Unit II: Computational Thinking and Programming - I

● Lists: introduction, indexing, list operations (concatenation, repetition, membership and slicing), traversing a list using loops,built-in functions/methods–len(), list(), append(), extend(), insert(), count(), index(), remove(), pop(), reverse(), sort(), sorted(), min(), max(), sum();nested lists, suggested programs: finding the maximum, minimum, mean of numeric values stored in a list; linear search on list of numbers and counting the frequency of elements in a list.

Main point

- **list** is an **ordered sequence** having **elements** of **different data types** enclosed in **square brackets** that are **separated** by **comma.** Example:- l1=[4, "CS", 25.5]
- list **not** support **vector operation**
- List are accessed by using syntax:- listname[index]
- Lists are **mutable** that mean the **contents of the list** can be **changed** after it has been created

List Operation:- Concatenation

- join two or more lists using concatenation operator depicted by the symbol +
- syntax: list1/listname1 + list2/listname2 , Example:- **[4,5]+ [7,8,9]** gives **[4,5,7,8,9]**
- **no changes** made by Concatenation operation **in original lists**
- **List** can **be concatenated with** another **list only.**
- try to **concatenate** a list **with other data type** will result in **TypeError**

List Operation:- Repetition

- **Replicate a list** using repetition operator depicted **by symbol ***
- syntax: list1/listname1 * n , Example:- **[5,6] * 3** gives **[5,6,5,6,5,6]**
- **no changes** made by repletion operation **in original list**

List Operation:- Membership

- membership operators '**in**' and '**not in'** returns **True or False**
- syntax: value in listname or value not in listname, Example:- 5 in [4,5,7] gives **True**
- **given value** is checked with **each element in list**

List Operation: - Slicing

- **access some part** of a list by method called **slicing**
- syntax:- listname[**start: end: step**] Example **l1[2:6:2]**
- **start , end , step** must be **integer (positive, zero or negative)**
- **default** value of **start is 0** and **end is n** where **n is length** of string and **step is 1** if not provided
- **default** value of **start is -1** and **end is –(n+1)** where **n is length** of string if **step =-1**

Traversing a String

- **access each element** of a list by using **for or while loop**
- Syntax:- for i in list_name:
- starts from the first element of the list and automatically ends when the last element is accessed from list.

Built-in Function

>>> l2=["ana","ravi","pankaj"]
>>> l=[7,8,11,3]

Method	Description	Example
len(list_name)	**Returns** the **length of** the given **list**	>>> len(l) 4

list(iterable=())	**Creates a list** having **element in iterable**, creates an **empty list if no argument** is passed	>>> l1=list("hello")
		>>> print(l1)
		['h', 'e', 'l', 'l', 'o']
		>>> l3=list(range(5,8))
		>>> print(l3)
		[5, 6, 7]

min(list_name)	**Returns** minimum or **smallest element** of the list
	List must contain **same type of value**
		>>> min(l)
		3
		>>> min(l2)
		'ana'

max(list_name)	**Returns** maximum or **largest element** of the list
	List must contain **same type of value**
		>>> max(l)
		11
		>>> max(l2)
		'ravi'

sum(iterable, start=0)	Returns **sum of the elements** of the list
	Work with **numeric values only** and reject non-numeric types
	Add start in result **if** start is **given**
		>>> sum(l)
		29
		>>> sum(l,10)
		39

sorted(iterable, reverse=False)	Return **new list** consisting of the **same elements** as original list **arranged in sorted order**	>>> sorted(l)
		[3, 7, 8, 11]
		>>> sorted(l2)
		['ana', 'pankaj', 'ravi']

List Methods

>>> list1 = [10,20,30,40]
>>> list2=[40,15,25,5]
>>> list3=[14,18,20,16]

Method	Description	Example

append(element)	Used to **add single element** in list **at the end** of list.
	Appends only a **single element** passed as an argument at the end of the list.
	That element can be a **number , string list , dictionary** or user defined object
		>>> list1.append(50)
		>>> list1
		[10, 20, 30, 40, 50]
		--- 	
		>>> list1.append([50,60])
		>>> list1
		[10, 20, 30, 40, [50, 60]]

extend(iterable)	Used to **add multiple elements** in list **at the end** of list.
	Appends each element of the list passed as argument to the end of the given list.
		>>> list4 = [50,60]
		>>> list1.extend(list4)
		>>> list1
		[10, 20, 30, 40, 50,60]

insert(index, element)	Used to **add single element** in list **before given index.**	>>> list1.insert(2,25)
		>>> list1
		[10, 20, 25, 30, 40]

