur([], sorted(sset))

    def subsetsRecur(self, current, sset):
        if sset:
            return self.subsetsRecur(current, sset[1:]) + self.subsetsRecur(curre
        return [current]

print(py_solution().sub_sets([4,5]))
```

```
[[], [5], [4], [4, 5]]
```

In [0]:
```
class Set:
    def __init__(self):
        pass

    def get_subsetof(self, sset):
        _list = [ i+1 for i in range(len(sset))]
        print(_list)
        _list_of_set = []
        for i in _list:
            _set = []
            for j in range(i):
                for k in sset:
                   _set.append(k)
            _list_of_set.append(_set)
        print(_list_of_set)

obj_set = Set()
obj_set.get_subsetof([4,5])
```

```
[1, 2]
[[4, 5], [4, 5, 4, 5]]
```

Q31. Write the Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals the specific target number?

Input: numbers-[10,20,10,40,50,60,70], target=50

Output-3, 4

In [0]:
```
class AddTwoIndexValues:

    def __init__(self):
        pass

    def to_find_target(self, list_of_nums, target):

        for index, value in enumerate(list_of_nums):
            _sublist =  [ x + value for x in _list_of_nums[:index]]
            if target in _sublist:
                return _sublist.index(target)+1, index+1
        else:
          print('no numbers with indexes to find target')

addtwo_indexvalues = AddTwoIndexValues()
addtwo_indexvalues.to_find_target([20,20,10,40,50,60,70], 30)
```

Out[214]:
```
(1, 3)
```

Q32. Write the Python class to find the three elements that sum to zero from the set of n real numbers?

Input array-[-25, -10, -7, -3, 2, 4, 8, 10]

Output -[[-10, 2, 8], [-7, -3, 10]]

In [0]:
```
class ThreeElementSum:
    def __init__(self):
        pass

    def to(self, list_of_nums, target):
        is_exists = False
        _list_of_triples = []

        for i in range(len(list_of_nums) -2):
            for j in range(1, len(list_of_nums)-1):
                for k in range(2, len(list_of_nums)):
                    if (list_of_nums[i] + list_of_nums[j] + list_of_nums[k]) == t
                        for subset in _list_of_triples:
                            if( (list_of_nums[i]  in subset) and (list_of_nums[j]
                                is_exists = True
                                break
                        if is_exists == False:
                            _list_of_triples.append( [list_of_nums[i], list_of_nu
                    is_exists = False
        return _list_of_triples


threelementsum = ThreeElementSum()
threelementsum.to([-25, -10, -7, -3, 2, 4, 8, 10], 0)
```

Out[278]:
```
[[-10, 2, 8], [-7, -3, 10]]
```

Q33. Write the Python class to implement pow(x, n)?

In [0]:
```
class Power:
    def __init__(self):
        pass

    def of(self, x, n):
        return x**n


power = Power()
power.of(3,3)
```

Out[280]:
```
27
```

Q34. Write the Python class which has two methods get_String and print_String. get_String accept thestring from the user and print_String print the string in upper case.

In [0]:
```
class String:
    def __init__(self):
        pass

    def get_String(self):
        self.string = input()

    def print_String(self):
        print(self.string)

mystring = String()
mystring.get_String()
mystring.print_String()
```

```
vijay
vijay
```

Q35. Write the Python class named Rectangle constructed by a length and width and themethod which will compute the area of the rectangle?

In [0]:
```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def area(self):
        return self.length * self.width

rectangle = Rectangle(40,50)
print( 'Area of Rectangle is', rectangle.area() )
```

```
Area of Rectangle is 2000
```

Q36. Write the Python class named Circle constructed by the radius and two methods which will compute the area and perimeter of the circle?

In [0]:
```
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return 3.142 * (self.radius * self.radius)

    def perimeter(self):
        return 2 * 3.142 * self.radius

circle = Circle(11.7)
print( 'Area =', circle.area())
print( 'Perimeter =', circle.perimeter())
```

```
Area = 430.10837999999995
Perimeter = 73.52279999999999
```

Q37. Write the Python program to get the class name of an instance in Python?

In [0]:
```
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return 3.142 * (self.radius * self.radius)

    def perimeter(self):
        return 2 * 3.142 * self.radius

circle = Circle(11.7)
print(type(circle).__name__)
```

```
Circle
```

Q38. Write thePython program to count the number of students of individual class?

In [0]:
```
from collections import Counter
classes = (
    ('V', 1),
    ('VI', 1),
    ('V', 2),
    ('VI', 2),
    ('VI', 3),
    ('VII', 1),
)
students = Counter(class_name for class_name, no_students in classes)
print(students)
```

```
Counter({'VI': 3, 'V': 2, 'VII': 1})
```

Q39. Write thePython program to create an instance of an OrderedDict using thegiven dictionary andsort dictionary during the creation and printmembers of the dictionary in reverse order?

In [0]:
```
from collections import OrderedDict
dict = OrderedDict({'Afghanistan': 93, 'Albania': 355, 'Algeria': 213, 'Andorra':

for key, value in dict.items():
    print(key, value)

print('\nIn reverse order\n')
for key in reversed(dict):
    print(key, dict[key])
```

```
Afghanistan 93
Albania 355
Algeria 213
Andorra 376
Angola 244

In reverse order

Angola 244
Andorra 376
Algeria 213
Albania 355
Afghanistan 93
```

Q40. Write thePython program to compare two unordered lists (not sets)?

In [0]:
```
from collections import Counter

def comparelist(list1, list2):
    return Counter(list1) == Counter(list2)

print(comparelist([1,2,3], [1,3,4]))
```

```
False
```

Q41. Write thePython program to get an array buffer information?

In [0]:
```
from array import array
arr = array("I", (1,2))
print('Array buffer start address in memory and number of elements.\n',arr.buffer
```

```
Array buffer start address in memory and number of elements.
 (140312645457528, 2)
```

Q42. Write thePython program to convert an array to an array of machine values and return thebytes representation?

In [0]:
```
from array import array
import binascii

A1 = array("i",[1,2,3,4,5,6])
print('Original array-\nA1:', A1)
print('\nArray of bytes:\n', binascii.hexlify(A1.tobytes())  )
```

```
Original array-
A1: array('i', [1, 2, 3, 4, 5, 6])

Array of bytes:
 b'010000000200000003000000040000000500000006000000'
```

Q43. Write thePython program to read a string and interpreting the string as an array of machine values?

In [0]:
```
from array import array
import binascii
array1 = array('i', [7, 8, 9, 10])
print('array1:', array1)
as_bytes = array1.tobytes()
print('Bytes:', binascii.hexlify(as_bytes))
array2 = array('i')
array2.frombytes(as_bytes)
print('array2:', array2)
```

```
array1: array('i', [7, 8, 9, 10])
Bytes: b'0700000008000000090000000a000000'
array2: array('i', [7, 8, 9, 10])
```

Q44. Write the Python program to push three items into the heap and return the smallest item from the heap. Also, return and popthe smallest item from the heap?

In [0]:
```
import heapq

heap = []
heapq.heappush(heap, ('VI', 1))
heapq.heappush(heap, ('VII', 2))
heapq.heappush(heap, ('VIII' ,3))
print('Items in the heap-')
for item in heap:
    print(item)
print('-----------------')

print('The smallest item in the heap:')
print(heapq.nsmallest(1, heap))

print('-----------------\n')
print('Pop the smallest item in the heap:')
heapq.heappop(heap)

for item in heap:
    print(item)
```

```
Items in the heap-
('VI', 1)
('VII', 2)
('VIII', 3)
-----------------
The smallest item in the heap:
[('VI', 1)]
-----------------

Pop the smallest item in the heap:
('VII', 2)
('VIII', 3)
```

Q45. Write the Python program to locate the left insertion point for a specified value in sorted order?

In [0]:
```
import bisect
def index(a, x):
    i = bisect.bisect_left(a, x)
    return i

a = [1,2,4,5]
print(index(a, 6))
print(index(a, 3))
```

```
4
2
```

Q46. Write thePython program to create theFIFO queue?

In [0]:
```
import queue

q = queue.Queue()
q.put(0)
q.put(1)
q.put(2)
q.put(3)

while not q.empty():
    print(q.get(), end=" ")
```

```
0 1 2 3
```

Q47. Write the Python program to calculate the harmonic sum of n-1.Note: The harmonic sum is the sum of reciprocals of the positive Integers?

In [0]:
```
n = 10
sum = 1
for i in range(2, n+1):
    sum += 1/i
print(sum)
```

```
2.9289682539682538
```

Q48. Write the NumPy program to create a 2d array with 6 on the border and 0 inside?

In [0]:
```
import numpy as np
arr = np.zeros((5,5), dtype = int)
arr[:] = 6
print('Original array-')
print(arr)
arr[1:-1, 1:-1] = 0
print('\n\n6 on the border and 0 inside in the array-')
print(arr)
```

```
Original array-
[[6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]
 [6 6 6 6 6]]


6 on the border and 0 inside in the array-
[[6 6 6 6 6]
 [6 0 0 0 6]
 [6 0 0 0 6]
 [6 0 0 0 6]
 [6 6 6 6 6]]
```

Q49. Write the NumPy program to create a 8x8 matrix and fillit with the checkerboard pattern?

In [0]:
```
import numpy as np
arr = np.zeros((8,8), dtype=int)
arr[:] = [3,9,3,9,3,9,3,9]
arr
```

Out[94]:
```
array([[3, 9, 3, 9, 3, 9, 3, 9],
       [3, 9, 3, 9, 3, 9, 3, 9],
       [3, 9, 3, 9, 3, 9, 3, 9],
       [3, 9, 3, 9, 3, 9, 3, 9],
       [3, 9, 3, 9, 3, 9, 3, 9],
       [3, 9, 3, 9, 3, 9, 3, 9],
       [3, 9, 3, 9, 3, 9, 3, 9],
       [3, 9, 3, 9, 3, 9, 3, 9]])
```

Q50. Write theNumPy program to create an empty and a full array

In [0]:
```
import numpy as np
arr_embty = np.empty((3,4))
print(arr_embty)

arr_full = np.full((3,3), 6)
print(arr_full)
```

```
[[2.70584339e-316 8.48798317e-313 2.05833592e-312 2.18565567e-312]
 [9.33678149e-313 8.91238232e-313 2.27053550e-312 2.05833592e-312]
 [2.18565567e-312 8.70018275e-313 1.35570810e+295 1.46030983e-319]]
[[6 6 6]
 [6 6 6]
 [6 6 6]]
```

Q51. Write the NumPy program to convert the values of Centigradedegrees into the Fahrenheit degrees and the centigrade values are storedintheNumPy array

In [0]:
```
Fahrenheit_degress = [0, 12, 45.21 ,34, 99.91]
centigrade_degrees = list(map(lambda x: (x - 32) * 5/9,  Fahrenheit_degress))
print('Values in Fahrenheit degrees-')
print(Fahrenheit_degress)
print('Values in Centigrade degrees-')
print(centigrade_degrees)
```

```
Values in Fahrenheit degrees-
[0, 12, 45.21, 34, 99.91]
Values in Centigrade degrees-
[-17.77777777777778, -11.11111111111111, 7.33888888888889, 1.1111111111111112,
37.727777777777774]
```

Q52. Write the NumPy program to find the real and imaginary parts of an array of complex numbers?

In [0]:
```
import numpy as np
x =  [1.00000000+0.j, 0.70710678+0.70710678j]
real = np.array([])
imag = np.array([])
for complex_no in x:
    real = np.append(real, complex_no.real)
    imag = np.append(imag, complex_no.imag)
print('Original array ', x)
print('Real part of the array-')
print(real)
print('maginary part of the array-')
print(imag)
```

```
Original array  [(1+0j), (0.70710678+0.70710678j)]
Real part of the array-
[1.         0.70710678]
maginary part of the array-
[0.         0.70710678]
```

Q53. Write theNumPy program to test whether each element of a 1-D array is also present in thesecond array?

In [0]:
```
import numpy as np
arr1 = np.array([0, 10, 20, 40, 60])
arr2 = np.array([0, 40])
print('Array1: ', arr1)
print('Array2: ', arr2)
print('Compare each element of array1 and array2')
print(np.in1d(arr1, arr2))
```

```
Array1:  [ 0 10 20 40 60]
Array2:  [ 0 40]
Compare each element of array1 and array2
[ True False False  True False]
```

Q54. Write theNumPy program to find common values between two arrays?

In [0]:
```
import numpy as np
arr1 = np.array([0, 10, 20, 40, 60])
arr2 = np.array([10, 30, 40])
print('Array1:', arr1)
print('Array2:', arr2)
print('Common values between two arrays-')
print(np.intersect1d(arr1, arr2))
```

```
Array1: [ 0 10 20 40 60]
Array2: [10 30 40]
Common values between two arrays-
[10 40]
```

Q55. Write theNumPy program to get the unique elements of an array?

In [0]:
```
import numpy as np

arr = np.array([10, 10, 20, 20, 30, 30])
print('Original array-')
print(arr)
print('Unique elements of the above array-')
print(np.unique(arr))

arr = np.array( [[1,1], [2,3]]  )
print('Original array-')
print(arr)
print('Unique elements of the above array-')
print(np.unique(arr))
```

```
Original array-
[10 10 20 20 30 30]
Unique elements of the above array-
[10 20 30]
Original array-
[[1 1]
 [2 3]]
Unique elements of the above array-
[1 2 3]
```

Q56. Write the NumPy program to find the set exclusive-or of two arrays. Set exclusive-or will return the sorted, unique values that are in only one (not both) of the input arrays?

In [0]:
```
import numpy as np
arr1 = np.array([0, 10, 20, 40, 60, 80])
arr2 = np.array([10, 30, 40, 50, 79])
print('Array1-', arr1)
print('Array2-', arr2)
print('Unique values that are in only one (not both) of the input arrays-')
print(np.setxor1d(arr1, arr2))
```

```
Array1- [ 0 10 20 40 60 80]
Array2- [10 30 40 50 79]
Unique values that are in only one (not both) of the input arrays-
[ 0 20 30 50 60 79 80]
```

Q57. Write theNumPy program to test if all elements in an array evaluate to True?

In [0]:
```
import numpy as np
print(np.all([12, 34, 3, 1]))
print(np.all([12, 0, 3, 1]))
print(np.all([[12, 34], [3, 1]]))
print(np.all([[12, 34], [0, 1]]))
```

```
True
False
True
False
```

Q58 Write theNumPy program to test whether any array element along thegiven axis evaluates to True?

In [0]:
```
import numpy as np
print(np.all([[12, 34], [0, 1]],axis=1))
print(np.all([[12, 34], [0, 1]],axis=0))
```

```
[ True False]
[False  True]
```

Q59. Write the NumPy program to construct an array by repeating?

In [0]:
```
arr = np.array([1, 2, 3, 4])
print('Original array')
print(arr)
print('Repeating 2 times')
print(np.tile(arr, 2))
print('Repeating 3 times')
print(np.tile(arr, 3))
```

```
Original array
[1 2 3 4]
Repeating 2 times
[1 2 3 4 1 2 3 4]
Repeating 3 times
[1 2 3 4 1 2 3 4 1 2 3 4]
```

Q60. Write the NumPy program to find the indices of the maximum and minimum values withthe given axis of an array?

In [0]:
```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6])
print('Original array-', arr)
print('Maximum Values-', np.argmax(arr))
print('Minimum Values-', np.argmin(arr))
```

```
Original array- [1 2 3 4 5 6]
Maximum Values- 5
Minimum Values- 0
```

Q61. Write theNumPy program compare two arrays using numpy?

In [0]:
```
import numpy as np
a = np.array([1,2])
b = np.array([4,5])
print('Array a-', a)
print('Array b-', b)
print('a>b')
print(np.greater(a, b))
print('a>=b')
print(np.greater_equal(a,b))
print('a<b')
print(np.less(a,b))
print('a<=b')
print(np.less_equal(a,b))
```

```
Array a- [1 2]
Array b- [4 5]
a>b
[False False]
a>=b
[False False]
a<b
[ True  True]
a<=b
[ True  True]
```

Q62. Write the NumPy program to sort an along the first, last axis of an array?

In [0]:
```
import numpy as np
a = np.array([[4,6],[2,1]])
print('Original array: ')
print(a)
print('Sort along the first axis:')
x = np.sort(a, axis=0)
print(x)
print('Sort along the last axis:')
y = np.sort(x, axis=1)
print(y)
```

```
Original array:
[[4 6]
 [2 1]]
Sort along the first axis:
[[2 1]
 [4 6]]
Sort along the last axis:
[[1 2]
 [4 6]]
```

Q63. Write the NumPy program to sort pairs of first name and last name

return their indices(first by last name, then by first name).

In [0]:
```
import numpy as np
first_names =    ('Margery', 'Betsey', 'Shelley', 'Lanell', 'Genesis')
last_names = ('Woolum', 'Battle', 'Plotner', 'Brien', 'Stahl')
x = np.lexsort((first_names, last_names))
print(x)
```

```
('Margery', 'Betsey', 'Shelley', 'Lanell', 'Genesis')
[1 3 2 4 0]
```

Q64. Write the NumPy program to get the values and indices of the elements that are bigger than 10 int he given array?

In [0]:
```
import numpy as np
x = np.array([[0,10,20],[20,30,40]])
print('Original array-')
print(x)
print('Values bigger than 10=', x[x>10])
print('Their indices are ', np.nonzero(x>10))
```

```
Original array-
[[ 0 10 20]
 [20 30 40]]
Values bigger than 10= [20 20 30 40]
Their indices are  (array([0, 1, 1, 1]), array([2, 0, 1, 2]))
```

Q65. Write the NumPy program to find the memory size of a NumPy array?

In [0]:
```
import numpy as np
x = np.ones((4,4))
print("%d bytes" % (x.size * x.itemsize) )
```

```
128 bytes
```

Q66. Write the NumPy program to create an array of ones and an array of zeros?

In [0]:
```
import numpy as np
print('Create an array of zeros')
x = np.zeros((1,2))
print('Default type is float')
print(x)
print('Type changes to int')
x = x.astype(np.int)
print(x)

print('Create an array of ones')
y = np.ones((1,2))
print('Default type is float')
print(y)
print('Type changes to int')
y = y.astype(np.int)
print(y)
```

```
Create an array of zeros
Default type is float
[[0. 0.]]
Type changes to int
[[0 0]]
Create an array of ones
Default type is float
[[1. 1.]]
Type changes to int
[[1 1]]
```

Q67. Write the NumPy program to change the dimension of an array?

In [0]:
```
import numpy as np
arr1 = np.array([1,2,3,4,5,6])
print('6 rows and 0 columns')
print(arr.shape)

arr2 = np.array([[1,2,3],[4,5,6],[7,8,9]])
print('(3, 3) -> 3 rows and 3 columns')
print(arr2)

arr3 = np.array([1,2,3,4,5,6,7,8,9])
print('Change array shape to (3, 3) -> 3 rows and 3 columns')
arr3 = arr3.reshape(3,3)
print(arr3)
```

```
6 rows and 0 columns
(6,)
(3, 3) -> 3 rows and 3 columns
[[1 2 3]
 [4 5 6]
 [7 8 9]]
Change array shape to (3, 3) -> 3 rows and 3 columns
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Q68. Write the NumPy program to create a new shape to an array without changing its data ?

In [0]:
```
import numpy as np
x = np.array([1,2,3,4,5,6])
x = x.reshape((3,2))
print('Reshape 3x2-')
print(x)
x = x.reshape((2,3))
print('Reshape 2x3-')
print(x)
```

```
Reshape 3x2-
[[1 2]
 [3 4]
 [5 6]]
Reshape 2x3-
[[1 2 3]
 [4 5 6]]
```

Q69. Write the NumPy program to create a new array of 3*5, filled with 2?

In [0]:
```
import numpy as np
x = np.full((3,5), 2, dtype=np.int)
print(x)

y = np.ones([3,5], dtype= np.int) * 2
print(y)
```

```
[[2 2 2 2 2]
 [2 2 2 2 2]
 [2 2 2 2 2]]
[[2 2 2 2 2]
 [2 2 2 2 2]
 [2 2 2 2 2]]
```

Q70. Write the NumPy program to create a 3-D array with ones on a diagonal and zeros elsewhere?

In [0]:
```
import numpy as np
x = np.eye(3)
print(x)
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

Q71. Write the NumPy program to split an array of 14 elements into the 3 arrays and each of which has 2, 4,and 8 elements in original order?

In [0]:
```
import numpy as np
x = np.arange(1,15)
print('Oroginal array-', x)
print('After splitting-')
print( np.split(x, [2,6]))
```

```
Oroginal array- [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14]
After splitting-
[array([1, 2]), array([3, 4, 5, 6]), array([ 7,  8,  9, 10, 11, 12, 13, 14])]
```

Q72. Write the NumPy program to split of an array of shape 4x4 it into two arrays along the second axis ?

In [0]:
```
import numpy as np
x = np.arange(16).reshape(4,4)

print( np.hsplit(x, [2,6]))
```

```
[array([[ 0,  1],
       [ 4,  5],
       [ 8,  9],
       [12, 13]]), array([[ 2,  3],
       [ 6,  7],
       [10, 11],
       [14, 15]]), array([], shape=(4, 0), dtype=int64)]
```

Q73. Write the NumPy program to create a 5x5 matrix with row values ranging from 0 to 4?

In [0]:
```
import numpy as np
x = np.zeros((5,5))
print('Original array-')
print(x)

print('Row values ranging from 0 to 4.')
x = x+ np.arange(5)
print(x)
```

```
Original array-
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
Row values ranging from 0 to 4.
[[0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]]
```

Q74. Write the NumPy program to create an array of zeros and three column types (integer, float, character)?

In [0]:
```
import numpy as np
x = np.zeros(3, dtype = ('i4, f4, a40'))
x[:] = [(1, 2., "Albert Einstein"), (2, 2., "Edmond Halley"), (3, 3., "Gertrude B
print(x)
```

```
[(1, 2., b'Albert Einstein') (2, 2., b'Edmond Halley')
 (3, 3., b'Gertrude B. Elion')]
```

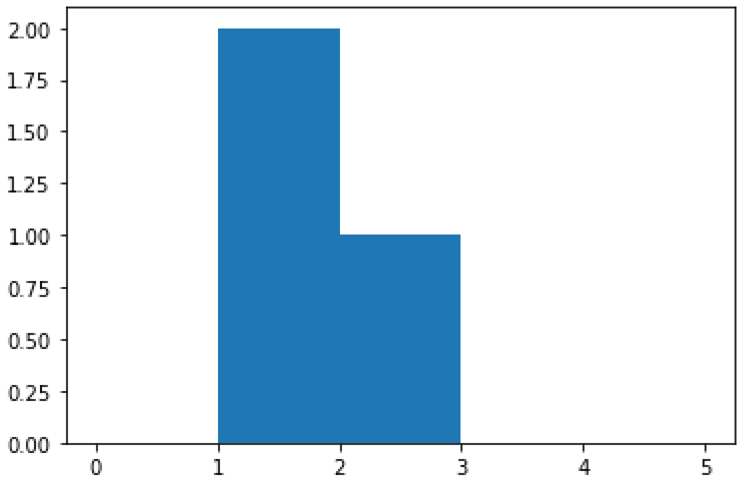
Q75. Write the NumPy program toremove the negative values in thenumpy array with 0?

In [0]:
```
import numpy as np
x = np.array([-1, -4, 0, 2, 3, 4, 5, -6])
print('Original array-')
print(x)
print("Replace the negative values of the said array with 0:")
x[ x < 0] = 0
print(x)
```

```
Original array-
[-1 -4  0  2  3  4  5 -6]
Replace the negative values of the said array with 0:
[0 0 0 2 3 4 5 0]
```

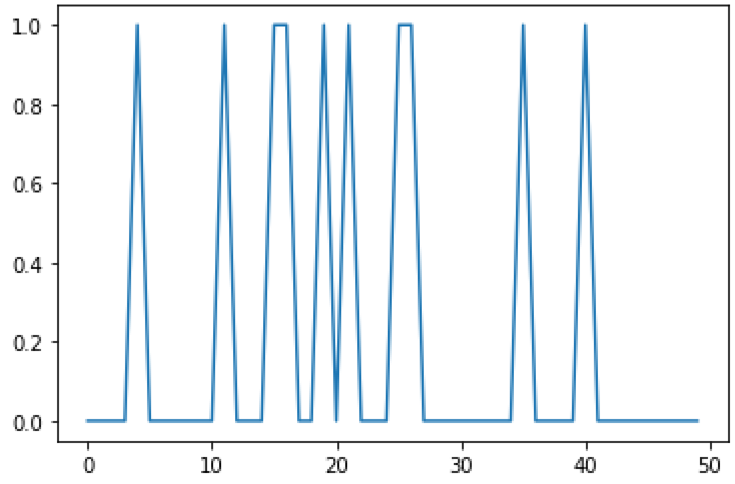
Q76. Write the NumPy program to compute the histogram of a set of data?

In [0]:
```
import numpy as np
import matplotlib.pyplot as plt
x = [1, 2.3, 1.5]
plt.hist(x, bins = [0, 1, 2, 3, 5])
plt.show()
```



Q77.Write the NumPy program to compute the line graph of a set of data?

In [0]:
```
import numpy as np
import matplotlib.pyplot as plt
arr = np.random.randint(1, 50, 10)
y, x = np.histogram(arr, bins=np.arange(51))
fig, ax = plt.subplots()
ax.plot(x[:-1], y)
fig.show()
```



Q78. Write theNumPy program to extractsall the elements fromsecond row from given (4x4) array?

In [0]:
```
import numpy as np
arr = np.arange(16).reshape(4,4)
print('Original array-')
print(arr)
print('Extracted data- Second row')
print( arr[1])
```

```
Original array-
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]]
Extracted data- Second row
[4 5 6 7]
```

Q79. Write the NumPy program to extract first element of the second row and fourth element of fourth row from a given (4x4) array

In [0]:
```
import numpy as np
arr = np.arange(16).reshape(4,4)
print('Original array-')
print(arr)
print('Extracted data-First element of the second row and fourth element of fourt
print( arr[[1,3], [0,3]])
```

```
Original array-
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]]
Extracted data-First element of the second row and fourth element of fourth row
w
[ 4 15]
```

Q80. Write theNumPy program to add two arrays Aand B of sizes (3,3) and (,3)?

In [0]:
```
import numpy as np
A = np.ones((3,3))
B = np.arange(3)
print('Original array-')
print('Array-1')
print(A)
print('Array-2')
print(B)
print('A+B\n', A+B)
```

```
Original array-
Array-1
[[1. 1. 1.]
 [1. 1. 1.]
 [1. 1. 1.]]
Array-2
[0 1 2]
A+B
 [[1. 2. 3.]
 [1. 2. 3.]
 [1. 2. 3.]]
```

Q81. Write the NumPy program to copy data from a given array to another array?

In [0]:
```
import numpy as np
x = np.array([24, 27, 30, 29, 18, 14])
print('Original array-')
print(x)
y = np.copy(x)
print('Copy of the said array-')
print(y)
```

```
Original array-
[24 27 30 29 18 14]
Copy of the said array-
[24 27 30 29 18 14]
```

Q82. Write theNumPy program to calculate the sum of all columns of the2D numpy array?

> Indented block

In [0]:
```
import numpy as np
x = np.array([[ 0 ,1, 2, 3, 4, 5, 6, 7, 8],
              [ 9, 10, 11, 12, 13, 14, 15, 16, 17],
              [18, 19, 20, 21, 22, 23, 24, 25, 26],
              [27, 28 ,29, 30, 31, 32 ,33 ,34 ,35]])
print('Original array-')
print(x)
print('Sum of all columns-')
x.sum(axis=0)
```

```
Original array-
[[ 0  1  2  3  4  5  6  7  8]
 [ 9 10 11 12 13 14 15 16 17]
 [18 19 20 21 22 23 24 25 26]
 [27 28 29 30 31 32 33 34 35]]
Sum of all columns-
```

Out[4]:
```
array([54, 58, 62, 66, 70, 74, 78, 82, 86])
```

Q83. Write theNumPy program to calculate averages without NaNs along thegiven array?

In [0]:
```
import numpy as np

arr1 = np.array([[10, 20 ,30], [40, 50, np.nan], [np.nan, 6, np.nan], [np.nan, np
print('Original array-')
print(arr1)

masked_arr = np.ma.masked_array(arr1, np.isnan(arr1))
average = np.mean(masked_arr, axis=1)

print('Averages without NaNs along the said array-')
print(result.filled(np.nan))
```

```
Original array-
[[10. 20. 30.]
 [40. 50. nan]
 [nan  6. nan]
 [nan nan nan]]
Averages without NaNs along the said array-
[20. 45.  6. nan]
```

Q84. Create two arrays of six elements. Write the NumPy program to count the number of instances of a value occurring in one array on the condition of another array.

In [0]:
```
import numpy as np

a = np.array( [ 10, -10, 10, -10, -10, 10] )
b = np.array( [0.85, 0.45, 0.9, 0.8, 0.12, 0.6 ] )
c = np.sum( (a == 10) & (b >= .6))
print('Number of instances of a value occurring in one array on the condition of
print(c)
```

```
Number of instances of a value occurring in one array on the condition of anoth
er
3
```

Q85.Write the NumPy program to convert a Python dictionary to a Numpy ndarray ?

In [0]:
```
import numpy as np
dict = {'column0': {'a': 1, 'b': 0.0, 'c': 0.0, 'd': 2.0},
        'column1': {'a': 3.0, 'b': 1, 'c': 0.0, 'd': -1.0},
        'column2': {'a': 4, 'b': 1, 'c': 5.0, 'd': -1.0},
        'column3': {'a': 3.0, 'b': -1.0, 'c': -1.0, 'd': -1.0}}
print('Original dictionary-')
print(dict)
print(type(dict))

print('ndarray-')
arr = np.array( [[ v for v in key.values()] for key in dict.values()] )
print(arr)
print(type(arr))
```

```
Original dictionary-
{'column0': {'a': 1, 'b': 0.0, 'c': 0.0, 'd': 2.0}, 'column1': {'a': 3.0, 'b':
1, 'c': 0.0, 'd': -1.0}, 'column2': {'a': 4, 'b': 1, 'c': 5.0, 'd': -1.0}, 'col
umn3': {'a': 3.0, 'b': -1.0, 'c': -1.0, 'd': -1.0}}
<class 'dict'>
ndarray-
[[ 1.  0.  0.  2.]
 [ 3.  1.  0. -1.]
 [ 4.  1.  5. -1.]
 [ 3. -1. -1. -1.]]
<class 'numpy.ndarray'>
```

Q86. Write the Numpy program to find and storethenon-zero unique rows inan array after comparingeach row with other row in the givenmatrix?

