In [2]:

```
#1.Python Program to count the number of blank spaces in a text file.

fname = input("Enter file name: ")
k = 0

with open(fname, 'r') as f:
    for line in f:
        words = line.split()
        for i in words:
            for letter in i:
                if(letter.isspace):
                    k=k+1
print("no. of blank spaces:")
print(k)
```

```
Enter file name: out.txt
no. of blank spaces:
16234
```

In [3]:

```
#2.Write python program to get a list of tuples of Rollno,Name for 5 students through
#keyboard and sort them by Name wise descending order

import operator
n=int(input("Enter no of records"))
d={}
for i in range(1,n+1):
    name= input("Enter name %d"%(i))
    mark=int(input("Enter mark %d"%(i)))
    d[name]=mark
    print(d)
sorted_a= sorted(d.items(), key=operator.itemgetter(0),reverse=True)
print(sorted_a)
```

```
Enter no of records5
Enter name 1vinod
Enter mark 110
{'vinod': 10}
Enter name 2kiran
Enter mark 220
{'vinod': 10, 'kiran': 20}
Enter name 3baste
Enter mark 320
{'vinod': 10, 'kiran': 20, 'baste': 20}
Enter name 4im karan
Enter mark 425
{'vinod': 10, 'kiran': 20, 'baste': 20, 'im karan': 25}
Enter name 5im arjun
Enter mark 515
{'vinod': 10, 'kiran': 20, 'baste': 20, 'im karan': 25, 'im arjun': 15}
[('vinod', 10), ('kiran', 20), ('im karan', 25), ('im arjun', 15), ('baste',
20)]
```

In [1]:

```python
#3.Write python program to get a list of tuples of Rollno,Name for 5 students through
#keyboard and sort them by Name wise ascending order

import operator
n=int(input("Enter no of records"))
d={}
for i in range(1,n+1):
    name= input("Enter name %d"%(i))
    mark=int(input("Enter rollno %d"%(i)))
    d[name]=mark
    print(d)
sorted_a= sorted(d.items(), key=operator.itemgetter(1),reverse=False)
print(sorted_a)

```

```
Enter no of records5
Enter name 1vinod
Enter rollno 11
{'vinod': 1}
Enter name 2apoorva
Enter rollno 22
{'vinod': 1, 'apoorva': 2}
Enter name 3bhanish
Enter rollno 33
{'vinod': 1, 'apoorva': 2, 'bhanish': 3}
Enter name 4charan
Enter rollno 44
{'vinod': 1, 'apoorva': 2, 'bhanish': 3, 'charan': 4}
Enter name 5druvil
Enter rollno 55
{'vinod': 1, 'apoorva': 2, 'bhanish': 3, 'charan': 4, 'druvil': 5}
[('vinod', 1), ('apoorva', 2), ('bhanish', 3), ('charan', 4), ('druvil', 5)]
```

In [2]:

```python
#4.Python Program to append the contents of one file to another file by getting the bot
#names through keyboard

name1 = input("Enter file to be read from: ")
name2 = input("Enter file to be appended to: ")
fin = open(name1, "r")
data2 = fin.read()
fin.close()
fout = open(name2, "a")
fout.write(data2)
fout.close()
```

```
Enter file to be read from: Untitled2.ipynb
Enter file to be appended to: Untitled3.ipynb
```

In [3]:

```python
#5.Python program to convert a 3 digit number into words

def convert_to_words(num):
    l = len(num);
    if (l == 0):
        print("empty string");
        return;

    if (l > 4):
        print("Length more than 4 is not supported");
        return;
    single_digits = ["zero", "one", "two", "three",
                     "four", "five", "six", "seven",
                     "eight", "nine"];
    two_digits = ["", "ten", "eleven", "twelve",
                  "thirteen", "fourteen", "fifteen",
                  "sixteen", "seventeen", "eighteen",
                  "nineteen"];
    tens_multiple = ["", "", "twenty", "thirty", "forty",
                     "fifty", "sixty", "seventy", "eighty",
                     "ninety"];
    tens_power = ["hundred", "thousand"];
    print(num, ":", end = " ");
    if (l == 1):
        print(single_digits[ord(num[0]) - '0']);
        return;
    x = 0;
    while (x < len(num)):
        if (l >= 3):
            if (ord(num[x]) - 48 != 0):
                print(single_digits[ord(num[x]) - 48],
                                            end = " ");
                print(tens_power[l - 3], end = " ");

            l -= 1;
        else:
            if (ord(num[x]) - 48 == 1):
                sum = (ord(num[x]) - 48 +
                       ord(num[x]) - 48);
                print(two_digits[sum]);
                return;
            elif (ord(num[x]) - 48 == 2 and
                  ord(num[x + 1]) - 48 == 0):
                print("twenty");
                return;
            else:
                i = ord(num[x]) - 48;
                if(i > 0):
                    print(tens_multiple[i], end = " ");
                else:
                    print("", end = "");
                x += 1;
                if(ord(num[x]) - 48 != 0):
                    print(single_digits[ord(num[x]) - 48]);
        x += 1;
convert_to_words("001");
convert_to_words("100");
```

```
001 : one
100 : one hundred
```

In [8]:

```
#6.Print perfect squares and divisible by 5 between 500 and 1000 (both inclusive)
#using list comprehension

print (x for x in  range(500,1000) if (x**0.5)%5 == 0 )
```

```
<generator object <genexpr> at 0x0000022845C5B430>
```

In [9]:

```
#7.Print lists of odd,even and multiples of 5 numbers from 1 to 1000 using list
#comprehension

print ([x for x in range (1001)if x%2==0 and x%5==0])
print ([x for x in range (1001)if x%2==1 and x%5==0])
```

```
[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 1
70, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 32
0, 330, 340, 350, 360, 370, 380, 390, 400, 410, 420, 430, 440, 450, 460, 47
0, 480, 490, 500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600, 610, 62
0, 630, 640, 650, 660, 670, 680, 690, 700, 710, 720, 730, 740, 750, 760, 77
0, 780, 790, 800, 810, 820, 830, 840, 850, 860, 870, 880, 890, 900, 910, 92
0, 930, 940, 950, 960, 970, 980, 990, 1000]
[5, 15, 25, 35, 45, 55, 65, 75, 85, 95, 105, 115, 125, 135, 145, 155, 165, 1
75, 185, 195, 205, 215, 225, 235, 245, 255, 265, 275, 285, 295, 305, 315, 32
5, 335, 345, 355, 365, 375, 385, 395, 405, 415, 425, 435, 445, 455, 465, 47
5, 485, 495, 505, 515, 525, 535, 545, 555, 565, 575, 585, 595, 605, 615, 62
5, 635, 645, 655, 665, 675, 685, 695, 705, 715, 725, 735, 745, 755, 765, 77
5, 785, 795, 805, 815, 825, 835, 845, 855, 865, 875, 885, 895, 905, 915, 92
5, 935, 945, 955, 965, 975, 985, 995]
```

In [ ]:

```

```