	Inserts an element before the given index in the list	----------
		>>> list1.insert(-1,25)
		>>> list1
		[10, 20, 30, 25, 40]

remove(value)	**Remove** first occurrence of **value** from the list.
	Raises **ValueError if** the value is **not present**
		>>> list1.remove(30)
		>>> list1
		[10, 20, 40]
		>>> list1.remove(50)
		ValueError: list.remove(x): x not in list

pop(index=-1)	**Remove** and **return item** at **given index** (default last if index is not given)
	Raises **IndexError if** list is **empty** or index is **out of range**.
		>>> list1.pop(1)
		20
		>>> list1.pop()
		40

count(value)	Return **number of occurrences** of value.	>>> l=[2,4,2,'a']
		>>> l.count(2)
		2
		>>> l.count("a")
		1

index(value, start, stop)	Return **index of first occurrence** of value.
	Raises **ValueError if** the value is **not present**
		>>> list1.index(20)
		1
		>>> list1.index(90)
		ValueError: 90 is not in list

reverse()	**Reverse** the **element** of **list *IN PLACE*.**	>>> list2.reverse()
		>>> list2
		[5, 25, 15, 40]

sort(reverse=False)	**Sort** the list in **ascending order**
	The sort is **in-place** (i.e. the **list** itself is **modified**)
	The **reverse** flag can be set(**True**) to sort in **descending order.**
		>>> list2.sort()
		>>> list2
		[5, 15, 25, 40]
		>>> list3.sort(reverse=True)
		>>> list3
		[20, 18, 16, 14]

Nested Lists

- **list appears** as an **element of another list** Example:- >>> list1 = [1,2,'a','c'**,[6,7,8],**4,9]
- To **access** the **element** of the **nested list of list1**:- specify two indices **list1[i][j]**
- first index will **i take to the desired nested list** and second index **j will take to the desired element** in that nested list

Copying Lists

- **list2 = list1** does not create a new list. Rather, it just makes **list1 and list2 refer** to the **same list object** so **any changes** made to either of them **will be reflected** in the other list
- we can create a copy or clone of the list **as a distinct object** by **three methods**

1. list2=list1[:] 2. List2=list(list1)
3. import copy
   List2=copy.copy(list1)

List as Argument to a Function

Whenever a list is passed as an argument to a function, we have to consider two scenarios:

(A) Elements of the **original list may be changed**, i.e. **changes made** to the list in the function are **reflected back** in the calling function.

```
>>> list1 = [10,20,30,40]
>>> def add(l):
        i=int(input("enter number to inserted"))
        l.append(i)
        print("list in function:",l)
>>> add(list1)
enter number to inserted65
list in function: [10, 20, 30, 40, 65]
>>> print("original list:",list1)
original list: [10, 20, 30, 40, 65]
```

B) If the **list is assigned a new value** inside the function then a **new list object is created** and it becomes the **local copy of the function**. Any **changes made inside** the local copy of the function are **not reflected back** to the calling function.

```
>>> list1 = [10,20,30,40]
>>> def rep(l):
        l=l*2                    # list assigned with new value i.e. l*2
        l.append(len(l))
        print("list in function:",l)
>>> rep(list1)
list in function: [10, 20, 30, 40, 10, 20, 30, 40, 8]
>>> print("original list:",list1)
original list: [10, 20, 30, 40]
```

What you learned

List Manipulation:- list1 = [10,20,30,40] list2=[35,45,65]

- Append an element:- list1.append(20)
- Insert an element:- list1.insert(2,50)
- Append a list to the given list:- list1.extend(list2)
- Modify an existing element:- list1[3]=90
- Delete an existing element from its position:- list1.pop(3)
- Delete an existing element with a given value:- list1.remove(30)
- Sort the list in ascending order:- list1.sort()
- Sort the list in descending order :- list2.sort(reverse=True)
- Display the list. :- print(list1)

Question based on list

2023-main

1. Fill in the blank :

_____ is not a valid built-in function for list manipulations.

(a) count() (b) length() (c) append() (d) extend()

2. Which of the following functions is a valid built-in function for both list and dictionary datatype ?

(a) items() (b) len() (c) update() (d) values()

3. Write a function search_replace() in Python which accepts a list L of numbers and a number to be searched. If the number exists, it is replaced by 0 and if the number does not exist, an appropriate message is displayed. Example : L = [10,20,30,10,40]

