CSX3001/ITX3001
FUNDAMENTALS OF COMPUTER PROGRAMMING

CLASS 09 NESTED LIST

NESTED LIST, INDEXING, AND MATRIX

PYTHON

NESTED LISTS

It is possible to nest lists into another list. With a nested list, a new dimension is created. To access nested lists, it needs additional square brackets([]).

```
### Example#1
# nList contains the other three lists
nList = [[1,2,3],[4,5,6],[7,8,9]]

# print the whole nList
print(nList);

### Example#2
# print each list in nList
print(nList[0])
print(nList[1])
print(nList[2])

# the above 3 statements code are equivalent to the following code
for eachL in nList:
  print(eachL)

# or using index to access each list
for i in range(len(nList)):
  print(nList[i])

### Example#3
# each element in each individual sub-list can be accessed using [ ][ ] indexes
nList[2][0] *= 2
nList[2][1] *= 3
nList[2][2] *= 4
print(nList[2])
```

In Example#1, nList is created which contains other three lists. Printing this list, nList, will show all nested lists. In Example#2, Printing each sub-list in the nested nList can be performed by using single index to access each sub-list. nList is possible to iterate through for loop. In Example#3, an additional index is needed to access each element in a nested list, nList.

NESTED LIST INDEXES

Each item in a nested list can be accessed via multiple index operators ([ ][ ]).

```
### Example#4
exList = []
exList.append([2,4,6])
print(exList)
exList.append([8,10,12])
print(exList)

for i in range(len(exList)):
  for j in range(len(exList[i])):
      print(exList[i][j],end=' ')
  print()
```

Run the fragment of code in Example#4 and answer the following questions.

- Observe what results are printed out

_____

_____

What is/are the different between len(exList) and len(exList[i])?

_____

_____

What will happen if you remove the last print() statement?

_____

_____

If you want to print out only 6 and 10 in exList, what will be the index of these two elements?

_____print(exList[__][__])_____

_____print(exList[__][__])_____

USING NESTED LIST TO REPRESENT MATRIX

In other programming languages, matrix can be presented by using 2-dimensional array. In Python, one possible way is to use a nested list to represent matrix. Note: The first index in a list is 0.

```
# 2x2 matrix
matrixA = [[1,3], [5,7]]
#3x3 matrix
matrixB = [[0.5,1.6,7.9], [2.2,4.0,5.6], [3.5,9.8,2.9]]

nRow = len(matrixB)
nColumn = len(matrixB[0])
for row in range(nRow):
  for col in range(nColumn):
     print(matrixB[row][col], end=' ')
  print()
```

Each element in matrixA and matrixB can be accessed by using two indexes which represent row# and column#, respectively.

◆ LIST EXERCISES

Complete the following exercises in Python IDLE or Jupyter notebook.

1) With any two lists of integer values where the first list is always smaller than the second list, if the short list is a subset of a long list, the code prints "Yes". Otherwise, the code prints "No." For examples

   List_1 = [3, 4]
   List_2 = [3,6,7,4]

   Output is: "Yes".

   List_1 = [6,7,0]
   List_2 = [3,5,7,8,9,0]

   Output is: "No".

   Turn the above code into a function, namely SubsetList(List_1, List_2). One style is to print an output inside a function. Another style is to return an output, and print a returned answer outside a function.

2) Write a Python code to split a list of values (either string, integer or floating-point values) into a list of integers and a list of floating-point values.

   For example:

   NumList = [1, 4.9, 4, "Five", 6, 7, "Eight", 100.2, 15]

   Outputs
   StrList = ["Five", "Eight"]
   IntList = [1,4,6,7,15]
   FloatList = [4.9, 100.2]

   Turn the above code into a function, namely SeparateList(NumList). One style is to print an output inside a function. Another style is to return an output, and print a returned answer outside a function.

3) With any two lists of integers with a length of *m* and *n*, write a Python code that prints a multiplication table in a form of a matrix *m by n* (and also *n by m*), with fact that the matrix shall print only integer values less than 100 (substitute integer values of 100 or over by ***). For example:

   List_1 = [2,4,10]
   List_2 = [1,5,10,20]

   Output#1 is

```
 2    4    10
10   20    50
20   40   ***
40   80   ***
```

and

Output#2 is

```
2      10     20     40
4      20     40     80
10     50    ***    ***
```

Turn the above code into a function, namely MultiplicationMatrix(List_1, List_2). One style is to print an output inside a function. Another style is to return an output, and print a returned answer outside a function.

4) Write a Python code to replace the first and last elements in a list (List_1) with another two lists (List_2 and List_3). For example:

List_1 = [1,3,5,6,7,8]
List_2 = [10,20,30]
List_3 = [11,22,33]

Output is [10,20,30,3,5,6,7,11,22,33]

List_1 = [5,6,7,8]
List_2 = [11,22]
List_3 = [33,44]

Output is [11,22,6,7,8,33,44]

Turn the above code into a function, namely ReplaceElement(List_1, List_2, List_3). One style is to print an output inside a function. Another style is to return an output, and print a returned answer outside a function.

5) Write a Python code to find a maximum and a minimum integer value in any given nested list.

nList = [[2,5,99],[-3,8,9,10],[1, 7,100]]

Output is:
The max value is 100.
The min value is -3

Turn the above code into a function, namely FindMaxMin(nList). One style is to print an output inside a function. Another style is to return an output, and print a returned answer outside a function.

6) Write a Python code to print all duplicated integer values from a given list.

nList = [[2,5,99,99],[-3,8,1,2,10],[1, 7,100,10]]

Output is:
oList = [[2, 5, 99], [-3, 8, 1, 10], [7, 100]]

Turn the above code into a function, namely RemoveDup(nList).

ASSIGNMENTS

Complete the following assignments. You must name the python file as, {your-id}_class0{number}_{ section-number}_as{number}.py
for example, for assignment 1 will be named,

6120001_class09_ 541_as1.py

1) With any two pre-defined lists of integer values, the code prints "Yes, {small list} is a subset of {large list}." if a small list is a subset of a long list. Otherwise, the code prints "No {short list} is not a subset of {long list}.".

   For examples,
   List_1 = [3, 4]
   List_2 = [3,6,7,4]
   Output: Yes List_1 is a subset of List_2.

   List_1 = [3,5,7,8,9,0]
   List_2 = [5,0]
   Output: Yes List_2 is a subset of List_1.

   List_1 = [3,5,7,8,9,0]
   List_2 = [5,0,0]
   Output: Yes List_2 is a subset of List_1.

2) For any pre-defined nested list, write a code namely OddEvenList(nList) that separates odd and even number into oddList[ ] and evenList[ ]. For example,

   nList = [[2,5,99,99], [-3,8,1,2,10], [1, 7,100,10]] #pre-defined list

   Output: oddList = [[2], [8, 2, 10], [100, 10]]
   Output: evenList = [[5, 99, 99], [-3, 1], [1, 7]]