```
In [0]: import numpy as np
        x = np.array([[1, 1, 0],
                      [0, 0, 0],
                      [0, 2, 3],
                      [0, 0, 0],
                      [0, -1, 1],
                      [0, 0, 0]])
        print('Original array-')
        print(x)
        temp = {(0, 0, 0 )}
        result = []

        for idx, row in enumerate(map(tuple, x)):
            print(idx, row)
            if row not in temp:
                result.append(idx)
        print('Non-zero unique rows-')
        print(x[result])
```

```
Original array-
[[ 1  1  0]
 [ 0  0  0]
 [ 0  2  3]
 [ 0  0  0]
 [ 0 -1  1]
 [ 0  0  0]]
0 (1, 1, 0)
1 (0, 0, 0)
2 (0, 2, 3)
3 (0, 0, 0)
4 (0, -1, 1)
5 (0, 0, 0)
Non-zero unique rows-
[[ 1  1  0]
 [ 0  2  3]
 [ 0 -1  1]]
```

Q87. Write the NumPy program to multiply thematrix by another matrix of complex numbers and create a new matrix of complex numbers?

In [0]:
```
import numpy as np

x = np.array([1+2j, 3+4j])
print('First array-')
print(x)

y = np.array([5+6j, 7+8j])
print('Second array-')
print(y)

print('Product of above two arrays-')
z = np.vdot(x,y)
print(z)
```

```
First array-
[1.+2.j 3.+4.j]
Second array-
[5.+6.j 7.+8.j]
Product of above two arrays-
(70-8j)
```

Q88. Write a NumPy program to generate thematrix product of two Arrays?

In [0]:
```
import numpy as np
x = np.array([[1,0],[1,1] ])
y = np.array([[3,1],[2,2]])
print('x-')
print(x)
print('y-')
print(y)
print('Matrix product of above two arrays-')
print(np.matmul(x,y))
```

```
x-
[[1 0]
 [1 1]]
y-
[[3 1]
 [2 2]]
Matrix product of above two arrays-
[[3 1]
 [5 3]]
```

Q89. Write the NumPy program to findroots of the following Polynomials?

In [0]:
```
import numpy as np
print("Roots of the first polynomial-")
print(np.roots([1, -2, 1]))
print("Roots of the second polynomial-")
print(np.roots([1, -12, 10, 7, -10]))
```

```
Roots of the first polynomial:
[1. 1.]
Roots of the second polynomial:
[11.04461946+0.j        -0.8711421 +0.j         0.91326132+0.4531004j
  0.91326132-0.4531004j]
```

Q90. Write the NumPy program to calculate inverse of sine, cosine,and inverse tangent for all elements in a given array?

In [0]:
```
import numpy as np
x = np.array([-1., 0, 1.])
print("Inverse sine-", np.arcsin(x))
print("Inverse cosine-", np.arccos(x))
print("Inverse tangent-", np.arctan(x))
```

```
Inverse sine: [-1.57079633  0.          1.57079633]
Inverse cosine: [3.14159265 1.57079633 0.        ]
Inverse tangent: [-0.78539816  0.          0.78539816]
```

Q91.Write the NumPy program to calculate the difference betweeninneighbouring elements, element-wise of a given array?

In [0]:
```
import numpy as np
x = np.array([1, 3, 5, 7, 0])
print('Original array-')
print(x)
print('Difference between neighboring elements, element-wise of the said array-')
print(np.diff(x))
```

```
Original array-
[1 3 5 8 0]
Difference between neighboring elements, element-wise of the said array-
[ 2  2  3 -8]
```

Q93. Write the NumPy program to calculate the difference between in the maximum and the minimum values of a given array along the second axis?

In [0]:
```
import numpy as np
x = np.arange(12).reshape((2,6))
print('Original array-')
print(x)
r1 = np.ptp(x , 1)
r2 = np.max(x,1) - np.min(x,1)
print('Difference between the maximum and the minimum values of the said array-')
print(r1)
```

```
Original array-
[[ 0  1  2  3  4  5]
 [ 6  7  8  9 10 11]]
Difference between the maximum and the minimum values of the said array-
[5 5]
```

Q94. Write the NumPy program to compute the weighted of the given array?

In [0]:
```
import numpy as np
x = np.array([0, 1, 2, 3, 4])
print('Original array-')
print(x)
weights = np.arange(1, 6)
print('\nWeighted average of the said array-')
r1 = np.average(x, weights = weights)
print(r1)
```

```
Original array-
[0 1 2 3 4]

Weighted average of the said array-
2.6666666666666665
```

Q95. Write the NumPy program to compute the mean, standard deviation, and the variance of a given array along the second axis?

In [0]:
```
import numpy as np
x = np.arange(6)
print('Original array-')
print(x)

mean = np.mean(x)
avg = np.average(x)
print('\nMean-', mean)

std = np.std(x)
print('\nstd-', 1)
var = np.var(x)
print('\nvariance-', var)
```

```
Original array-
[0 1 2 3 4 5]

Mean- 2.5

std- 1

variance- 2.9166666666666665
```

Q96. Write the Numpy program to compute the covariance matrix of the two given arrays?

In [0]:
```
import numpy as np
x = np.array([0, 1, 2])
y = np.array([2, 1, 0])
print('\nOriginal array1-')
print(x)
print('\nOriginal array-2')
print(y)
print('\nCovariance matrix of the said arrays:\n')
print(np.cov(x,y))
```

```

Original array1-
[0 1 2]

Original array-2
[2 1 0]

Covariance matrix of the said arrays:

[[ 1. -1.]
 [-1.  1.]]
```

Q97.Write a NumPy program to compute the cross-correlation of two given arrays ?

In [2]:
```
import numpy as np
x = np.array([0, 1, 3])
y = np.array([2, 4, 5])
print('\nOriginal array1-')
print(x)
print('\nOriginal array2-')
print(y)
print('\nCross-correlation of the said arrays-')
print(np.cov(x, y))
```

```

Original array1-
[0 1 3]

Original array2-
[2 4 5]

Cross-correlation of the said arrays-
[[2.33333333 2.16666667]
 [2.16666667 2.33333333]]
```

Q98. Write theNumPy program to compute Pearson product-moment correlation coefficients of two given arrays?

In [4]:
```
import numpy as np
x = np.array([0, 1, 3])
y = np.array([2, 4, 5])
print('\nOriginal array1-')
print(x)
print('\nOriginal array2-')
print(y)
print('\nPearson product-moment correlation coefficients of the said arrays')
np.corrcoef(x, y)
```

```

Original array1-
[0 1 3]

Original array2-
[2 4 5]

Pearson product-moment correlation coefficients of the said arrays
```

Out[4]:
```
array([[1.        , 0.92857143],
       [0.92857143, 1.        ]])
```

Q99.Write the python program to count the number of occurrences of each value in a given array of non-negative integers?

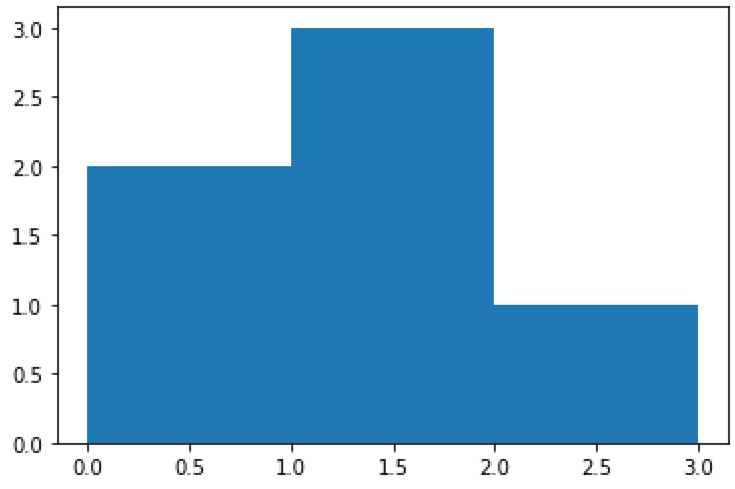
```
In [5]: import numpy as np
        x = [0, 1, 6, 1, 4, 1, 2, 2, 7]
        print('Original array')
        print(x)
        print('Number of occurrences of each value in array-')
        print(np.bincount(x))
```

```
Original array
[0, 1, 6, 1, 4, 1, 2, 2, 7]
Number of occurrences of each value in array-
[1 3 2 0 1 0 1 1]
```

Q100. Write a Numpy program to compute the histogram of nums against the bins?

```
In [7]: import numpy as np
        import matplotlib.pyplot as plt
        nums = np.array([0.5, 0.7, 1.0, 1.2, 1.3, 2.1])
        bins = np.array([0, 1, 2, 3])
        print('nums- ', nums)
        print('bins-', bins)
        print('Result-', np.histogram(nums, bins))
        plt.hist(nums, bins=bins)
        plt.show()
```

```
nums-  [0.5 0.7 1.  1.2 1.3 2.1]
bins- [0 1 2 3]
Result- (array([2, 3, 1]), array([0, 1, 2, 3]))
```



Q101. Write the Python program to add, subtract, multiplyand divide two pandas series?

In [12]:
```
import pandas as pd
s1 = pd.Series([2, 4, 6, 8, 10])
s2 = pd.Series([1, 3, 5, 7, 9])
print('s1 + s2')
print(s1+s2)
print('\ns1 - s2')
print(s1 + s2)
print('\ns1 * s2')
print(s1 * s2)
print('\ns1 / s2')
print(s1/s2)
```

```
s1 + s2
0     3
1     7
2    11
3    15
4    19
dtype: int64

s1 - s2
0     3
1     7
2    11
3    15
4    19
dtype: int64

s1 * s2
0     2
1    12
2    30
3    56
4    90
dtype: int64

s1 / s2
0    2.000000
1    1.333333
2    1.200000
3    1.142857
4    1.111111
dtype: float64
```

Q102.Write a Python program to convert a dictionary to thePandas Series?

In [19]:
```
import pandas as pd
dict = {'a': 100, 'b':200, 'c': 300, 'd': 400, 'e': 800}
s = pd.Series(dict)
print('Original dictionary-')
print(dict)
print('Converted series-')
print(s)
```

```
Original dictionary-
{'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}
Converted series-
a    100
b    200
c    300
d    400
e    800
dtype: int64
```

Q103.Write a python program to change the data typeof given a column or a Series?

In [23]:
```
import pandas as pd
s1 = pd.Series(['100', '200', 'python', '300.12', '400'])
print('Original Data Series-')
print(s1)
print('Change the said data type to numeric-')
s2 = pd.to_numeric(s1, errors='coerce' )
print(s2)
```

```
Original Data Series-
0       100
1       200
2    python
3    300.12
4       400
dtype: object
Change the said data type to numeric-
0    100.00
1    200.00
2       NaN
3    300.12
4    400.00
dtype: float64
```

Q104. Write the python pandas program to convert the first column of a DataFrame as a Series?

In [33]:
```
import pandas as pd
d = {'col1': [1, 2, 3, 4, 7, 11], 'col2': [4, 5, 6, 9, 5, 0], 'col3': [7, 5, 8, 1
df = pd.DataFrame(d)
print('Original DataFrame-')
print(df)

s = pd.Series(df['col1'])
print('1st column as a Series-')
print(s)
print(type(s))
```

```
Original DataFrame-
   col1  col2  col3
0     1     4     7
1     2     5     5
2     3     6     8
3     4     9    12
4     7     5     1
5    11     0    11
1st column as a Series-
0     1
1     2
2     3
3     4
4     7
5    11
Name: col1, dtype: int64
<class 'pandas.core.series.Series'>
```

Q105. Write a pandas program to create the mean and standard deviation of the data of a given Series?

In [35]:
```
import pandas as pd
s = pd.Series([1,2,3,4,5,6,7,8,9,5,3])
print('Original Data Series:')
print(s)
print('Mean of the said Data Series-')
print(s.mean())
print('Standard deviation of the said Data Series-')
print(s.std())
```

```
Original Data Series:
0     1
1     2
2     3
3     4
4     5
5     6
6     7
7     8
8     9
9     5
10    3
dtype: int64
Mean of the said Data Series-
4.818181818181818
Standard deviation of the said Data Series-
2.522624895547565
```

Q106.Write a pandas program to get powers of an array values element-wise?

In [40]:
```
import pandas as pd
import numpy as np
df = pd.DataFrame( {'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,72,8
print(df)
np.power(df, 3)
```

```
    X   Y   Z
0  78  84  86
1  85  94  97
2  96  89  96
3  80  83  72
4  86  86  83
```

Out[40]:

	X	Y	Z
0	474552	592704	636056
1	614125	830584	912673
2	884736	704969	884736
3	512000	571787	373248
4	636056	636056	571787

Q107. Write thepandas program to get the first 3 rows of a given DataFrame?

In [52]:
```
import pandas as pd

exam_data= {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael
            'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
            'attempts':[1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
            'qualify':['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no','no','y
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data, index = labels)
print('\nFirst three rows of the data frame-')
df1 = df.iloc[:3]
print(df1[ ['attempts','name', 'qualify', 'score'] ])
```

```

First three rows of the data frame-
   attempts       name qualify  score
a         1  Anastasia     yes   12.5
b         3       Dima      no    9.0
c         2  Katherine     yes   16.5
```

Q108: Write thepandas program to select the specified columns and the rows from a given data frame?

In [56]:
```
import pandas as pd

exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michae
             'score':[12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
             'attempts':[1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
             'qualify':['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no',

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data, index = labels)
df.loc[['b','d','f','g'],['name','score']]
```

Out[56]:

	name	score
b	Dima	9.0
d	James	NaN
f	Michael	20.0
g	Matthew	14.5

Q109.Write the pandas program to calculate mean score for each different student in DataFrame?

In [71]:
```
import pandas as pd

exam_data= {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael
            'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
            'attempts':[1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
            'qualify':['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no',

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data, index= labels)

print('Mean score for each different student in data frame')
print(df['score'].mean())
```

```
Mean score for each different student in data frame
13.5625
```

Q110.Write thePandas program to rename columns of a given DataFrame?

In [78]:
```
import pandas as pd
data = {'col1':[1, 2, 3], 'col2':[4, 5, 6], 'col3':[7, 8, 9]}
df = pd.DataFrame(data)
print('Original DataFrame')
print(df)

df = df.rename(columns={'col1':'Column1', 'col2':'Column2', 'col3':'Column3'})
print('New dataframe after renaming columns')
print(df)
```

```
Original DataFrame
   col1  col2  col3
0     1     4     7
1     2     5     8
2     3     6     9
New dataframe after renaming columns
   Column1  Column2  Column3
0        1        4        7
1        2        5        8
2        3        6        9
```

Q111.Write a pandas program to count city-wise number of people from a given of data set (city,name of the person)?

In [88]:
```
import pandas as pd
data = { 'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',
        'city': ['California', 'Los Angeles', 'California', 'California', 'Califo
df = pd.DataFrame(data)
g1 = df.groupby(by=['city']).size().reset_index(name='Number of people')

print(g1)
```

```
          city  Number of people
0   California                 4
1      Georgia                 2
2  Los Angeles                 4
```

Q112. Write a pandas program to widen output display to see more columns?

In [97]:
```
import pandas as pd
d = {'col1': [1, 4, 3, 4, 5], 'col2': [4, 5, 6, 7, 8], 'col3': [7, 8, 9, 0, 1]}
df = pd.DataFrame(data=d)
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
print('Original DataFrame')
print(df)
```

```
Original DataFrame
   col1  col2  col3
0     1     4     7
1     4     5     8
2     3     6     9
3     4     7     0
4     5     8     1
```

Q113.Write a pandas program to convert the dataframe column type from string to DateTime?

