NAME :D.HARIHARASUDHAN
DEP.NO :205229112
ASSIGNMENT :PYTHON

Question3.write a function lastN(lst,n)that takes a list of integers and n and returns n largest numbers.

Source code:

```
def lastN(lst ,N):
    final_list = []
    for i in range(0,N):
        max1 = 0
        for j in range(len(lst)):
            if lst[j]>max1:
                max1=lst[j];
        lst.remove(max1);
        final_list.append(max1)
    print(final_list)
```

Output:

```
lst = [20, 40, 60, 70, 80, 90]
N = 3
lastN(lst,N)

[70, 80, 90]
```

Question4.

Source code:

```
def front_x(words):
    xlist=[]
    alist=[]

    for word in words:
        if word.startswith('x'):
            xlist.append(word)
    else:
        alist.append(word)
    return sorted(xlist)+sorted(alist)
```

Output:

```
words = [’ccc’,’bbb’,’aaa’,’xcc’,’xaa’]
front_x(words)

[’xaa’, ’xcc’, ’xaa’]
```

Question5.

Source code:

```
def sort_last(n,tuples):
        return sorted(tuples, key=last)
print(sort_last(-1,[(1,7), (1,3), (3,4,5), (2,2)]))
```

Output:

```
([2,2], [1,3], [3,4,5], [1,7])
```

Question6.

Source code:

```
def first_sort(tlist):
        print("sorted list using key first")
        tlist.sort()
        print(tlist)
def middle_sort(tlist):
        print("\nsorted list by using key middle")
        tlist=sorted(tlist,key=lambda mid:mid[1])
        print(tlist)
def first(tlist):
        print("\nfirst element")
        for j in tlist:
                a,b,c=j
                print(a)
def middle(tlist):
        print("\nmiddle element")
        for j in tlist:
                a,b,c=j
                print(b)

tlist=[(1,2,3),(9,5,7),(16,7,5),(35,32,9),(20,8,65)]
first_sort(tlist)
middle_sort(tlist)
first(tlist)
middle(tlist)
```

Output:

```
sorted list using key first
[(1, 2, 3), (9, 5, 7), (16, 7, 5), (20, 8, 65), (35, 32, 9)]

sorted list by using key middle
[(1, 2, 3), (9, 5, 7), (16, 7, 5), (20, 8, 65), (35, 32, 9)]

first element
1
9
16
20
35

middle element
2
5
7
8
32
```

Question7.

Source code:

```
def remove_adjacent(nums):

    result = []
    for num in nums:
        if len(result) == 0 or num != result[-1]:
            result.append(num)
    return result
```

Output:

```
nums = [1, 2, 2, 2, 3]

[1, 2, 3]
```

The manual practical excercise photos are given below

Question3. Write a function **lastN(lst, n)** that takes a list of integers and **n** and returns **n** largest numbers.

```
How many numbers you want to enter?: 6
Enter a number: 12
Enter a number: 32
Enter a number: 10
Enter a number: 9
Enter a number: 52
Enter a number: 45
How many largest numbers you want to find?: 3
Largest numbers are: 52, 45, 32
```

```
def lastN(lst, N)
    final_list = []
    for i in range(0, N):
        max1 = 0
        for j in range(len(list1)):
            if list1[j]> max1:
                max1 = list1[j];
        list1.remove(max1);
        final_list.append(max1)
    Print(final_list)
```

```
In [ ]: lst = [10,20,30,40,50,60]
        N = 2
        lastN(lst,N)
```

Question4. Given a list of strings, return a list with the strings in sorted order, except group all the strings that begin with 'x' first. Hint: this can be done by making 2 lists and sorting each of them before combining them.

Test Cases:

1. Input: ['mix', 'xyz', 'apple', 'xanadu', 'aardvark']
   Output: ['xanadu', 'xyz', 'aardvark', 'apple', 'mix']
2. Input: ['ccc','bbb','aaa','xcc','xaa']
   Output: ['xaa','xcc','aaa','bbb','ccc']
3. Input: ['bbb','ccc','axx','xzz','xaa']
   Output: ['xaa','xzz','axx','bbb','ccc']

```
def front-x(words):
    x list =[]
    alist =[]

    for word in words:
        if word.startswith('x'):
            x list.append(word)
        else:
            alist.append(word)
    return sorted(x list) + sorted(alist)

words = ['mix', 'xyz', 'apple', 'xanadu', 'aardvark']
    front_x(words)
```

Question5. Develop a function **sort_last()**. Given a list of non-empty tuples, return a list sorted in increasing order by the last element in each tuple. Hint: use a custom key= function to extract the last element form each tuple.

Test Cases:

1. Input: [(1, 7), (1, 3), (3, 4, 5), (2, 2)]
   Output: [(2, 2), (1, 3), (3, 4, 5), (1, 7)]
2. Input: [(1,3),(3,2),(2,1)]
   Output: [(2,1),(3,2),(1,3)]
3. Input: [(2,3),(1,2),(3,1)]
   Output: [(3,1),(1,2),(2,3)]

```
def Sort-last(n, tuples):
    return Sorted(tuples, key=last)
print(sort-last(-1,[(1,7),(1,3),(3,4,5),(2,2)]))
```

Question6. Other String Functions

a) Define a function **first()** that receives a tuple and returns its first element
b) Define a function **sort_first()** that receives a list of tuples and returns the sorted
c) Print lists in sorted order
d) Define a function **middle()** that receives a a tuple and returns its middle element
e) Define a functino **sort_middle**() that receives a list of tuples and returns it sorted using the key middle
f) Print the list [(1,2,3), (2,1,4), (10,7,15), (20,4,50), (30, 6, 40)] in sorted order. Output should be: [(2, 1, 4), (1, 2, 3), (20, 4, 50), (30, 6, 40), (10, 7, 15)]

```
def last(n): return n[-1]
def Sort_list_last(tuples):
    return Sorted(tuples, key=last)
Print
```

```
def first-Sort(tlist):
    Print("Sorted list using key first")
    tlist.Sort()
    Print(tlist)
def middle-Sort(tlist):
    Print("\nSorted list by using key middle")
    tlist=sorted(tlist,key=lambda mid:mid[1])
    Print(tlist)
```

```
def first(tlist):
    Print("\nfirst element")
    for i in tlist:
        a,b,c=i
        Print(a)
def middle(tlist):
    Print("\nmiddle element")
    for i in tlist:
        a,b,c=i
        Print(b)
```

```
tlist=[(1,2,3),(9,5,7),(16,7,5),(35,32,7),
                    (20,8,65)]
first_sort(tlist)
middle_sort(tlist)
first(tlist)
middle(tlist)
```

Question7. Develop a function **remove_adjacent().** Given a list of numbers, return a list where all adjacent same elements have been reduced to a single element. You may create a new list or modify the passed in list.

Test Cases:

1. Input: [1, 2, 2, 3] and output: [1, 2, 3]
2. Input: [2, 2, 3, 3, 3] and output: [2, 3]
3. Input: [ ]. Output: [ ].
4. Input: [2,5,5,6,6,7]
   Output: [2,5,6,7]
5. Input: [6,7,7,8,9,9]
   Output: [6,7,8,9]

```
def remove_adjacent (nums):
        result =[]
    for num in nums:
        if len(result) == 0 or num != result[-1]:
                result.append(num)
        return result

getting result :
        nums = [1,2,2,3]
         remove_adjacent( )
```