



# **SNS COLLEGE OF ENGINEERING**

**Kurumbapalayam(Po), Coimbatore – 641 107**

**Accredited by NAAC-UGC with 'A' Grade**

**Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai**

**Department of Artificial Intelligence and  
Data Science**

**Course Name – Computational Thinking and  
Python Programming**

**I Year / I Semester**

**Unit 4-LISTS, TUPLES, DICTIONARIES**



## Advanced list processing:

### List Comprehension:

- √ List comprehensions provide a concise way to apply operations on a list.
- √ It creates a new list in which each element is the result of applying a given operation in a list.
- √ It consists of brackets containing an expression followed by a “for” clause, then a list.
- √ The list comprehension always returns a result list.

### Syntax

**list=[ expression for item in list if conditional ]**

List Comprehension	Output
<pre>&gt;&gt;&gt;L=[x**2 for x in range(0,5)] &gt;&gt;&gt;print(L)</pre>	[0, 1, 4, 9, 16]
<pre>&gt;&gt;&gt;[x for x in range(1,10) if x%2==0]</pre>	[2, 4, 6, 8]
<pre>&gt;&gt;&gt;[x for x in 'Python Programming' if x in ['a','e','i','o','u']]</pre>	['o', 'o', 'a', 'i']
<pre>&gt;&gt;&gt;mixed=[1,2,"a",3,4.2] &gt;&gt;&gt; [x**2 for x in mixed if type(x)==int]</pre>	[1, 4, 9]
<pre>&gt;&gt;&gt;[x+3 for x in [1,2,3]]</pre>	[4, 5, 6]
<pre>&gt;&gt;&gt; [x*x for x in range(5)]</pre>	[0, 1, 4, 9, 16]
<pre>&gt;&gt;&gt; num=[-1,2,-3,4,-5,6,-7] &gt;&gt;&gt; [x for x in num if x&gt;=0]</pre>	[2, 4, 6]
<pre>&gt;&gt;&gt; str=["this","is","an","example"] &gt;&gt;&gt; element=[word[0] for word in str] &gt;&gt;&gt; print(element)</pre>	['t', 'i', 'a', 'e']



### **Nested list:**

List inside another list is called nested list.

#### **Example:**

```
>>> a=[56,34,5,[34,57]]
```

```
>>> a[0]
```

```
56
```

```
>>> a[3]
```

```
[34, 57]
```

```
>>> a[3][0]
```

```
34
```

```
>>> a[3][1]
```

```
57
```

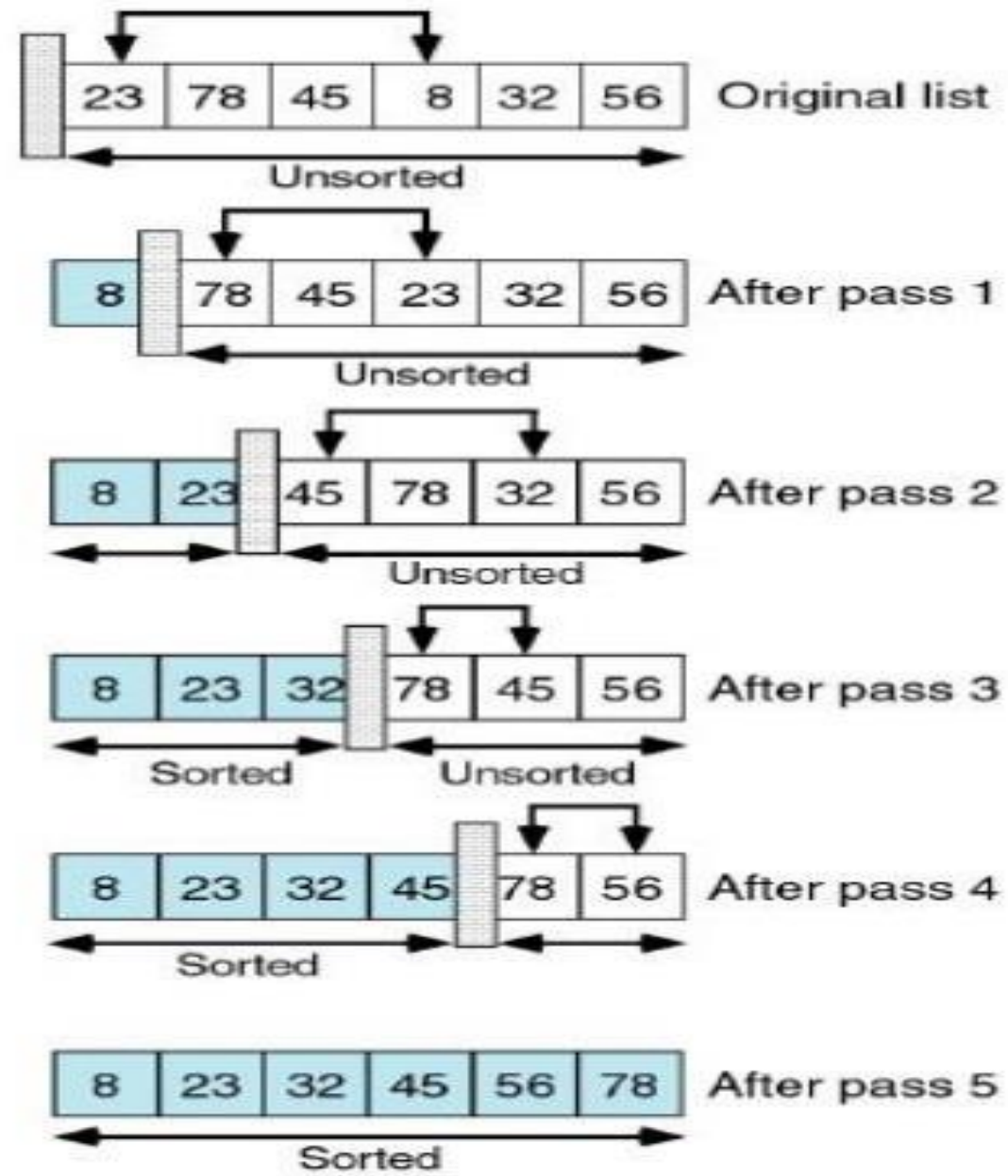
## Illustrative programs:

### Selection sort

```
a=input("Enter list:").split()
a=list(map(eval,a))
for i in range(0,len(a)):
    smallest = min(a[i:])
    sindex= a.index(smallest)
    a[i],a[sindex] = a[sindex],a[i]
print (a)
```

### Output

Enter list:23 78 45 8 32 56  
[8, 23, 32, 45, 56, 78]



## Insertion sort

```
a=input("enter a list:").split()
```

```
a=list(map(int,a))
```

```
for i in a:
```

```
    j = a.index(i)
```

```
    while j>0:
```

```
        if a[j-1] > a[j]:
```

```
            a[j-1],a[j] = a[j],a[j-1]
```

```
        else:
```

```
            break
```

```
        j = j-1
```

```
    print (a)
```

### output

```
enter a list: 8 5 7 1 9 3
```

```
[1,3,5,7,8,9]
```

8 5 7 1 9 3

5 8 7 1 9 3

5 7 8 1 9 3

1 5 7 8 9 3

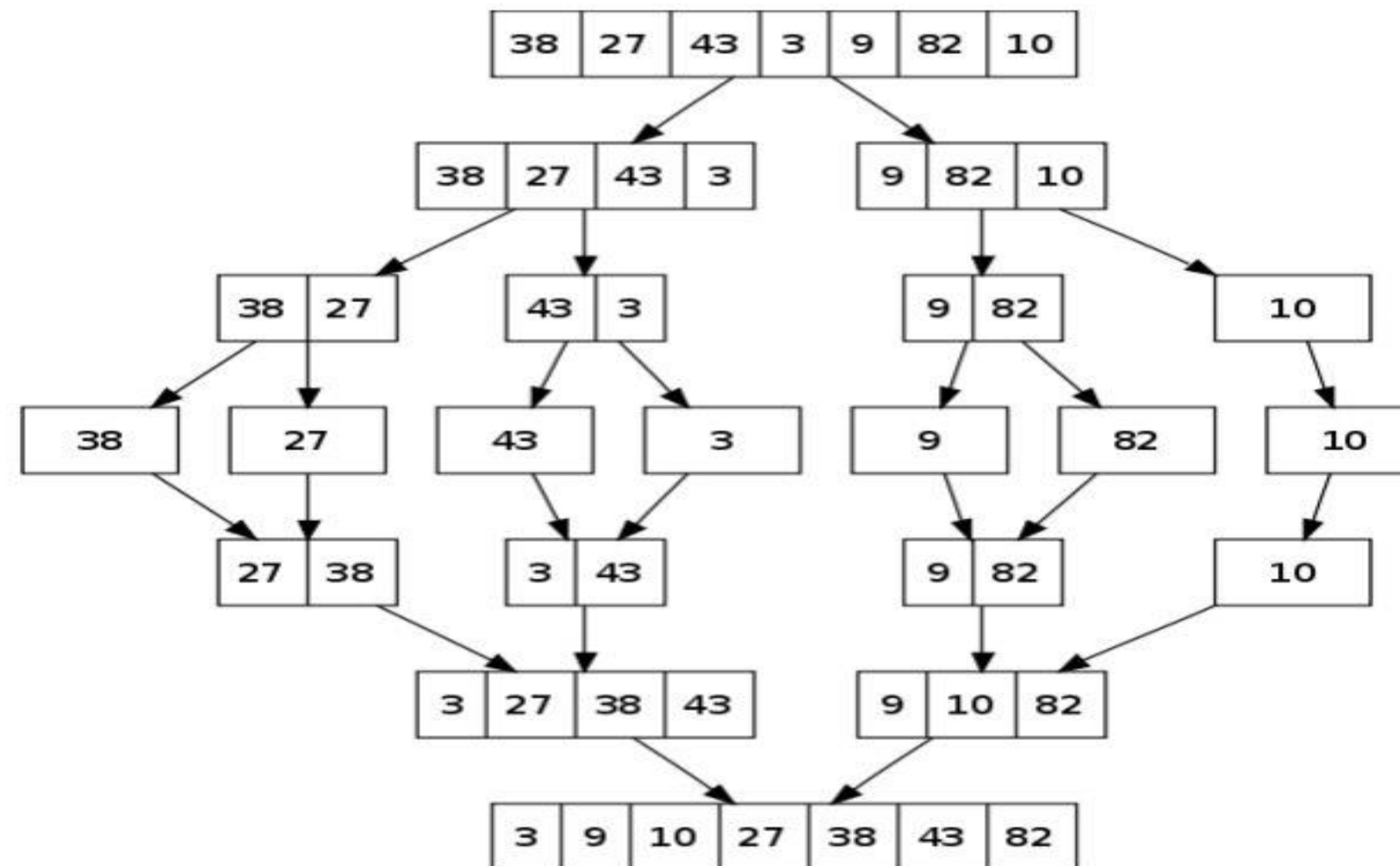
1 5 7 8 9 3

1 3 5 7 8 9

# Merge sort

```
def merge(a,b):
c = []
while len(a) != 0 and len(b) != 0:
if a[0] < b[0]:
c.append(a[0])
a.remove(a[0])
else:
c.append(b[0])
b.remove(b[0])
if len(a) == 0:
c=c+b
else:
c=c+a
return c
def divide(x):
if len(x) == 0 or len(x) == 1:
return x
```

```
else:
middle = len(x)//2
a = divide(x[:middle])
b = divide(x[middle:])
return merge(a,b)
x=[38,27,43,3,9,82,10]
c=divide(x)
print(c)
output
[3,9,10,27,38,43,82]
```





## #Python Program to find Student Grade

```
english = float(input(" Please enter English Marks: "))
math = float(input(" Please enter Math score: "))
computers = float(input(" Please enter Computer Marks: "))
physics = float(input(" Please enter Physics Marks: "))
chemistry = float(input(" Please enter Chemistry Marks: "))

total = english + math + computers + physics + chemistry
percentage = (total / 500) * 100

print("Total Marks = %.2f" %total)
print("Marks Percentage = %.2f" %percentage)
```

## Output:

```
Please enter English Marks: 60
Please enter Math score: 80
Please enter Computer Marks: 95
Please enter Physics Marks: 87
Please enter Chemistry Marks: 75
Total Marks = 397.00
Marks Percentage = 79.40
```

```
# Python program to create a histogram using the matplotlib  
function
```

```
# Importing the required libraries
```

```
import numpy as np
```

```
from matplotlib import pyplot as plt
```

```
# Creating a sample dataset
```

```
array = np.array([23, 56, 87, 87, 98,  
                 12, 76, 98, 34, 87,  
                 67, 23, 87, 56, 34,  
                 26, 85, 47, 35, 86,  
                 76, 45, 86, 34, 37])
```