```
In [103]: import pandas as pd
          date = ['3/11/2000', '3/12/2000', '3/13/2000']
          s = pd.Series(date)
          print('String Date-')
          print(s)
          r = pd.to_datetime(s)
          df = pd.DataFrame(r)
          print('Original DataFrame (string to datetime)')
          print(df)
```

```
String Date-
0    3/11/2000
1    3/12/2000
2    3/13/2000
dtype: object
Original DataFrame (string to datetime)
           0
0 2000-03-11
1 2000-03-12
2 2000-03-13
```

Q114. Write a pandas program to append the data to an empty DataFrame?

```
In [110]: import pandas as pd
          import numpy as np
          df = pd.DataFrame()
          data = pd.DataFrame({'col1':range(3), 'col2':range(3)})
          df = df.append(data)
          print('Original DataFrame-After appending some data')
          print(df)
```

```
Original DataFrame-After appending some data
   col1  col2
0     0     0
1     1     1
2     2     2
```

Q115. Write a pandas program to count the number of columns of a DataFrame?

In [114]:
```
import pandas as pd
d = {'col1': [1, 2, 3, 4, 7], 'col2': [4, 5, 6, 9, 5], 'col3': [7, 8, 12, 1, 11]}
df = pd.DataFrame(d)
print('Original DataFrame')
print(df)
print('\nNumber of columns-', len(df.columns))
```

```
Original DataFrame
   col1  col2  col3
0     1     4     7
1     2     5     8
2     3     6    12
3     4     9     1
4     7     5    11

Number of columns- 3
```

Q116. Write a Pandas program to remove the last n rows of a given DataFrame ?

In [119]:
```
import pandas as pd

data = {'col1': [1, 2, 3, 4, 7, 11],
     'col2': [4, 5, 6, 9, 5, 0],
     'col3': [7, 5, 8, 12, 1,11]}

df =pd.DataFrame(data)
print('Original DataFrame')
print(df)
print("\nAfter removing last 3 rows of the said DataFrame-")
df1 = df.iloc[:3]
print(df1)
```

```
Original DataFrame
   col1  col2  col3
0     1     4     7
1     2     5     5
2     3     6     8
3     4     9    12
4     7     5     1
5    11     0    11

After removing last 3 rows of the said DataFrame-
   col1  col2  col3
0     1     4     7
1     2     5     5
2     3     6     8
```

Q117. Write a Pandas program to import excel data (coalpublic2013.xlsx) into a Pandas dataframe.

In [1]:
```
import pandas as pd
import numpy as np
df = pd.read_excel('coalpublic2013.xlsx')
print(df.head())
```

```
   Year  MSHA ID                      Mine_Name Production Labor_Hours
0  2013   103381           Tacoa Highwall Miner      56004       22392
1  2013   103404               Reid School Mine      28807        8447
2  2013   100759  North River #1 Underground Min  14,40,115    4,74,784
3  2013   103246                     Bear Creek      87587       29193
4  2013   103451                    Knight Mine   1,47,499       46393
```

Q118. Write a Pandas program to import excel data (coalpublic2013.xlsx) into a dataframe and find details where "Mine Name" starts with "P.

In [1]:
```
import pandas as pd
import numpy as np

df = pd.read_excel('coalpublic2013.xlsx')
mask = df['Mine_Name'].str.startswith('P')
print(df[mask])
```

```
    Year  MSHA ID                     Mine_Name  Production  Labor_Hours
13  2013   103332                  Powhatan Mine      140521        61394
18  2013   102976  Piney Woods Preparation Plant           0        14828
19  2013   102976  Piney Woods Preparation Plant           0        23193
```

Q119. Write a Pandas program to import excel data (employee.xlsx)into a Pandas dataframe and find the list of employees where hire_date> 01-01-07.

In [3]:
```
import pandas as pd
import numpy as np
df = pd.read_excel('employee.xlsx')
df[df['hire_date']>'01-01-07']
```

Out[3]:

	emp_id	first_name	last_name	hire_date
4	104	Bruce	Ernst	2007-05-21
7	107	Diana	Lorentz	2007-02-07
13	113	Luis	Popp	2007-12-07
19	119	Karen	Colmenares	2007-08-10

Q120.Write a Pandas program to import excel data (employee.xlsx) into a Pandas dataframe and find a list of the employees of a specified year

In [4]:
```
import pandas as pd
import numpy as np
df = pd.read_excel('employee.xlsx')
df[pd.DatetimeIndex(df['hire_date']).year == 2005]
```

Out[4]:

	emp_id	first_name	last_name	hire_date
1	101	Neena	Kochhar	2005-09-21
5	105	David	Austin	2005-06-25
10	110	John	Chen	2005-09-28
11	111	Ismael	Sciarra	2005-09-30
16	116	Shelli	Baida	2005-12-24
17	117	Sigal	Tobias	2005-07-24

Q121.Write a pandas program to import three datasheets from a given excel data (coalpublic2013.xlsx) in to a single dataframe.

In [9]:
```python
import pandas as pd
import numpy as np
df1 = pd.read_excel('coalpublic2013.xlsx', sheetname='Sheet-1')
df2 = pd.read_excel('coalpublic2013.xlsx', sheetname='Sheet-2')
df3 = pd.read_excel('coalpublic2013.xlsx', sheetname='Sheet-3')

df = pd.concat([df1, df2, df3], axis = 0)
df
```

Out[9]:

	Year	MSHA ID	Mine_Name	Production	Labor_Hours
0	2013	103381	Tacoa Highwall Miner	56004	22392
1	2013	103404	Reid School Mine	28807	28447
2	2013	100759	North River #1 Underground Min	1440115	474784
3	2013	103246	Bear Creek	87587	29193
4	2013	103451	Knight Mine	147499	46393
5	2013	103433	Crane Central Mine	69339	47195
6	2013	100329	Concord Mine	0	144002
7	2013	100851	Oak Grove Mine	2269014	1001809
8	2013	102901	Shoal Creek Mine	0	12396
9	2013	102901	Shoal Creek Mine	1453024	1237415
10	2013	103180	Sloan Mountain Mine	327780	196963
11	2013	103182	Fishtrap	175058	87314
12	2013	103285	Narley Mine	154861	90584
13	2013	103332	Powhatan Mine	140521	61394
14	2013	103375	Johnson Mine	580	1900
15	2013	103419	Maxine-Pratt Mine	125824	107469
16	2013	103432	Skelton Creek	8252	220
17	2013	103437	Black Warrior Mine No 1	145924	70926
18	2013	102976	Piney Woods Preparation Plant	0	14828
19	2013	102976	Piney Woods Preparation Plant	0	23193
0	2013	103155	Corinth Prep Plant	0	27996
1	2013	103155	Corinth Prep Plant	0	51994
2	2013	103195	Mccollum/Sparks Branch Mine	71910	17411
3	2013	103342	Reese's Branch Mine	263888	115123
4	2013	103370	Cresent Valley Mine	2860	621
5	2013	103372	Cane Creek Mine	66258	32401
6	2013	103376	Town Creek	299167	176499

	Year	MSHA ID	Mine_Name	Production	Labor_Hours
7	2013	103389	Carbon Hill Mine	76241	84966
8	2013	103410	Coal Valley Mine	407841	158591
9	2013	103423	Dutton Hill Mine	37275	9162
10	2013	1519322	Ghm #25	25054	3108
11	2013	103321	Poplar Springs	189370	76366
12	2013	103358	Old Union	284563	161805
13	2013	5000030	Usibelli	1631584	286079
14	2013	201195	Kayenta Mine	7602722	1015333
0	2013	103380	Calera	0	12621
1	2013	103380	Calera	0	1402
2	2013	103422	Clark No 1 Mine	122727	140250
3	2013	103467	Helena Surface Mine	59664	30539
4	2013	101247	No 4 Mine	2622528	1551141
5	2013	101401	No 7 Mine	5405412	2464719
6	2013	103172	Searles Mine No. 2, 3, 4, 5, 6	258078	119542
7	2013	103179	Fleetwood Mine No 1	75937	63745
8	2013	103303	Shannon Mine	317491	164388
9	2013	103323	Deerlick Mine	133452	46381
10	2013	103364	Brc Alabama No. 7 Llc	0	14324
11	2013	103436	Swann's Crossing	137511	77190
12	2013	100347	Choctaw Mine	537429	215295
13	2013	101362	Manchester Mine	219457	116914
14	2013	102996	Jap Creek Mine	375715	164093

Q122.Write a pandas program to import three datasheets from a given excel data (employee.xlsx) into a single dataframe and export the result into new Excel file.

In [19]:
```
import pandas as pd
import numpy as np

df1 = pd.read_excel('employee.xlsx', sheetname='Sheet1')
df2 = pd.read_excel('employee.xlsx', sheetname='Sheet2')
df3 = pd.read_excel('employee.xlsx', sheetname='Sheet3')
df = pd.concat([df1, df2, df3])
df.to_excel('employee-1.xlsx', index = False)
```

Q123.Write a pandas program to create the Pivot table with multiple indexes from the data set of the titanic.csv.

In [48]:
```
import pandas as pd
import numpy as np
df = pd.read_csv('titanic.csv')
print(pd.pivot_table(df, index = ['sex', 'age'], aggfunc=np.sum))
```

```
                adult_male  alone      fare  parch  pclass  sibsp  survived
sex    age
female 0.75            0.0    0.0   38.5166      2       6      4         2
       1.00            0.0    0.0   26.8750      3       6      1         2
       2.00            0.0    0.0  259.4750      9      15      9         2
       3.00            0.0    0.0   62.6542      3       5      4         1
       4.00            0.0    0.0  114.1417      6      13      4         5
       5.00            0.0    1.0   90.8708      5      11      7         4
       6.00            0.0    0.0   64.2750      3       5      4         1
       7.00            0.0    0.0   26.2500      2       2      0         1
       8.00            0.0    0.0   47.3250      3       5      3         1
       9.00            0.0    0.0  108.7958      7      12     10         0
       10.00           0.0    0.0   24.1500      2       3      0         0
       11.00           0.0    0.0   31.2750      2       3      4         0
       13.00           0.0    1.0   26.7292      1       5      0         2
       14.00           0.0    1.0  169.1667      2       9      3         3
       14.50           0.0    0.0   14.4542      0       3      1         0
       15.00           0.0    2.0  241.0459      1      10      1         4
       16.00           0.0    3.0  246.2625      4      12      5         5
       17.00           0.0    3.0  210.7833      2      12      6         5
       18.00           0.0    4.0  697.0167      9      31      6         8
       19.00           0.0    3.0  215.0959      2      13      3         7
       20.00           0.0    1.0   18.4875      0       6      1         0
       21.00           0.0    4.0  410.4333      4      16      5         4
       22.00           0.0    7.0  444.1084      6      26      3        10
       23.00           0.0    3.0  405.5417      2      10      4         4
       24.00           0.0    7.0  772.1708     15      31     10        14
       25.00           0.0    1.0  223.2500      4      11      3         2
       26.00           0.0    3.0  136.7292      1      12      2         3
       27.00           0.0    2.0   76.8916      3      15      2         5
       28.00           0.0    4.0  110.9458      1      16      3         5
       29.00           0.0    2.0  320.6208      7      16      3         5
...                    ...    ...       ...    ...     ...    ...       ...
male   42.00          10.0    6.0  216.1084      1      21      3         3
       43.00           3.0    2.0   40.7500      1       8      1         0
       44.00           6.0    3.0  156.1250      1      15      3         1
       45.00           6.0    5.0  187.1000      0      10      1         2
       45.50           2.0    2.0   35.7250      0       4      0         0
       46.00           3.0    2.0  166.3750      0       4      1         0
       47.00           7.0    7.0  181.3583      0      12      0         0
       48.00           5.0    3.0  176.1334      0       8      2         3
       49.00           4.0    1.0  256.9167      1       6      3         2
       50.00           5.0    2.0  317.0250      0       8      4         1
       51.00           6.0    5.0  123.3084      1      13      0         1
       52.00           4.0    3.0  136.6500      1       6      1         1
       54.00           5.0    3.0  195.1500      1       8      1         0
       55.00           1.0    1.0   30.5000      0       1      0         0
       55.50           1.0    1.0    8.0500      0       3      0         0
       56.00           3.0    3.0   92.7458      0       3      0         1
       57.00           1.0    1.0   12.3500      0       2      0         0
       58.00           2.0    1.0  142.9750      2       2      0         0
       59.00         2.0    2.0   20.7500      0       5      0          0
       60.00         3.0    1.0  144.7500      2       4      2          1
       61.00         3.0    3.0   72.0583      0       5      0          0
       62.00         3.0    3.0   63.6000      0       4      0          1
       64.00         2.0    1.0  289.0000      4       2      1          0
       65.00         3.0    2.0   96.2792      1       5      0          0
       66.00         1.0    1.0   10.5000      0       2      0          0
       70.00         2.0    1.0   81.5000      1       3      1          0
       70.50         1.0    1.0    7.7500      0       3      0          0
       71.00         2.0    2.0   84.1584      0       2      0          0
       74.00         1.0    1.0    7.7750      0       3      0          0
       80.00         1.0    1.0   30.0000      0       1      0          1

[145 rows x 7 columns]
```

Q124. Write a Pandas program to create thePivot table and find survival rate by gender?

In [56]:
```
import pandas as pd
import numpy as np
df = pd.read_csv('titanic.csv')
pd.pivot_table(df, index=['sex'], values=['survived'],aggfunc=np.sum)
```

Out[56]:

	survived
sex	
female	233
male	109

Q125.Write a pandas program to make partition each of the passengers into 4 categories based on their age.

In [16]:
```
import numpy as np
import pandas as pd

df = pd.read_csv('titanic.csv')
result = pd.cut(df['age'], bins= [0, 10, 30, 60, 80])
print(result)
```

