**Ex 7: Lists, Nested Lists and Lists Comprehension**

**Aim**

To do python programs using the list,nested list and list comprehension.

**Q.No 1 : Write a function Assign\_grade(Lst) which reads the marks of a student from a list and assigns a grade and display the grade based on the following conditions:**

**if Marks >=90 then grade A**

**if Marks >=80 && 65 && < 80 then grade C**

**if Marks > =40 && <=65 then grade D**

**if Marks>40 then grade F**

**Create the List of Marks of 5 Students in English Subject by getting the input from the user**

**Python Code**

#Assign grade to marks

'''1. Write a function Assign\_grade(Lst) which reads the marks of a student from a list

and assigns a grade and display the grade based on the following conditions:

if Marks >=90 then grade A

if Marks >=80 && <90 then grade B

if Marks >65 && < 80 then grade C

if Marks > =40 && <=65 then grade D

if Marks <40 then grade F

Create the List of Marks of 5 Students in English Subject by getting the input

from the user'''

grades=[]

def Assign\_grade(list):

for i in list:

if i>=90:

grade="A"

grades.append(grade)

elif i>=80 and i<90:

grade="B"

grades.append(grade)

elif i>65 and i<80:

grade="C"

grades.append(grade)

elif i>=40 and i<=65:

grade="D"

grades.append(grade)

else:

grade="F"

grades.append(grade)

return grades

list1=eval(input("Enter the marks of the students in english subject"))

list1=list(list1)

grades=Assign\_grade(list1)

for i in grades:

print(i)

**Test Cases:**

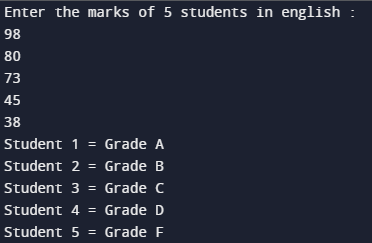
Student 1 = Grade A

Student 2 = Grade B

Student 3 = Grade C

Student 4 = Grade D

Student 5 = Grade F

**Output :**

**Q.No 2 : Generate a list of 10 elements between 0 to 6 using random number generator. Write a function rem\_dup( glist) which gets the generated list as input and remove the duplicates and return the updated list.**

**PYTHON CODE**

import random

a=[]

for i in range(10):

x=random.randrange(0,7)

a.append(x)

def rem\_dup(glist):

b=[]

for i in glist:

if i not in b:

b.append(i)

return b

print("Original list :",a)

print("List after deletion :",rem\_dup(a))

**Test Cases**

Original list : [4, 0, 5, 1, 3, 1, 4, 1, 5, 5]

List after deletion : [4, 0, 5, 1, 3]

**OUTPUT**

****

**Q.NO 3:** **Create a list arr with “n” integers, construct a new list prod\_list of same size such that prod\_list[i] is equal to the product of all the elements of arr except arr[i]. Eg: l1=[1,2,3] where l1[0]==2\*3, l1[1]=1\*3 and l1[2]=1\*3 Prod\_list=[6,3,2]**

**PYTHON CODE**

'''3. Create a list arr with “n” integers, construct a new list prod\_list of same size

such that prod\_list[i] is equal to the product of all the elements of arr except

arr[i].

Eg: l1=[1,2,3] where l1[0]==2\*3, l1[1]=1\*3 and l1[2]=1\*3

Prod\_list=[6,3,2]'''

list1=eval(input("Enter the list "))

list1=list(list1)

prod=[]

x=1

for i in range(0,len(list1)):

for j in range(0,len(list1)):

if i!=j:

x\*=list1[j]

prod.append(x)

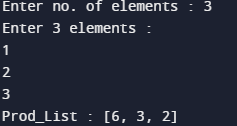
x=1

print(“prod\_list”,prod)

**TEST CASES**

Prod\_List : [6, 3, 2]

**OUTPUT**

****

**Q.NO 4:** **Write a function print\_reverse(Lst) to reverse the elements of a list. Pass the list as parameter to a function which reverses the list without using slice operation. Print the reversed list in the main program. (Avoid using return statement in function)**

**PYTHON CODE**

'''4. Write a function print\_reverse(Lst) to reverse the elements of a list. Pass the list

as parameter to a function which reverses the list without using slice operation.

