algorithmica

Programming Language Learning Series
Mastery of Python Language
(Interview Questions/Assignment-List&Tuple)

Q1: Create a list with the values 1,2,4,6,6,9,10,11,14. Use indexing to

- print the 3rd value
- print the last 2 values
- print only the odd values

Q2: Convert the following logic of list normalization into list comprehension.

```
num_list = [6,4,2,8,9,10,3,2,1,3]
total = float(sum(num_list))
for i in range(len(num_list)):
        num_list[i] = num_list[i]/float(total)
```

Q3: Convert the following logic of zero-matrix creation into list comprehension.

```
zerosmatrix = []
for i in range(10):
        matrix.append([0]*10)
```

Q4: By using list comprehension, write a program for the following:

- print the list after removing numbers which are divisible by 3 and 5 in [5, 10, 40, 7, 53, 31, 12, 101]
- print the list after removing the 0th, 2nd, 4th,6th numbers in [5, 10, 40, 7, 53, 31, 12, 101]
- generate a 3 * 4 2D-list whose elements are initialized to 0
- generate 2*3*5 3D-list whose each elements are initialized to 0

Q5: Which of the following functions is faster? Why?

```
def concat1(alist):
 """Using string concatenation."""
        s = alist[0]
        for item in alist[1:]:
                s += " " + item
        return s

def concat2(alist):
"""Using join."""
        return " ".join(alist)
```

Q6: Write a function to generate all possible sentences, using list comprehension, that can be formed using following lists: *["Python", "Julia", "Scala"], ["learning", "language"], ["beautiful","fun"]*. Here, you will get 12 possible sentences in total.

Programming Language Learning Series
Mastery of Python Language
(Interview Questions/Assignment-List&Tuple)

Q7: Write a function that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.
Input: education, is, experience, joyful, fun, not, mess
Output: education, experience, fun, is, joyful, mess, not

Q8: What is the output of the following program?

```
def h(n, xs=[]):
    for i in range(n):
        xs.append(i)
    return xs

n = 5
xs = [1,2,3]
h(n)
h(n)
```

Q9: Write a function that can flatten a nested list of arbitrary depth.
Input: [1,[2,3],[4,[5,[6,7],8],9],10,[11,12]])
Output: [1,2,3,4,5,6,7,8,9,10,11,12]

Q10: Write a function to sort the list of (name, age, height) tuples by ascending order where name is string, age and height are numbers. The sort criteria is:

1. Sort based on name
2. Then sort based on age
3. Then sort by score
4. The priority is that name > age > score

Input: [('Tagore',19,80), ('Joseph',20,90), ('Jony',17,91), ('Joseph',17,94), ('Aurobindo',21,85)]
Output: [('Aurobindo',21,85), ('Jony',17,91), ('Joseph',17,94), ('Joseph',20,90), ('Tagore',19,80)]

Q11: A robot moves in a plane starting from the original point (0,0). The robot can move toward UP, DOWN, LEFT and RIGHT with a given steps. The numbers after the direction are steps. Write a program to compute the distance between the position after a sequence of movements and original point. If the distance is a float, then just print the nearest integer.
Input: [('UP',5), ('DOWN',3), ('LEFT',3), ('RIGHT',2)]
Output: 2

Programming Language Learning Series
Mastery of Python Language
(Interview Questions/Assignment-List&Tuple)

Q12: What is the output of the following code snippets? Which of the following code snippets perform deep copy of list?

Snippet1:

```
list1 = [['a'],['b'],['c']]
list2 = list1
list1.append(['d'])
print(list2)
```

Snippet2:

```
list1 = [['a'],['b'],['c']]
list2 = list1.copy()
list1.append(['d'])
print(list2)
list1[0][0] = ['d']
print(list2)
```

Snippet3:

```
import copy
list1 = [['a'],['b'],['c']]
list2 = copy.deepcopy(list1)
list1.append(['d'])
print(list2)
list1[0][0] = ['d']
print(list2)
```