```
0      (10, 30]
1      (30, 60]
2      (10, 30]
3      (30, 60]
4      (30, 60]
5           NaN
6      (30, 60]
7       (0, 10]
8      (10, 30]
9      (10, 30]
10      (0, 10]
11     (30, 60]
12     (10, 30]
13     (30, 60]
14     (10, 30]
15     (30, 60]
16      (0, 10]
17          NaN
18     (30, 60]
19          NaN
20     (30, 60]
21     (30, 60]
22     (10, 30]
23     (10, 30]
24      (0, 10]
25     (30, 60]
26          NaN
27     (10, 30]
28          NaN
29          NaN
         ...
861    (10, 30]
862    (30, 60]
863         NaN
864    (10, 30]
865    (30, 60]
866    (10, 30]
867    (30, 60]
868         NaN
869     (0, 10]
870    (10, 30]
871    (30, 60]
872    (30, 60]
873    (30, 60]
874    (10, 30]
875    (10, 30]
876    (10, 30]
877    (10, 30]
878         NaN
```

```
879    (30, 60]
880    (10, 30]
881    (30, 60]
882    (10, 30]
883    (10, 30]
884    (10, 30]
885    (30, 60]
886    (10, 30]
887    (10, 30]
888         NaN
889    (10, 30]
890    (30, 60]
Name: age, dtype: category
Categories (4, object): [(0, 10] < (10, 30] < (30, 60] < (60, 80]]
```

Q126.Write a pandas program to create thePivot table and find survival rate by the gender, age of the different categories of various classes.

In [19]:
```
import numpy as np
import pandas as pd
df = pd.read_csv('titanic.csv')
age = pd.cut(df['age'], bins = [0, 20, 55])
df1 = df.pivot_table('survived', index=['sex', age], columns='class')
df1
```

Out[19]:

	class	First	Second	Third
sex	**age**			
female	**(0, 20]**	0.928571	1.000000	0.510638
	(20, 55]	0.968750	0.912281	0.407407
male	**(0, 20]**	0.571429	0.526316	0.197368
	(20, 55]	0.440000	0.054054	0.134503

Q127.Write a pandas program to create thePivot table and calculate number of women and men were in a particular cabin class.

In [27]:
```
import numpy as np
import pandas as pd

df = pd.read_csv('titanic.csv')
df1 = df.pivot_table(index=['sex'], columns=['pclass'], aggfunc='count')
df1
```

Out[27]:

	survived			age			sibsp			parch	...	deck	embark_town			alive	
pclass	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**...**	**3**	**1**	**2**	**3**	**1**	**2**
sex																	
female	94	76	144	85	74	102	94	76	144	94	...	6	92	76	144	94	76
male	122	108	347	101	99	253	122	108	347	122	...	6	122	108	347	122	108

2 rows × 39 columns

Q128.Write a pandas program to create thePivot table and separate the gender according to whether they travelled alone or not to get the probability of survival

In [38]:
```
import numpy as np
import pandas as pd

df = pd.read_csv('titanic.csv')
df1 = df.pivot_table('survived', index=['sex', 'alone'])
df1
```

Out[38]:
```
sex     alone
female  False    0.712766
        True     0.785714
male    False    0.271084
        True     0.155718
Name: survived, dtype: float64
```

Q129.Write a pandas program to create thePivot table and find the probability of survival by class, gender, solo boarding,and the port of embarkation.

In [43]:
```
import numpy as np
import pandas as pd
df = pd.read_csv('titanic.csv')
df1 = df.pivot_table('survived', index=['sex', 'alone'], columns=['embark_town',
print(df1)
```

```
embark_town  Cherbourg                      Queenstown                   \
class            First    Second     Third      First Second     Third
sex    alone
female False  1.000000  1.000000  0.611111        1.0    NaN  0.625000
       True   0.944444  1.000000  0.800000        NaN    1.0  0.760000
male   False  0.473684  0.166667  0.500000        0.0    NaN  0.100000
       True   0.347826  0.250000  0.151515        NaN    0.0  0.068966

embark_town  Southampton
class              First    Second     Third
sex    alone
female False    0.941176  0.923077  0.327586
       True     1.000000  0.892857  0.466667
male   False    0.407407  0.300000  0.142857
       True     0.326923  0.089552  0.123762
```

Q130.Write a pandas program to get current date, oldest date and number of days between Current date andthe oldest date of Ufo dataset.

In [178]:
```
import numpy as np
import pandas as pd
from datetime import datetime

df = pd.read_csv('ufo.csv')
for i in np.arange(df['datetime'].count()):
    try:
        #trying to convert datetime with type of object to datetime64
        # if there is error in the object convert those object into correct forma
        datetime.strptime(df['datetime'][i], '%m/%d/%Y %H:%M')
    except:
        df['datetime'][i] = df['datetime'][i].split()[0] + " 00:00"
        continue
df.to_csv('ufo-1.csv')
df['datetime'] = pd.to_datetime(df['datetime'])


print("\nCurrent date of Ufo dataset:")
print(df.datetime.max())
print("\nOldest date of Ufo dataset:")
print(df.datetime.min())
print("\nNumber of days between Current date and oldest date of Ufo dataset:")
print((df.datetime.max() - df.datetime.min()).days)
```

```
C:\Program Files (x86)\Anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py:2717: DtypeWarning: Columns (6,9) have mixed types. Specify dtype option o
n import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
C:\Program Files (x86)\Anaconda3\lib\site-packages\ipykernel\__main__.py:12: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stab
le/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy)


Current date of Ufo dataset:
2014-05-08 18:45:00

Oldest date of Ufo dataset:
1906-11-11 00:00:00

Number of days between Current date and oldest date of Ufo dataset:
39260
```

Q131.Write a pandas program to get all sighting days of the unidentified flying object (ufo) between 1950-10-10 and 1960-10-10.

In [194]:
```
import numpy as np
import pandas as pd
df = pd.read_csv('ufo-1.csv')
selected_period = df[(df['datetime'] >= '01-01-1950 00:00:00') & (df['datetime']
selected_period
```

```
C:\Program Files (x86)\Anaconda3\lib\site-packages\IPython\core\interactivesh
ell.py:2717: DtypeWarning: Columns (5,8) have mixed types. Specify dtype opti
on on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Q132.Write a Pandas program to extract the year, month, day, hour,minute, second,and weekday from unidentified flying object (UFO) reporting date.

In [202]:
```
print('Years')
print(pd.DatetimeIndex(df['datetime']).year)
print('\nMonth')
print(pd.DatetimeIndex(df['datetime']).month)
print('\nDay')
print(pd.DatetimeIndex(df['datetime']).day)
print('\nHour')
print(pd.DatetimeIndex(df['datetime']).hour)
print('\nMinute')
print(pd.DatetimeIndex(df['datetime']).minute)
print('\nSecond')
print(pd.DatetimeIndex(df['datetime']).second)
print('\nWeekday')
print(pd.DatetimeIndex(df['datetime']).weekday_name)
```

```
[1949 1949 1955 ..., 2013 2013 2013]
```

Q133.Write a pandas program to count year-country wise frequency of reporting dates of the unidentified flying object(UFO).

In [216]:
```
import numpy as np
import pandas as pd
df = pd.read_csv('ufo-1.csv')

print('Original DataFrame')
print(df.head())

df['Year'] = pd.DatetimeIndex(df['datetime']).year
result = df.groupby(['Year', 'country']).size()
print('count year-country wise frequency of reporting dates of the unidentified f
print(result)
```

```
C:\Program Files (x86)\Anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py:2717: DtypeWarning: Columns (5,8) have mixed types. Specify dtype option o
n import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)


Original DataFrame
           datetime                  city state country     shape  \
0  10/10/1949 20:30            san marcos    tx      us  cylinder
1  10/10/1949 21:00          lackland afb    tx     NaN     light
2  10/10/1955 17:00  chester (uk/england)   NaN      gb    circle
3  10/10/1956 21:00                  edna    tx      us    circle
4  10/10/1960 20:00               kaneohe    hi      us     light

  duration (seconds)                                           comments  \
0               2700  This event took place in early fall around 194...
1               7200  1949 Lackland AFB&#44 TX.  Lights racing acros...
2                 20  Green/Orange circular disc over Chester&#44 En...
3                 20  My older brother and twin sister were leaving ...
4                900  AS a Marine 1st Lt. flying an FJ4B fighter/att...

  date posted    latitude   longitude
0   4/27/2004  29.8830556  -97.941111
1  12/16/2005    29.38421  -98.581082
2   1/21/2008        53.2   -2.916667
3   1/17/2004  28.9783333  -96.645833
4   1/22/2004  21.4180556 -157.803611
count year-country wise frequency of reporting dates of the unidentified flying
object(UFO)
Year  country
1910  us           3
1920  us           1
1925  us           1
1929  us           1
1930  us           1
1931  us           2
1934  us           1
1936  ca           1
      us           2
1937  us           2
1939  us           3
1941  us           1
1942  us           3
1943  gb           1
```

```
      us          1
1944  us          3
1945  us          7
1946  ca          1
      us          8
1947  us         38
1948  us          8
1949  us         16
1950  us         24
1951  ca          2
      gb          1
      us         14
1952  ca          3
      gb          1
      us         42
1953  ca          2
                ...
2009  au         15
      ca        151
      de          6
      gb        213
      us       3938
2010  au         17
      ca        154
      de          4
      gb        119
      us       3825
2011  au         14
      ca        140
      de          3
      gb         56
      us       4666
2012  au         20
      ca        251
      de          6
      gb         87
      us       6749
2013  au         35
      ca        273
      de          6
      gb         52
      us       6464
2014  au         14
      ca         48
      de          3
      gb         22
      us       2119
dtype: int64
```

Q134.Write a pandas program to get the difference (in days) between documented date and reporting date of unidentified flying object (UFO).

In [226]:
```
import numpy as np
import pandas as pd
df = pd.read_csv('ufo-1.csv')
df['datetime'] = pd.to_datetime(df['datetime'])
df['date posted'] = pd.to_datetime(df['date posted'])
df['Difference'] = (df['date posted'] - df['datetime']).dt.days
print(df.head())
```

```
C:\Program Files (x86)\Anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py:2717: DtypeWarning: Columns (5,8) have mixed types. Specify dtype option o
n import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)

             datetime                  city state country     shape  \
0 1949-10-10 20:30:00            san marcos    tx      us  cylinder
1 1949-10-10 21:00:00          lackland afb    tx     NaN     light
2 1955-10-10 17:00:00  chester (uk/england)   NaN      gb    circle
3 1956-10-10 21:00:00                  edna    tx      us    circle
4 1960-10-10 20:00:00               kaneohe    hi      us     light

  duration (seconds)                                           comments  \
0               2700  This event took place in early fall around 194...
1               7200  1949 Lackland AFB&#44 TX.  Lights racing acros...
2                 20  Green/Orange circular disc over Chester&#44 En...
3                 20  My older brother and twin sister were leaving ...
4                900  AS a Marine 1st Lt. flying an FJ4B fighter/att...

  date posted    latitude   longitude  Difference
0  2004-04-27  29.8830556  -97.941111       19922
1  2005-12-16    29.38421  -98.581082       20520
2  2008-01-21        53.2   -2.916667       19095
3  2004-01-17  28.9783333  -96.645833       17264
4  2004-01-22  21.4180556 -157.803611       15808
```

Q135.Write a pandas program to generate sequences of fixed-frequency dates and time spans.

In [235]:
```
import pandas as pd
dtr = pd.date_range('2019-01-01', periods=12, freq='H')
print('Hourly Frequence')
print(dtr)

dtr = pd.date_range('2019-01-01', periods=12, freq='min')
print('\nMinutely Frequence')
print(dtr)

dtr = pd.date_range('2019-01-01', periods=12, freq='S')
print('\nSecondly Frequence')
print(dtr)
```

```
Hourly Frequence
DatetimeIndex(['2019-01-01 00:00:00', '2019-01-01 01:00:00',
               '2019-01-01 02:00:00', '2019-01-01 03:00:00',
               '2019-01-01 04:00:00', '2019-01-01 05:00:00',
               '2019-01-01 06:00:00', '2019-01-01 07:00:00',
               '2019-01-01 08:00:00', '2019-01-01 09:00:00',
               '2019-01-01 10:00:00', '2019-01-01 11:00:00'],
              dtype='datetime64[ns]', freq='H')

Minutely Frequence
DatetimeIndex(['2019-01-01 00:00:00', '2019-01-01 00:01:00',
               '2019-01-01 00:02:00', '2019-01-01 00:03:00',
               '2019-01-01 00:04:00', '2019-01-01 00:05:00',
               '2019-01-01 00:06:00', '2019-01-01 00:07:00',
               '2019-01-01 00:08:00', '2019-01-01 00:09:00',
               '2019-01-01 00:10:00', '2019-01-01 00:11:00'],
              dtype='datetime64[ns]', freq='T')

Secondly Frequence
DatetimeIndex(['2019-01-01 00:00:00', '2019-01-01 00:00:01',
               '2019-01-01 00:00:02', '2019-01-01 00:00:03',
               '2019-01-01 00:00:04', '2019-01-01 00:00:05',
               '2019-01-01 00:00:06', '2019-01-01 00:00:07',
               '2019-01-01 00:00:08', '2019-01-01 00:00:09',
               '2019-01-01 00:00:10', '2019-01-01 00:00:11'],
              dtype='datetime64[ns]', freq='S')
```

Q136.Write a pandas program to manipulate and convert date times with timezone information.

In [247]:
```
import pandas as pd
dt = pd.date_range('2019-01-01', periods=12, freq='H')
dt = dt.tz_localize('UTC')
print(dt)
print("\nFrom UTC to Asia/India:")
dt = dt.tz_convert('Asia/Kolkata')
print(dt)
```

```
DatetimeIndex(['2019-01-01 00:00:00+00:00', '2019-01-01 01:00:00+00:00',
               '2019-01-01 02:00:00+00:00', '2019-01-01 03:00:00+00:00',
               '2019-01-01 04:00:00+00:00', '2019-01-01 05:00:00+00:00',
               '2019-01-01 06:00:00+00:00', '2019-01-01 07:00:00+00:00',
               '2019-01-01 08:00:00+00:00', '2019-01-01 09:00:00+00:00',
               '2019-01-01 10:00:00+00:00', '2019-01-01 11:00:00+00:00'],
              dtype='datetime64[ns, UTC]', freq='H')

From UTC to Asia/India:
DatetimeIndex(['2019-01-01 05:30:00+05:30', '2019-01-01 06:30:00+05:30',
               '2019-01-01 07:30:00+05:30', '2019-01-01 08:30:00+05:30',
               '2019-01-01 09:30:00+05:30', '2019-01-01 10:30:00+05:30',
               '2019-01-01 11:30:00+05:30', '2019-01-01 12:30:00+05:30',
               '2019-01-01 13:30:00+05:30', '2019-01-01 14:30:00+05:30',
               '2019-01-01 15:30:00+05:30', '2019-01-01 16:30:00+05:30'],
              dtype='datetime64[ns, Asia/Kolkata]', freq='H')
```

Q137.Write a pandas program to create the graphical analysis of UFO (unidentified flying object) Sightings year.

In [5]:
```
import pandas as pd
import matplotlib.pyplot as plt

df =pd.read_csv('ufo-1.csv')
df['datetime'] = pd.to_datetime(df['datetime'])
df['date posted'] = pd.to_datetime(df['date posted'])
df['year'] = pd.DatetimeIndex(df['datetime']).year
years_data = df['year'].value_counts()
```

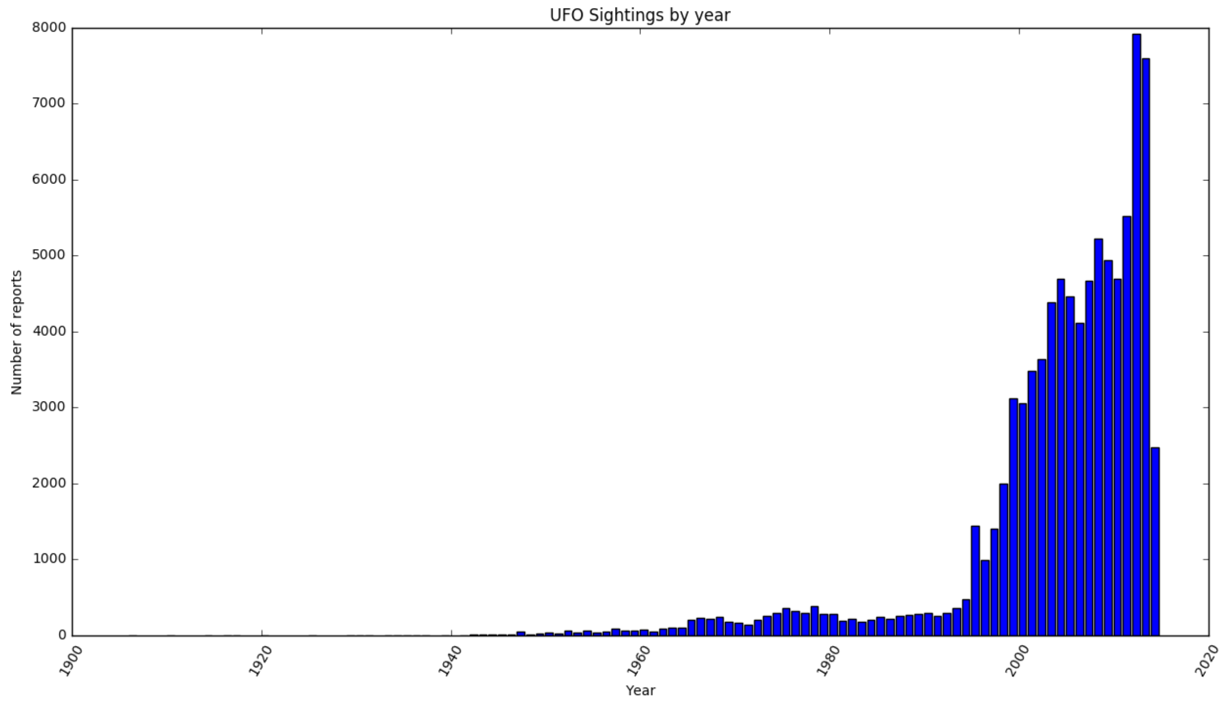
```
C:\Program Files (x86)\Anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py:2717: DtypeWarning: Columns (5,8) have mixed types. Specify dtype option o
n import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [11]:
```
years_index = years_data.index
```

In [13]:
```
year_values = years_data.get_values()
```

In [16]:
```
%matplotlib inline
plt.figure(figsize=(15,8))
plt.xticks(rotation = 60)
plt.title('UFO Sightings by year')
plt.xlabel('Year')
plt.ylabel('Number of reports')
plt.bar(years_index, year_values)
```

Out[16]:
```
<Container object of 90 artists>
```



Q138.Write a pandas program to create a comparison of the top 10 years in which the (UFO)was sighted VS each Month

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
df =pd.read_csv('ufo-1.csv')
df['datetime'] = pd.to_datetime(df['datetime'])
df['date posted'] = pd.to_datetime(df['date posted'])
```

```
C:\Program Files (x86)\Anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py:2717: DtypeWarning: Columns (5,8) have mixed types. Specify dtype option o
n import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [2]:
```
df['year'] = pd.DatetimeIndex(df['datetime']).year
years_data = df['year'].value_counts()
most_sightings_years  = years_data.head(10)
```

In [3]:
```
def is_top_years(year):
    if year in most_sightings_years.index:
        return year
```

In [6]:
```
columns=df['datetime'].dt.month
index=df['datetime'].dt.year.apply(is_top_years)
aggfunc='count'
values='city'

columns = columns.get_values()# columns.reset_index(level=1, drop=False,inplace=F
print(columns)
```

```
[10 10 10 ...,  9  9  9]
```

In [7]:
```
month_vs_year = df .pivot_table(columns=columns,index=index, aggfunc='count',valu
```

In [8]:
```
print("\nComparison of the top 10 years in which the UFO was sighted vs each mont
print(month_vs_year.head(10))
```

```

Comparison of the top 10 years in which the UFO was sighted vs each month:
           1    2    3    4    5    6    7    8    9    10   11   12
datetime
2003.0    328  266  191  225  251  306  449  542  505  473  470  383
2004.0    303  300  412  389  387  425  440  545  411  452  329  307
2005.0    256  279  357  314  300  400  456  352  531  482  465  273
2007.0    442  264  360  321  311  419  465  468  448  445  372  358
2008.0    472  369  340  442  343  475  570  502  383  530  453  346
2009.0    498  394  341  317  359  384  609  499  599  321  322  290
2010.0    292  186  260  294  328  379  837  527  449  471  361  306
2011.0    326  275  330  316  318  397  759  634  554  639  444  529
2012.0    579  388  529  496  513  750  926  886  755  668  773  661
2013.0    389  281  397  424  524  625  962  895  779  778  798  747
```

Q139.Write a pandas program to create a heatmap(rectangular data asa colour-encoded matrix) for comparison oftop 10 years in which (UFO)was sighted VSeach Month.

In [30]:
```
import pandas as pd
import matplotlib.pyplot as plt
df =pd.read_csv('ufo-1.csv')
df['datetime'] = pd.to_datetime(df['datetime'])

df['year'] = pd.DatetimeIndex(df['datetime']).year
years_data = df['year'].value_counts()
most_sightings_years  = years_data.head(10)

def is_top_years(year):
    if year in most_sightings_years.index:
        return year

columns=df['datetime'].dt.month
index=df['datetime'].dt.year.apply(is_top_years)
aggfunc='count'
values='city'

columns = columns.get_values()# columns.reset_index(level=1, drop=False,inplace=F
month_vs_year = df .pivot_table(columns=columns,index=index, aggfunc='count',valu
month_vs_year.columns = month_vs_year.columns.astype(int)
print("\nHeatmap for comparison of the top 10 years in which the UFO was sighted 
plt.figure(figsize=(10,8))
plt.imshow(month_vs_year, cmap='hot', interpolation='nearest')
plt.show()
```

```
C:\Program Files (x86)\Anaconda3\lib\site-packages\IPython\core\interactivesh
ell.py:2717: DtypeWarning: Columns (5,8) have mixed types. Specify dtype opti
on on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)


Heatmap for comparison of the top 10 years in which the UFO was sighted vs ea
ch month:
```

Q140.Write a pandas program to create a Timewheel of Hour VSYearcomparison of the top 10 years in which the (UFO)was sighted.

In [32]:
```
import pandas as pd
import matplotlib.pyplot as plt
df =pd.read_csv('ufo-1.csv')
df['datetime'] = pd.to_datetime(df['datetime'])
```

```
C:\Program Files (x86)\Anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py:2717: DtypeWarning: Columns (5,8) have mixed types. Specify dtype option o
n import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

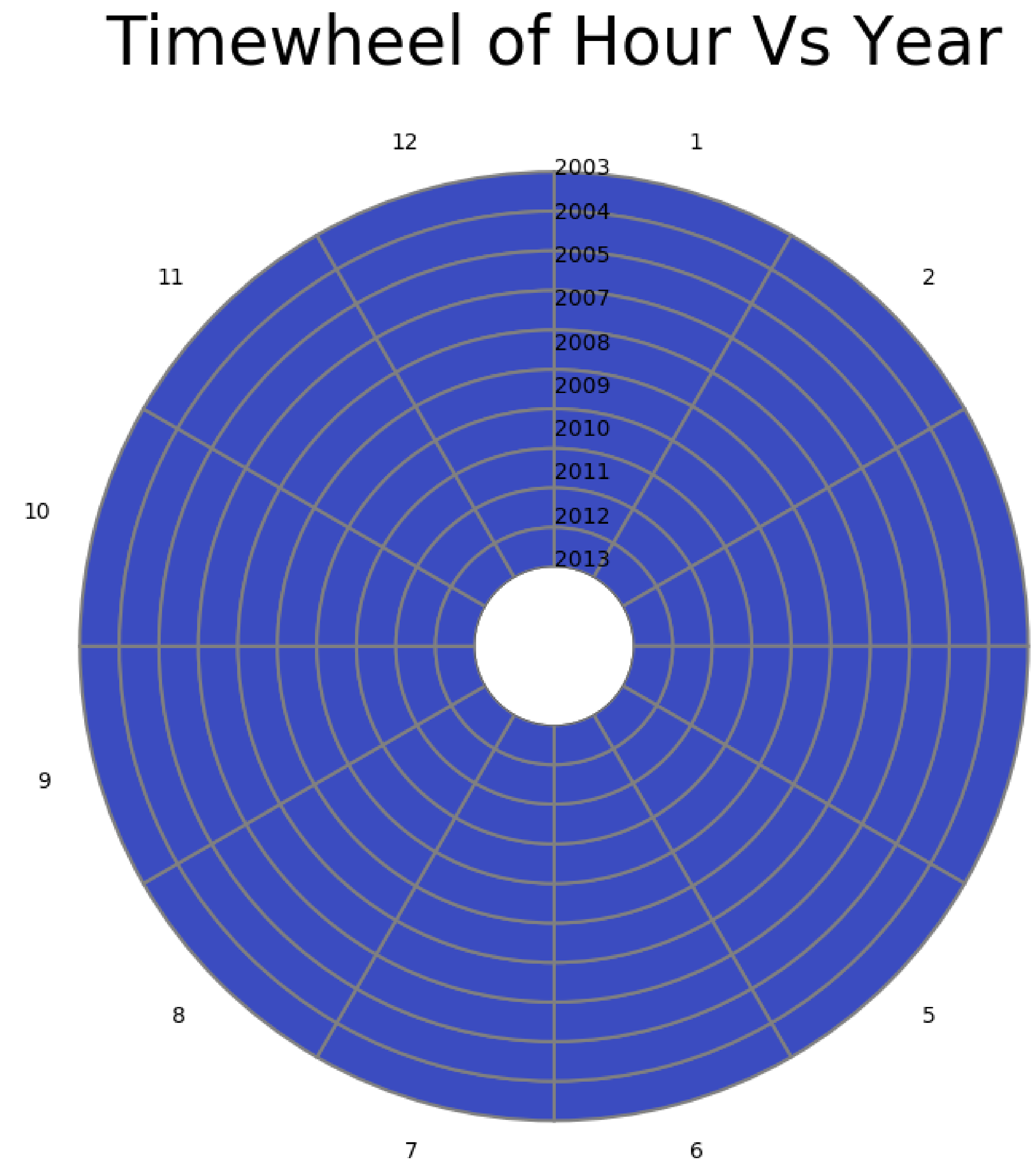
In [20]:
```
def is_top_years(year):
    if year in most_sightings_years.index:
        return year
```

In [33]:
```
most_sightigs_yeara = df['datetime'].dt.year.value_counts().head(10)


columns=df['datetime'].dt.month
index=df['datetime'].dt.year.apply(is_top_years)
columns = columns.get_values()
month_vs_year = df.pivot_table(columns=columns,
                               index=index,
                               aggfunc='count',
                               values='city'
                              )
```

In [34]:
```
month_vs_year.index = month_vs_year.index.astype(int)
month_vs_year.columns = month_vs_year.columns.astype(int)
```

In [35]:
```
import matplotlib as mpl
import matplotlib.cm as cm
%matplotlib inline
print("\nComparison of the top 10 years in which the UFO was sighted vs each mont
def pie_heatmap(table, cmap='coolwarm_r', vmin=None, vmax=None,inner_r=0.25, pie_
   n, m = table.shape
   vmin= table.min().min() if vmin is None else vmin
   vmax= table.max().max() if vmax is None else vmax

   centre_circle = plt.Circle((0,0),inner_r,edgecolor='black',facecolor='white',f
   plt.gcf().gca().add_artist(centre_circle)
   norm = mpl.colors.Normalize(vmin=vmin, vmax=vmax)
   cmapper = cm.ScalarMappable(norm=norm, cmap=cmap)

   for i, (row_name, row) in enumerate(table.iterrows()):
       labels = None if i > 0 else table.columns
       wedges = plt.pie([1] * m,radius=inner_r+float(n-i)/n, colors=[cmapper.to_r
           labels=labels, startangle=90, counterclock=False, wedgeprops={'linewid
       plt.setp(wedges[0], edgecolor='grey',linewidth=1.5)
       wedges = plt.pie([1], radius=inner_r+float(n-i-1)/n, colors=['w'], labels=
       plt.setp(wedges[0], edgecolor='grey',linewidth=1.5)
plt.figure(figsize=(8,8))
plt.title("Timewheel of Hour Vs Year",y=1.08,fontsize=30)
pie_heatmap(month_vs_year, vmin=-20,vmax=80,inner_r=0.2)
```

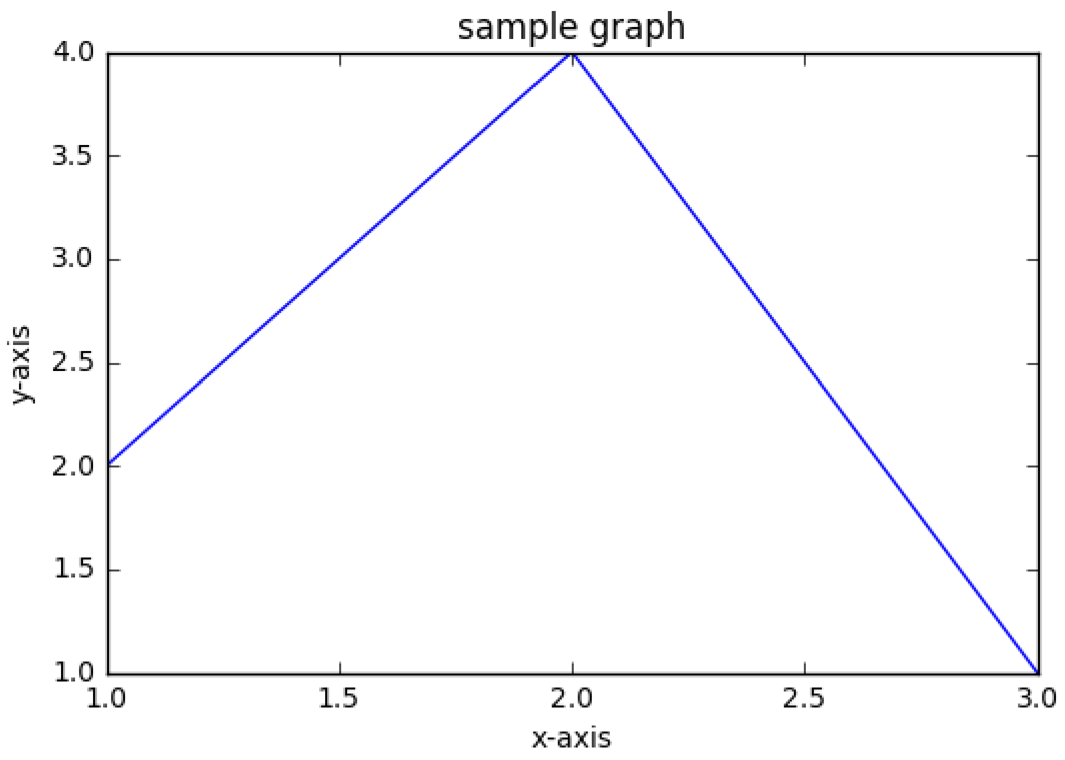
```
Comparison of the top 10 years in which the UFO was sighted vs each month:
```



Q141. Write a python program to draw the line using given axis values with the suitable label in the x-axis, y-axis,and a title.