Print the reversed list in the main program. (Avoid using return statement in

function)'''

list1=eval(input("enter the list"))

list1=list(list1)

def reverse(list1):

list2=list1.copy()

n=len(list1)

for i in range(0,n):

list1[i]=list2[n-1-i]

**reverse(list1)**

**print(list1)**

**TEST CASES**

The reversed list is:

[25, 11, 8, 7, 9, 5]

**OUTPUT**

****

**Q.NO 5:** **Create a list of names with 10 students and perform the following:**

**i. Print the list**

**ii. Search for a particular student (linear search)**

**iii. Sort the students list and then do search (binary search)**

**PYTHON CODE**

#linear search and binary search

def binary\_search(start,end,x):

if end>=start:

mid =start+(end-start)//2

if arr[mid]==x:

return mid

if x>arr[mid]:

return binary\_search(mid+1,end,x)

if x<arr[mid]:

return binary\_search(start,mid-1,x)

else:

print("Element not found")

arr=[]

n=int(input("Enter the length of the list"))

for i in range(0,n):

x=int(input("Enter a number"))

arr.append(x)

x=int(input("Enter the value to be found"))

#Bubble sort:

swap=True

while swap:

swap=False

for i in range(0,len(arr)):

for j in range(i+1,len(arr)):

if arr[i]>arr[j]:

arr[i],arr[j]=arr[j],arr[i]

swap=True

print(arr)

start=0

end=len(arr)

x=binary\_search(start,end,x)

print(x)

TEST CASES

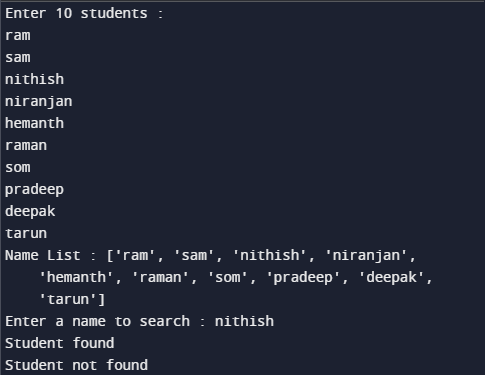
Name List : ['ram', 'sam', 'nithish', 'niranjan', 'hemanth', 'raman', 'som', 'pradeep', 'deepak', 'tarun']

Enter a name to search : nithish

Student found

Student not found

**OUTPUT**

****

**Q.NO 7: 7. Assume that the variable’ data’ refers to the list [5, 3, 7]. Write the expressions that perform the following tasks:**

**a. Replace the value at position 0 in data with that value’s negation.**

**b. Add the value 10 to the end of data.**

**c. Insert the value 22 at position 2 in data.**

**d. Remove the value at position 1 in data.**

**e. Add the values in the list newData = [9,11,13] to the end of data.**

**f. Locate the index of the value 7 in data, safely.**

**g. Find the maximum element in the list.**

**h. Find the sum of all elements in the list.**

**i. Sort the values in data.**

**PYTHON CODE**

lst=[5,3,7]

l=len(lst)

#a

lst[0]=-lst[0]

print(lst)

lst=[5,3,7]

#b

lst[l-1]=lst[l-1]+10

print(lst)

lst=[5,3,7]

#c

lst[2]=22

print(lst)

lst=[5,3,7]

#d

del lst[1]

print(lst)

lst=[5,3,7]

#e

newData=[9,11,13]

lst=lst+newData

print(lst)

lst=[5,3,7]

#f

print("Index of 7 :",lst.index(7))

#g

print("Maximum =",max(lst))

#e

print("Sum =",sum(lst))

#f

print("Sorted list =",sorted(lst))

**TEST CASES**

[-5, 3, 7]

[5, 3, 17]

[5, 3, 22]

[5, 7]

[5, 3, 7, 9, 11, 13]

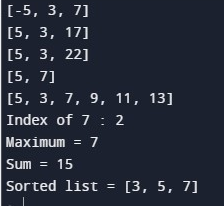
Index of 7 : 2

Maximum = 7

Sum = 15

Sorted list = [3, 5, 7]

**OUTPUT**

****

**Q.NO 8:** **Write a Python program that prompts the user to enter a list of names and stores them in a list. The program should display how many times the letter 'a' appears within the list.**

**PYHTON CODE**

'''Write a Python program that prompts the user to enter a list of names and stores them in a list. The program should display how many times the letter 'a' appears within the list.'''

list1=[]

count=0

n=int(input("enter the number of elements in a list"))

for i in range(0,n):

x=input("enter the name")

list1.append(x)

for i in range(0,len(list1)):

count+=list1[i].count("a")