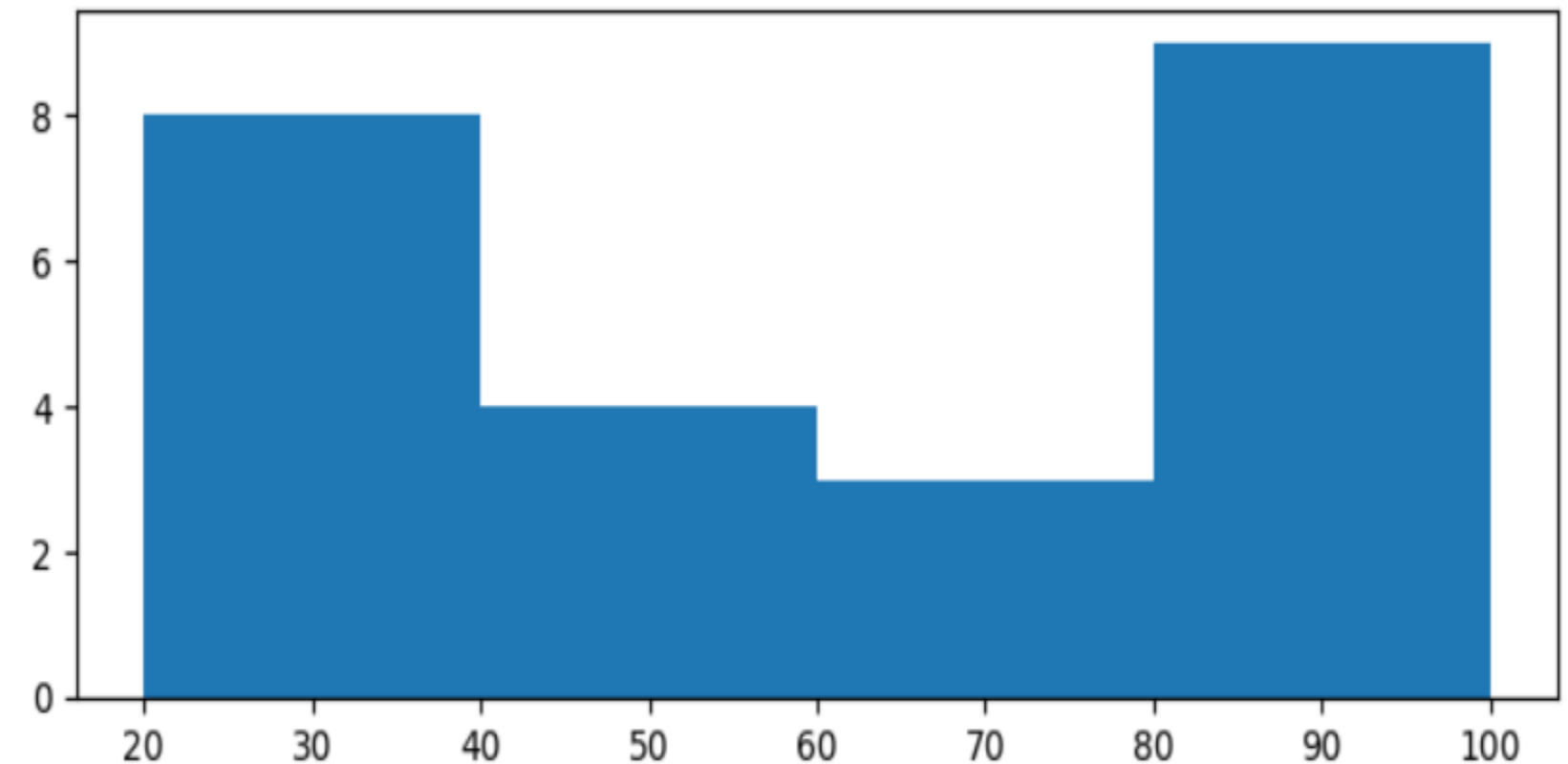
```
# Creating a histogram using .hist() function
```

```
figure, axis = plt.subplots(figsize = (8, 3))
```

```
axis.hist(array, bins = [20, 40, 60, 80, 100])
```

```
# Showing the plot
```

```
plt.show()
```







```
# This loop will go on until the budget is integer or float
while True:
```

```
    try:
```

```
        bg = float(input("Enter your budget : "))
```

```
        # if budget is integer or float it will be stored
```

```
        # temporarily in variable 's'
```

```
        s = bg
```

```
    except ValueError:
```

```
        print("PRINT NUMBER AS A AMOUNT")
```

```
        continue
```

```
    else:
```

```
        break
```

```
# dictionary to store product("name"), quantity("quant"),
```

```
# price("price") with empty list as their values
```

```
a = {"name":[], "quant":[], "price":[]}
```

```
# converting dictionary to list for further updation
```

```
b = list(a.values())
```

```
# variable na value of "name" from dictionary 'a'
```

```
na = b[0]
```

```
# variable qu value of "quant" from dictionary 'a'
```

```
qu = b[1]
```

```
# variable pr value of "price" from dictionary 'a'
```

```
pr = b[2]
```

```
# This loop terminates when user select 2.EXIT option when asked
```

```
# in try it will ask user for an option as an integer (1 or 2)
```

```
# if correct then proceed else continue asking options
```

```
while True:
```

```
    try:
```

```
        ch = int(input("1.ADD\n2.EXIT\nEnter your choice : "))
```

```
    except ValueError:
```

```
        print("\nERROR: Choose only digits from the given option")
```

```
        continue
```

```
    else:
```

```
        # check the budget is greater than zero and option selected
```

```
        # by user is 1 i.e. to add an item
```

```
        if ch == 1 and s>0:
```



```
# input products name
```

```
pn = input("Enter product name : ")
```

```
# input quantity of product
```

```
q = input("Enter quantity : ")
```

```
# input price of the product
```

```
p = float(input("Enter price of the product : "))
```

```
if p>s:
```

```
    # checks if price is less than budget
```

```
    print("\nCAN, T BUT THE PRODUCT")
```

```
    continue
```

```
else:
```

```
    # checks if product name already in list
```

```
    if pn in na:
```

```
        # find the index of that product
```

```
        ind = na.index(pn)
```

```
# remove quantity from "quant" index of the product
```

```
    qu.remove(qu[ind])
```

```
# remove price from "price" index of the product
```

```
    pr.remove(pr[ind])
```

```
# insert new value given by user earlier
```

```
    qu.insert(ind, q)
```

```
# insert new value given by user earlier
```

```
pr.insert(ind, p)
```

```
# subtracting the price from the budget and assign
```

```
# it to 's' sum(pr) is because pr = [100, 200] if
```

```
# budget is 500 then s = bg-sum(pr) = 200
```

```
# after updating for same product at index 0 let
```

```
# pr = [200, 200] so s = 100
```

```
s = bg-sum(pr)
```

```
print("\namount left", s)
```

```
else:
```

```
    # append value of in "name", "quantity", "price"
```

```
    na.append(pn)
```

```
# as na = b[0] it will append all the value in the
```

```
# list eg: "name":["rice"]
```

```
qu.append(q)
```

```
# same for quantity and price
```

```
pr.append(p)
```

```
# after appending new value the sum in price
```

```
# as to be calculated
```

```
s = bg-sum(pr)
```

```
print("\namount left", s)
```



```
# if budget goes zero print "NO BUDGET"
elif s<= 0:
    print("\nNO BUDGET")
else:
    break

# will print amount left in variable 's'
print("\nAmount left : Rs.", s)

# if the amount left equals to any amount in price list
if s in pr:
    # then printing the name of the product which can buy
    print("\nAmount left can buy you a", na[pr.index(s)])

print("\n\n\nGROCERY LIST")

# print final grocery list
for i in range(len(na)):
    print(na[i], qu[i], pr[i])
```

## Output:

```
User GO with following question
Enter Your budget : 500
1.Add an item
2.Exit
Enter your choice : 1

Enter product : corn flour
Enter quantity : 1.5 kg
Enter Price : 100

Amount left : 400

1.Add an item
2.Exit
Enter your choice : 1

Enter product : wheat
Enter quantity : 2 kg
Enter Price : 100
```



Amount left : 300

1.Add an item

2.Exit

Enter your choice : 1

Enter product : corn flour

Enter quantity : 2 kg

Enter Price : 250

Amount left : 150

1.Add an item

2.Exit

Enter your choice : 1

Enter product : rice

Enter quantity : 5 kg

Enter Price : 300

Can't Buy the product ###(because budget left is 150)

1.Add an item

2.Exit

Enter your choice : 1

Enter product : xyz

Enter quantity : 1 kg

Enter Price : 50

Amount left : 100

1.Add an item

2.Exit

Enter your choice : 2

Amount left can buy you wheat

GROCERY LIST is:

Product name	Quantity	Price
corn flour	2 kg	250
wheat	2 kg	100