Number to be searched = 10 , List after replacement :

L = [0,20,30,0,40]

2023

1. Fill in the blank.

_____ function is used to arrange the elements of a list in ascending order.

(a) sort() (b) arrange() (c) ascending() (d) asort()

2. Which function returns the sum of all elements of a list ?

(a) count() (b) sum() (c) total() (d) add()

3. Given is a Python list declaration :

Listofnames=["Aman","Ankit","Ashish","Rajan","Rajat"]

Write the output of :

print (Listofnames [–1:–4:–1])

4. Write the output of the code given below :

```
def short_sub (lst,n) :
    for i in range (0,n) :
   if len (lst)>4:
       lst [i]=lst [i]+lst[i]
   else:
       lst[i]=lst[i]
subject=['CS','HINDI','PHYSICS','CHEMISTRY','MATHS']
short_sub(subject,5)
print(subject)
```

5. Write a function EOReplace() in Python, which accepts a list L of numbers. Thereafter, it increments all even numbers by 1 and decrements all odd numbers by 1.

Example :

If Sample Input data of the list is : L=[10,20,30,40,35,55]

Output will be :L=[11,21,31,41,34,54]

2021-comp

1. A List is declared as L = ['ONE', 'TWO', 'THREE'], What will be the output?

```
print(max(L))
```

2. What shall be the ouput for the execution of the following Python Code ?

```
Cities = ['Delhi', 'Mumbai']
Cities[0], Cities[1] = Cities[1], Cities[0]
print(Cities)
```

3. Write the output for the execution of the following Python code : 2

```
def change(A):
  S=0
  for i in range(len(A)//2):
      S+=(A[i]*2)
  return S
B = [10,11,12,30,32,34,35,38,40,2]
C = Change(B)
Print('Output is',C)
```

4. Write the definition of a function Sum3(L) in Python, which accepts a list L of integers and displays the sum of all such integers from the list L which end with the digit 3. For example, if the list L is passed [123, 10, 13, 15, 23] then the function should display the sum of 123, 13, 23, i.e. 159 as follows :
Sum of integers ending with digit 3 = 159

2020- comp
1. Find and write the output of the following Python code :

```
def ChangeVal(M,N):
    for i in range(N):
       if M[i]%5 == 0:
        M[i] //= 5
       if M[i]%3 == 0:
        M[i] //= 3
L=[25,8,75,12]
ChangeVal(L,4)
for i in L :
    print(i, end='#')
```

2019
1. Write definition of a method/function AddOddEven(VALUES) to display sum of odd and even values separately from the list of VALUES.
For example : If the VALUES contain [15, 26, 37, 10, 22, 13]
The function should display
Even Sum: 58
Odd Sum: 65
2. Write definition of a method/function HowMany(ID,Val) to count and display number of times the value of Val is present in the list ID. 3 For example : If the ID contains [115,122,137,110,122,113] and Val contains 122 The function should display 122 found 2 Times

3. Write a python method/function Swapper(Numbers) to swap the first half of the content of a list Numbers with second half of the content of list Numbers and display the swapped values. 2 Note : Assuming that the list has even number of values in it. For example : If the

list Numbers contains [35,67,89,23,12,45] After swapping the list content should be displayed as [23,12,45,35,67,89]

4. Write a python method/function Count3and7(N) to find and display the count of all those numbers which are between 1 and N, which are either divisible by 3 or by 7. 2 For example : If the value of N is 15 The sum should be displayed as 7 (as 3,6,7,9,12,14,15 in between 1 to 15 are either divisible by 3 or 7)

2019

1. Write definition of a method/function DoubletheOdd(Nums) to add and display twice of odd values from the list of Nums. For example :If the Nums contains [25,24,35,20,32,41]
The function should display
Twice of Odd Sum: 202
OR
(b) Write definition of a method/function FindOut(Names, HisName) to search for HisName string from a list Names, and display the position of its presence. 3
For example : If the Names contain ["Arun","Raj","Tarun","Kanika"] and His Name contains "Tarun"
The function should display
Tarun at 2

2018

2. Find and write the output of the following python code :

```
Val   = [20,"A",40,"K",10,"H"]
Freq  = 0
Sum   = 0
Cat   = ""
For I in range(1,6,2):
    Freq = Freq + I
    Sum  = Sum + Val[I-1]
    Cat  = Cat + Val[I] + "*"
    print Freq,Sum,Cat
```

3. Write definition of a method Endingwith5(SCORES) to add all those values in the list of SCORES, which are ending with 5 and display the sum. For example,
If the SCORES contain [205,506,365,100,230,335]
The sum should be displayed as 905

4. Write definition of a Method COUNTNOW(REGIONS) to find and display names of those REGIONS, in which there are less than or equal to 5 characters. 2
For example : If the list REGIONS contains
["GOA","NEW DELHI","DAMAN","CHENNAI","BANGALORE"]
The following should get displayed
GOA
DAMAN

2017

(d) Find and write the output of the following Python code :

```
TXT   = ["20","50","30","40"]
CNT   = 3
TOTAL = 0
for C in [7,5,4,6]:
    T = TXT[CNT]
    TOTAL = float (T) + C
    print TOTAL
    CNT-=1
```

2. Write definition of a method EvenSum(NUMBERS) to add those values in the list of NUMBERS, which are odd.
3. Write definition of a method MSEARCH(STATES) to display all the state names from a list of STATES, which are starting with alphabet M. For example : If the list STATES contains ["MP","UP","WB","TN","MH","MZ","DL","BH","RJ","HR"]
The following should get displayed :
MP
MH
MZ

2017-delhi

(d) Find and write the output of the following Python code :

```
STR = ["90","10","30","40"]
COUNT = 3
SUM = 0
for I in [1,2,5,4]:
    S = STR[COUNT]
    SUM = float (S)+I
    print SUM
    COUNT-=1
```

2. Write definition of a method OddSum(NUMBERS) to add those values in the list of NUMBERS, which are odd.
3. Write definition of a Method AFIND(CITIES) to display all the city names from a list of CITIES, which are starting with alphabet A. For example :
If the list CITIES contains ["AHMEDABAD","CHENNAI","NEW DELHI","AMRITSAR","AGRA"]
The following should get displayed
AHMEDABAD
AMRITSAR
AGRA
2016

(d) Find and write the output of the following python code : 2

```
Numbers = [9,18,27,36]
for Num in Numbers:
    for N in range(1, Num%8):
        print(N,"#",end="")
    print()
```

2016-outside

(d) Find and write the output of the following Python code : 2

```
Values=[10,20,30,40]
for Val in Values:
    for I in range(1, Val%9):
        print(I,"*",end="")
    print()
```

SQP-2024

1. Assertion(A): List is an immutable data type
Reasoning(R): When an attempt is made to update the value of an immutable variable, the old variable is destroyed and a new variable is created by the same name in memory.

2. Predict the output of the following code:

```
S = "LOST"
L = [10,21,33,4]
D={}
for I in range(len(S)):
    if I%2==0:
        D[L.pop()] = S[I]
    else:
        D[L.pop()] = I+3

for K,V in D.items():
    print(K,V,sep="*")
```

3. Write the Python statement for each of the following tasks using BUILT-IN functions/methods only:
(i) To insert an element 200 at the third position, in the list L1.

SQP-2023
1. Write a function INDEX_LIST(L), where L is the list of elements passed as argument to the function. The function returns another list named 'indexList' that stores the indices of all Non-Zero Elements of L.For example:

If L contains [12,4,0,11,0,56]
The indexList will have - [0,1,3,5]
SQP-2020
1. Identify the output of the following Python statements.
x = [[10.0, 11.0, 12.0],[13.0, 14.0, 15.0]]
y = x[1][2]
print(y)
a. 12.0
b. 13.0
c. 14.0
d. 15.0
2. Identify the output of the following Python statements.
lst1 = [10, 15, 20, 25, 30]
lst1.insert(3, 4)
lst1.insert(2, 3)
print (lst1[-5])
a. 2
b. 3
c. 4
d. 20

3. What is the output of the following code snippet?

```
def ChangeVal(M,N):
    for i in range(N):
        if M[i]%5 == 0:
            M[i]//=5
        if M[i]%3 == 0:
            M[i]//=3
L = [25,8,75,12]
ChangeVal(L,4)
for i in L:
    print(i,end="#")
```

a) 5#8#15#4#
b) 5#8#5#4#
c) 5#8#15#14#
d) 5#18#15#4#

SQP-2021
1.Given the lists L=[1,3,6,82,5,7,11,92] , write the output of print(L[2:5])

3. Write a function LShift(Arr,n) in Python, which accepts a list Arr of numbers and n is a numeric value by which all elements of the list are shifted to left.
Sample Input Data of the list
Arr= [10,20,30,40,12,11], n=2
Output
Arr = [30,40,12,11,10,20