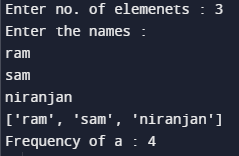
print(count)

**TEST CASES**

['ram', 'sam', 'niranjan']

Frequency of a : 4

**OUTPUT**

****

**Q.NO 9:** **Create two matrices and perform the following actions:**

**i) Pass 2 matrices to a function called add\_mat(m1,m2). Compute addition of 2 matrices and return the resultant matrix.**

**ii) Pass 2 matrices to a function called mul\_mat(m1,m2). Compute multiplication of 2 matrices.**

**PYTHON CODE**

a=[[1,2],[3,4]]

b=[[2,1],[3,3]]

def add\_mat(m1,m2):

mat=[[0,0],[0,0]]

for i in range(2):

for j in range(2):

mat[i][j]=m1[i][j]+m2[i][j]

return mat

def mul\_mat(m1,m2):

mat=[[0,0],[0,0]]

for i in range(2):

for j in range(2):

for k in range(2):

mat[i][j]+=(m1[i][k]\*m2[k][j])

return mat

print("ADDITION : ")

print(add\_mat(a,b))

print("MULTIPLICATION : ")

print(mul\_mat(a,b))

**TEST CASES**

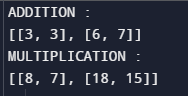
ADDITION :

[[3, 3], [6, 7]]

MULTIPLICATION :

[[8, 7], [18, 15]]

**OUTPUT**

****

**Q.NO 10: Give an appropriate list comprehension for each of the following:**

**i) L1 = [1, ’x’, 4, 5.6, ’z’, 9, ‘a’, 0, 4] create a list which consists of integer values.**

**ii) Producing a list of consonants that appear in string w.**

**iii) Multiples of 10 (n values)**

**iv) Construct a list of the form: [‘1a’,’2a’,’3a’,’4a’]**

**v) Create a list which stores the sum of each of the elements from the two lists.**

**PYTHON CODE**

import random

#1

L1 = [1, 'x', 4, 5.6, 'z', 9, 'a', 0, 4]

l = [i for i in L1 if str(i).isdigit()]

print(l)

#2

w = 'NOice'

v='aeiouAEIOU'

l = [i for i in w if i not in v]

print(l)

#3

n = int(input('Enter the number of terms required : '))

l = [10\*(i+1) for i in range(n)]

print(l)

#4

l = [str(i)+'a' for i in range(1,5)]

print(l)

#5

l1 = [random.randint(1,10) for i in range(10)]

l2 = [random.randint(1,10) for i in range(10)]

print(l1,l2,sep="\n")

l = [l1[i]+l2[i] for i in range(10)]

print(l)

**TEST CASES**

[1, 4, 9, 0, 4]

['N', 'c']

[10, 20, 30, 40, 50]

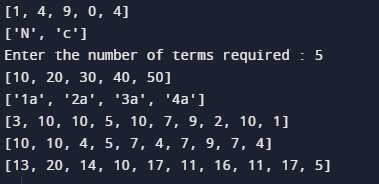
['1a', '2a', '3a', '4a']

[3, 10, 10, 5, 10, 7, 9, 2, 10, 1]

[10, 10, 4, 5, 7, 4, 7, 9, 7, 4]

[13, 20, 14, 10, 17, 11, 16, 11, 17, 5]

**OUTPUT**

****

**Q.NO 11:** **Find the transpose of a given matrix using list comprehension.**

**PYTHON CODE**

**#Transpose using list comprehenson:**

m = [[1,2],[3,4],[5,6]]

for row in m :

print(row)

rez = [[m[j][i] for j in range(len(m))] for i in range(len(m[0]))]

print("\n")

for row in rez:

print(row)

matrix=[(1,2,3),(4,5,6),(7,8,9),(10,11,12)]

for row in matrix:

print(row)

print("\n")

t\_matrix = zip(\*matrix)

for row in t\_matrix:

print(row)

**TEST CASES**

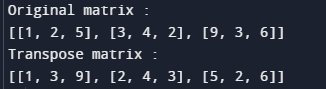
Original matrix :

[[1, 2, 5], [3, 4, 2], [9, 3, 6]]

Transpose matrix :

[[1, 3, 9], [2, 4, 3], [5, 2, 6]]

**OUTPUT**

****

**Learning Outcomes**

By completing this exercise i have learnt to use lists,nested lists and list comprehension in programs and i have done all the program using lists,nested lists and list comprehension.