Python DataStructures

- Lists
- Tuples
- Dictionaries

Basic problem set on Datastructures

Advanced problems set

packages and modules in python

In [16]:

```
# list
li=[123,978,333]
li
li[1]
li[1:]
li
li.append(33)#append
li
li.insert(0,1)#insert
li
li.sort()#sorting
li
li.pop()#poping element
li
li.pop(1)
li
l2=[1,2,3]
li.extend(l2)#extend
li
li.append([100,200])
li
li.extend([2,2,2,2])
li
```

Out[16]:

```
[1, 123, 333, 1, 2, 3, [100, 200], 2, 2, 2, 2]
```

In [19]:

```
l1=[1,2,4,3,55,33]
sum(l1)
max(l1)
min(l1)
len(l1)
```

Out[19]:

```
6
```

In [21]:

```
# avarage of elements
l=[]
for i in range(5):
    a=int(input("enter no"))
    l.append(a)
s=sum(l)
avg=s//len(l)
print(avg)
```

```
enter no1
enter no2
enter no3
enter no4
enter no5
3
```

In [28]:

```
# average of all alternate no
l=[]
aa=[]
for i in range(5):
    a=int(input("enter no"))
    l.append(a)
b=sum(l[::2])/len(l[1::2])
print(b)
```

```
enter no1
enter no2
enter no3
enter no4
enter no5
4.5
```

In [42]:

```
#second largest no
#    sort the data and select largest no
#    sort the data in reverse order and select
#     remove max value and print last no
def secondlargest(li):
    li.sort()
    print( len(li)-1 )
secondlargest([100,1,3,2])
```

```
3
```

In [49]:

```
#generical largest no
def genericallargest(li,n):
    li.sort(reverse=True)
    print(li[n-1])
genericallargest([1,100,3,33,2,4],5)
```

```
2
```

In [45]:

```
li
```

Out[45]:

```
[1, 123, 333, 1, 2, 3, [100, 200], 2, 2, 2, 2]
```

In [50]:

```
li=[1,2,3,55,44,33,22]
li
```

Out[50]:

```
[1, 2, 3, 55, 44, 33, 22]
```

In [51]:

```
li
```

Out[51]:

```
[1, 2, 3, 55, 44, 33, 22]
```

In [11]:

```
#search data in list
#search key in tbe list and return index valyue
def linearsearch(li,key):
    for i in range(0,len(li)):
        if key==li[i]:
            print(li.index(key))
        else:
            print("not in key")
linearsearch(li,55)

#def linear2([1,44,55],key):
```

```
not in key
not in key
not in key
3
not in key
not in key
not in key
```

In [8]:

```
li=[1, 2, 3, 55, 44, 33, 22]
```

In [68]:

```
def
n=int(input("enter no"))
li.index(n)
```

```
enter no55
```

Out[68]:

```
3
```

In [15]:

```
#count the string perticular string
def countletter(s,c):
    count=0
    for i in s:
        if i=="e":
            count=count+1
    return count
countletter("satheesh","e")
def countletter(s,c):
    ss=s.count(c)
    print(ss)
countletter("satheesh","e")
```

```
2
```

In [18]:

```
#function to find the number pf occurances of string in string
#"abcdab"..."ab"..>2
def substring(s,s1):
    c=0
    sub=len(s1)
    for i in range(len(s)):
        if s[i:i+sub]==s1:
            c=c+1
    return c
substring("aabbaabbaa","aa")
```

Out[18]:

```
3
```

In [19]:

```
s="1 2 3 4 5"
li=s.split()
nlist=[]
for i in li:
    nlist.append(int(i))
print(nlist)
```

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

In [23]:

```
n=int(input())
s=input()
s=s.split()
l=[]

for i in s:
    l.append(int(i))
print(l)
def closestzero(li):
    a=[]
    for i in li:
        if li[i]<0:
            a.append(li[i])
        print
```

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

In [ ]:

In [ ]: