**17/2/2019**

**16/02 Assignments Revision**

**Trace Bubble Sort with input 7,11,9,31,6,13**

**find attached image**

**trace with flag .**

**#WAP to accept list from user and print it in reverse order element by element using recursion**

#WAP to accept list from user and print it in reverse order element by element using recursion

def reverseListRecursion(li):

if len(li)==0:

return 0

reverseListRecursion(li[1:])

print(li[0])

def main():

li=input("Enter list")

reverseListRecursion(li)

if \_\_name\_\_=='\_\_main\_\_':

main()

#WAP to accept string from user and print it in reverse order using recursion

def reverseContainerRecursion(li):

if len(li)==0:

return li

x=reverseContainerRecursion(li[1:])

x.append(li[0])

return x

def main():

li=input("Enter List")

print(reverseContainerRecursion(li))

if \_\_name\_\_=='\_\_main\_\_':

main()

**#Combined function**

def ReverseContainer(li):

if len(li)==0:

if type(li)==str:

return li

return list()

x=ReverseContainer(li[1:])

if type(x) == str:

return x+li[0]

x.append(li[0])

return x

**#wap to accpet two lists from user, sort them and write function to merge the two sorted list element by element preserving the sort order.**

def mergeSortedList(l1,l2):

i=j=0

l3=[]

while(i<len(l1) and j<len(l2)):

if l1[i]<l2[j]:

l3.append(l1[i])

i+=1

else:

l3.append(l2[j])

j+=1

if(i<len(l1)):

l3.extend(l1[i:])

if(j<len(l2)):

l3.extend(l2[j:])

return l3

def main():

l1=input("Enter first list: ")

l2=input("Enter second list: ")

print(mergeSortedList(l1,l2))

if \_\_name\_\_=='\_\_main\_\_':

main()

**#WAP to accept two lists from user and find union of them, intersection of them, symmetric difference(without common)**

**l1[1,7,11,15,21,50]**

**l2[2,7,15,17]**

**union=[1,2,7,11,15,17,21,50]**

**intersection=[7,15]**

**symmetric diff=[1,2,11,17,21,50]**

**#WAP to check if isSubset of , isDisjoint, isSuperset**