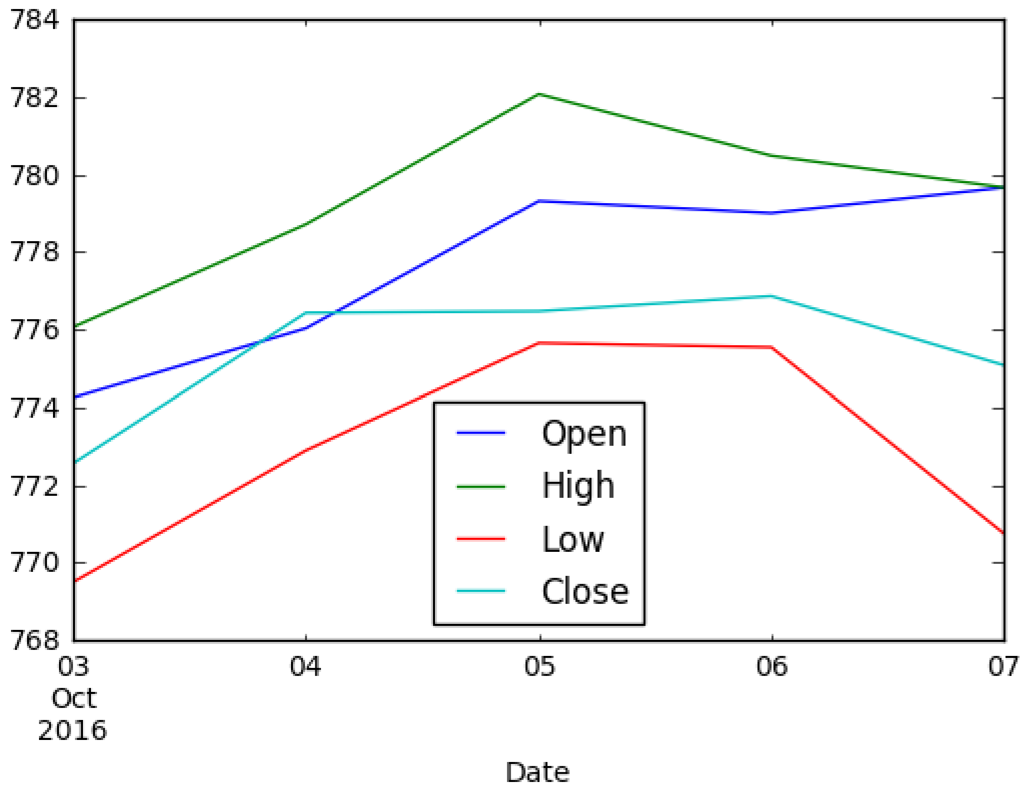
In [37]:
```python
import matplotlib.pyplot as plt
x = [1,2,3]
y = [2,4,1]
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('sample graph')
plt.plot(x,y)
```

Out[37]:
```
[<matplotlib.lines.Line2D at 0x13aa6e90>]
```



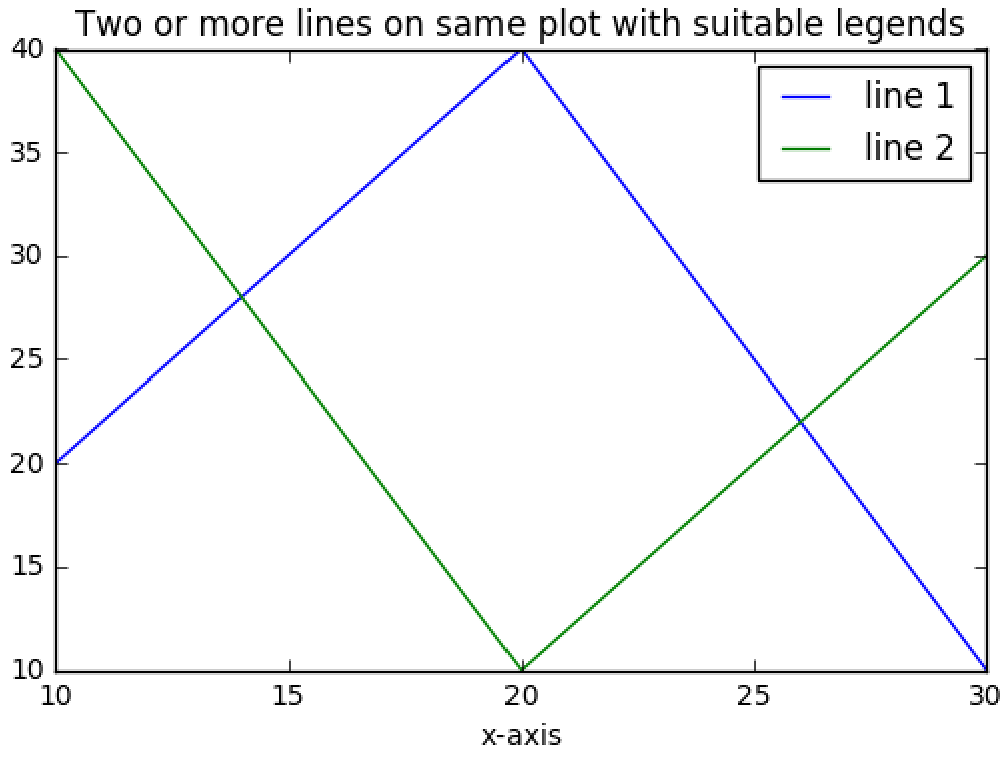
Q142. Write a python program to draw the line charts of the financial data of the Alphabet Inc., between October.

In [40]:
```
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('fdata.csv', sep=',', parse_dates=True, index_col=0)
df.plot()
plt.show()
```



Q143. Write a Python program to plot two or more lines on same plot with the suitable legends of eachline.

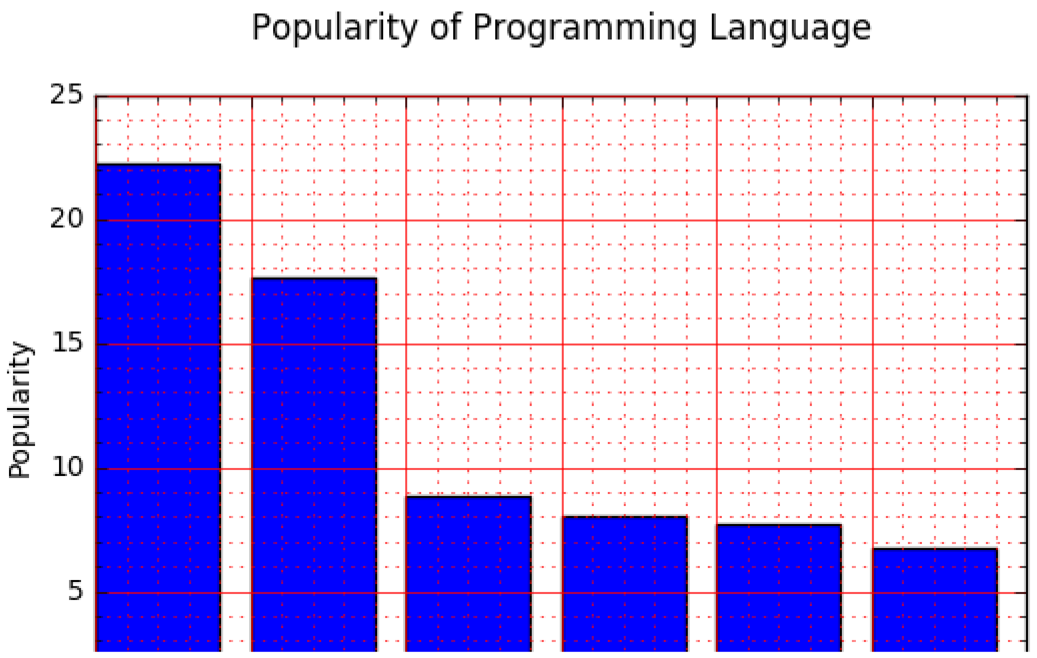
In [45]:
```
import matplotlib.pyplot as plt
x1 = [10,20,30]
y1 = [20,40,10]
plt.plot(x1, y1, label='line 1')

x2 = [10,20,30]
y2 = [40,10,30]
plt.plot(x2, y2, label = 'line 2')
plt.xlabel('x-axis')
plt.title('Two or more lines on same plot with suitable legends')
plt.legend()
plt.show()
```



Q144.Write a python programming to display a bar chart of the popularity of programming languages

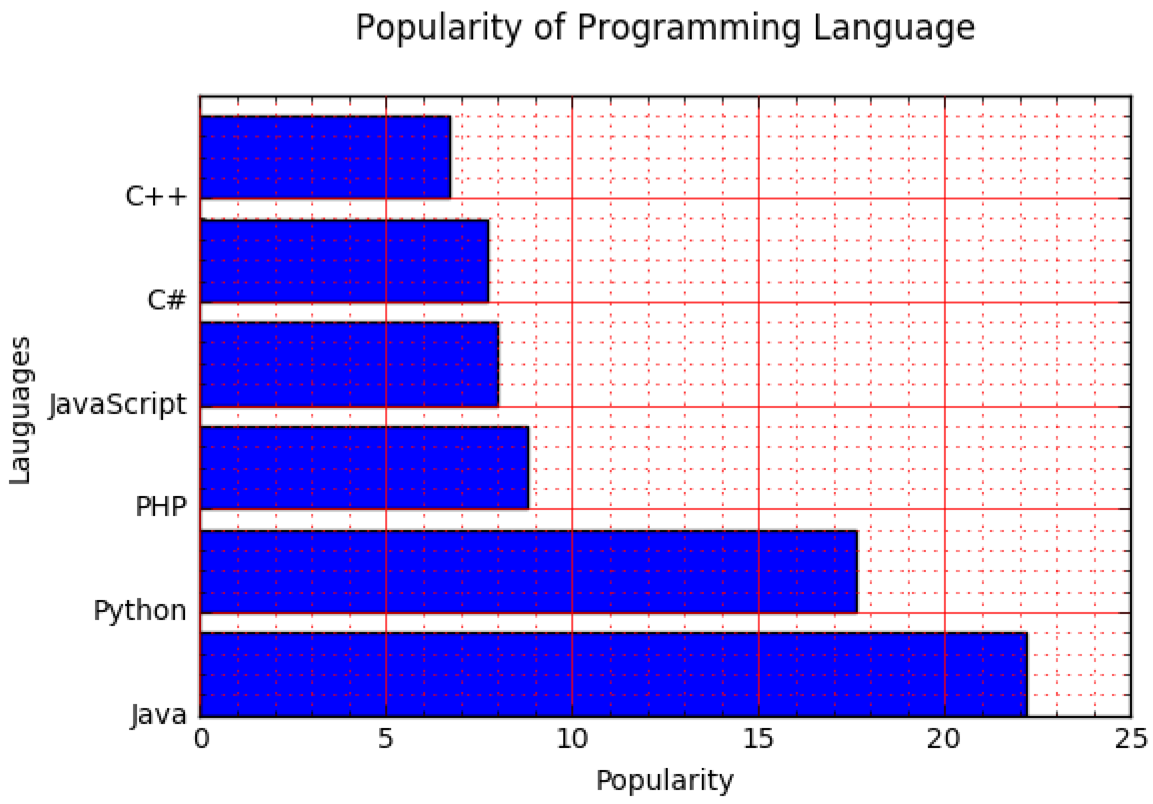
popularity of programming languages

In [57]:
```
import matplotlib.pyplot as plt
x = ['Java','Python','PHP',"JavaScript",'C#','C++']
popularity = [22.2,17.6,8.8,8,7.7,6.7]

x_pos = [i for i, _ in enumerate(x)]
plt.bar(x_pos, popularity, color='blue')
plt.xlabel('Lauguages')
plt.ylabel('Popularity')
plt.title('Popularity of Programming Language\n')
plt.xticks(x_pos, x)
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='red')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='red')
plt.show()
```



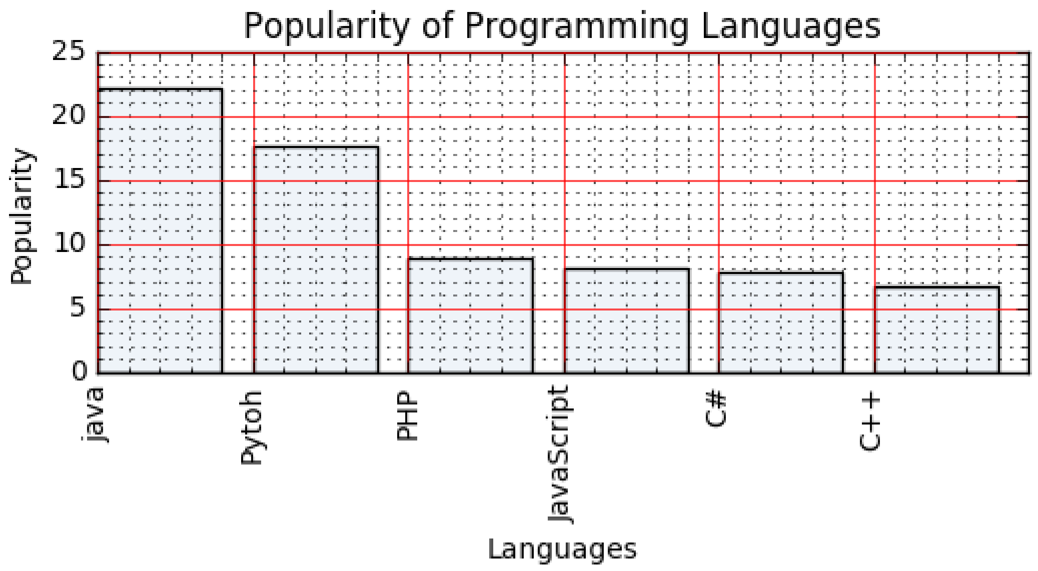
Q145.Write a python programming to display a horizontal bar chart of the popularity of programming languages.

In [62]:
```
import matplotlib.pyplot as plt
x = ['Java','Python','PHP',"JavaScript",'C#','C++']
popularity = [22.2,17.6,8.8,8,7.7,6.7]

x_pos = [i for i, _ in enumerate(x)]
plt.barh(x_pos, popularity, color='blue')
plt.ylabel('Lauguages')
plt.xlabel('Popularity')
plt.title('Popularity of Programming Language\n')
plt.yticks(x_pos, x)
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='red')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='red')
plt.show()
```



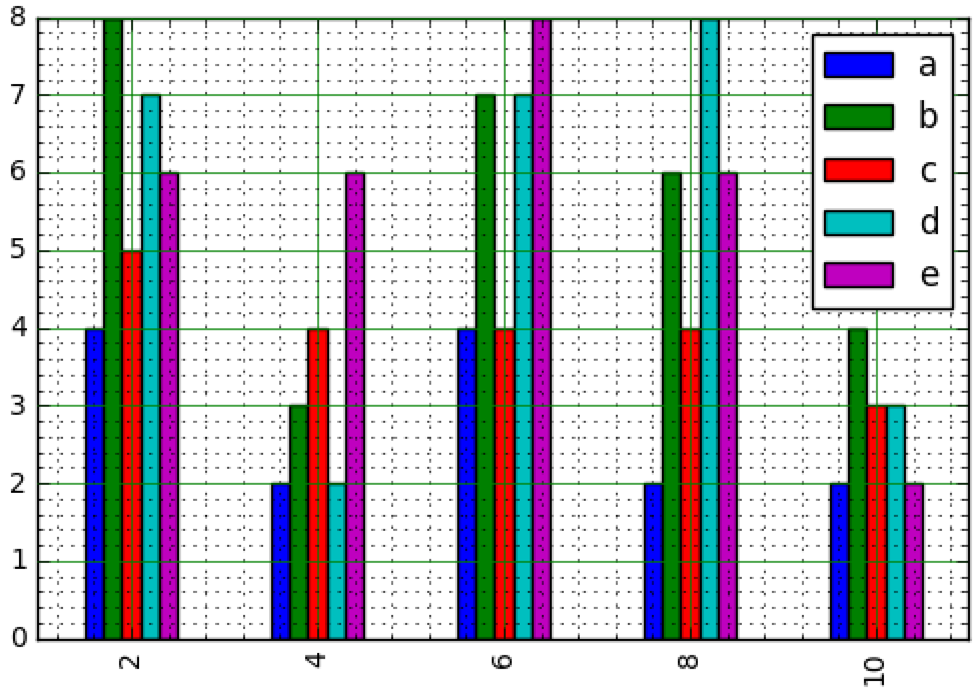
Q146.Write a python programming to display a bar chart of the Popularity of programming languages. Increase bottom margin

In [66]:
```
import matplotlib.pyplot as plt
x=['java','Pytoh','PHP','JavaScript','C#','C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
x_pos = [i for i, _ in enumerate(x)]
plt.bar(x_pos, popularity, color=(0.4, 0.6, 0.8, 0.10))
plt.xlabel('Languages')
plt.ylabel('Popularity')
plt.title('Popularity of Programming Languages')
plt.xticks(x_pos, x, rotation=90)
plt.subplots_adjust(bottom=0.4, top=0.8)
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='red')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```



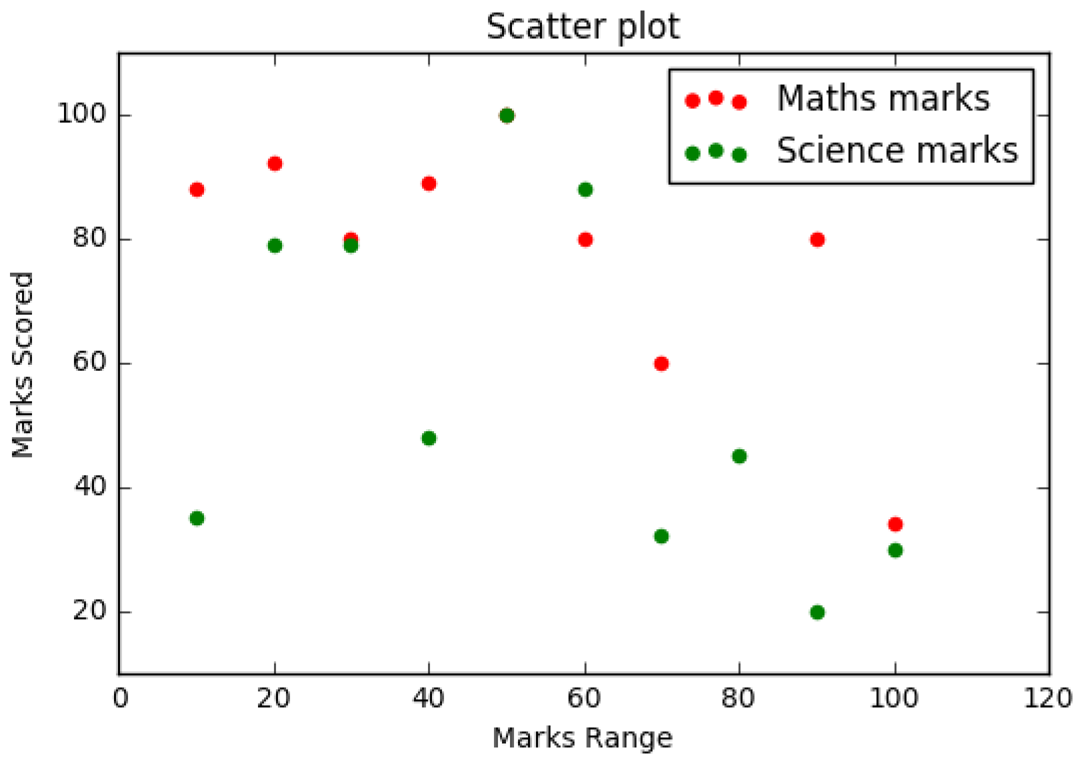
Q147.Write a python program to create the bar plot from a DataFrame.

In [70]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

a=np.array([[4,8,5,7,6],[2,3,4,2,6],[4,7,4,7,8],[2,6,4,8,6],[2,4,3,3,2]])
df=pd.DataFrame(a, columns=['a','b','c','d','e'], index=[2,4,6,8,10])

df.plot(kind='bar')
plt.minorticks_on()
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
plt.show()
```



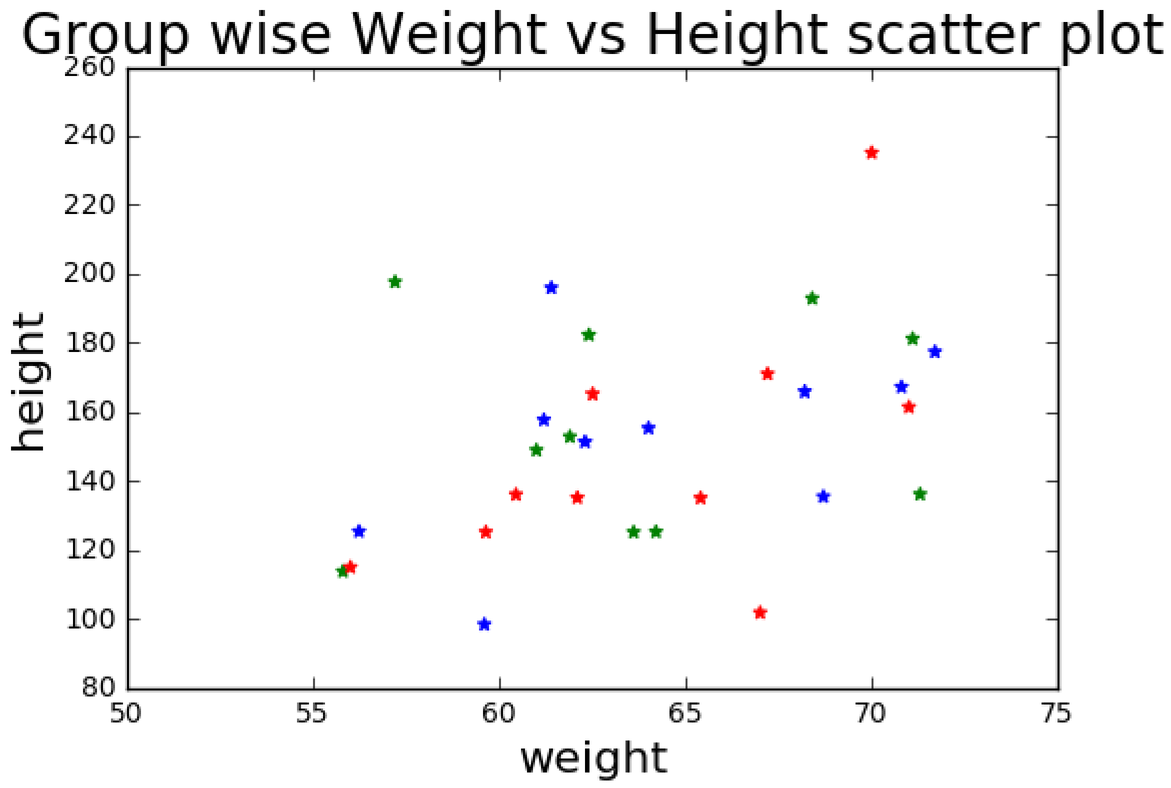
Q148.Write a python program to draw thescatter plot comparing two subject marks of Mathematics and Science. Use marks of 10 students

In [72]:
```
import matplotlib.pyplot as plt
import pandas as pd

math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
science_marks = [35,79,79,48,100,88,32,45,20,30]
marks_range=[10,20,30,40,50,60,70,80,90,100]
plt.scatter(marks_range, math_marks, label='Maths marks', color='r')
plt.scatter(marks_range, science_marks, label='Science marks', color='g')
plt.title('Scatter plot')
plt.xlabel('Marks Range')
plt.ylabel('Marks Scored')
plt.legend()
plt.show()
```



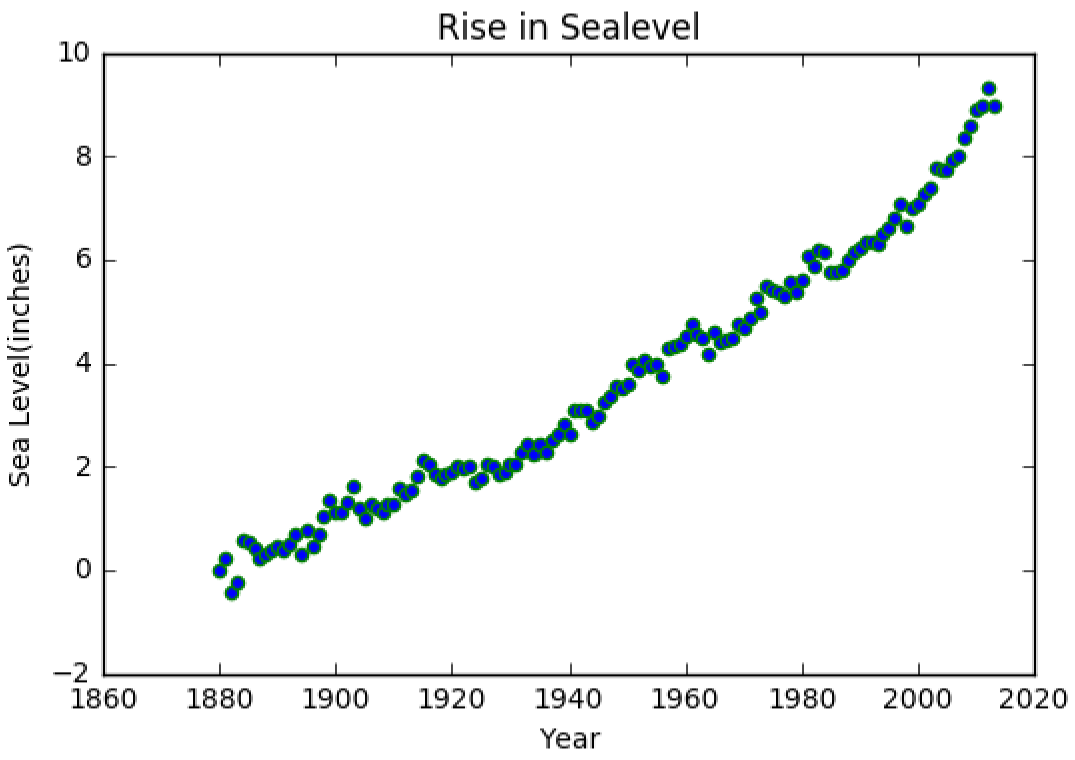
Q149.Write a python program to draw thescatter plot for three different groups comparing weights and heights.

In [76]:
```
import matplotlib.pyplot as plt
import numpy as np
w1=[67,57.2,59.6,59.64,55.8,61.2,60.45,61,56.23,56]
h1=[101.7,197.6,98.3,125.1,113.7,157.7,136,148.9,125.3,114.9]
w2=[61.9,64,62.1,64.2,62.3,65.4,62.4,61.4,62.5,63.6]
h2=[152.8,155.3,135.1,125.2,151.3,135,182.2,195.9,165.1,125.1]
w3=[68.2,67.2,68.4,68.7,71,71.3,70.8,70,71.1,71.7]
h3=[165.8,170.9,192.8,135.4,161.4,136.1,167.1,235.1,181.1,177.3]

weight = np.concatenate([w1, w2, w3])
height = np.concatenate([h1, h2, h3])
plt.scatter(weight, height, marker='*', color=['red','green','blue'])
plt.xlabel('weight', fontsize=16)
plt.ylabel('height', fontsize=16)
plt.title('Group wise Weight vs Height scatter plot', fontsize=20)
plt.show()
```



Q150.Write a python program to draw a scatter plot to find sea-level rise in past 100 years.

In [83]:
```
import matplotlib.pyplot as plt
import pandas as pd
data = pd.read_csv('data.csv')
year = data['year']
sea_levels = data['CSIRO_sea_level']
plt.scatter(year, sea_levels, edgecolors='g')
plt.xlabel('Year')
plt.ylabel('Sea Level(inches)')
plt.title('Rise in Sealevel')
plt.show()
```



In [ ]:
```
```