Extra Python Practice Questions

1. Which of the following statements create a dictionary?(multiple answers allowed)
a) d = {}
b) d = {“john”:40, “peter”:45}
c) d = {40:”john”, 45:”peter”}
d) d = (40:”john”, 45:”peter”)

2. What are the keys?
```
d = {"john":40, "peter":45}
```
a) “john”, 40, 45, and “peter”
b) “john” and “peter”
c) 40 and 45
d) d = (40:”john”, 45:”peter”)

3. What will be the output?
```
d = {"john":40, "peter":45}
"john" in d
```
a) True b) False
c) None d) Error

4. What will be the output?
```
d1 = {"john":40, "peter":45}
d2 = {"john":466, "peter":45}
d1 == d2
```
a) True b) False
c) None d) Error

5. What will be the output?
```
d1 = {"john":40, "peter":45}
d2 = {"john":466, "peter":45}
d1 > d2
```
a) True b) False
c) Error d) None

6. What is the output?
```
d = {"john":40, "peter":45}
d["john"]
```
a) 40 b) 45 c) “john” d) “peter”

7. Suppose d = {“john”:40, “peter”:45}, to delete the entry for “john” what command do we use
a) d.delete(“john”:40) b) d.delete(“john”)
c) del d[“john”] d) del d(“john”:40)

8. Suppose d = {“john”:40, “peter”:45}, to obtain the number of entries in dictionary what command do we use
a) d.size() b) len(d)
c) size(d) d) d.len()

9. What will be the output?
```
d = {"john":40, "peter":45}
print(list(d.keys()))
```
a) [“john”, “peter”] b) [“john”:40, “peter”:45] c) (“john”, “peter”)
d) (“john”:40, “peter”:45)

10. Suppose d = {“john”:40, “peter”:45}, what happens when retieving a value using d[“susan”]?
a) Since “susan” is not a value in the set, Python raises a KeyError exception.
b) It is executed fine and no exception is raised, and it returns None.
c) Since “susan” is not a key in the set, Python raises a KeyError exception.
d) Since “susan” is not a key in the set, Python raises a syntax error.

1. What will be the output?

```
>>>m = [[x, x + 1, x + 2] for x in range(0, 3)]
```
a) [[1, 2, 3], [4, 5, 6], [7, 8, 9]] b) [[0, 1, 2], [1, 2, 3], [2, 3, 4]] c) [1, 2, 3, 4, 5, 6, 7, 8, 9] d) [0, 1, 2, 1, 2, 3, 2, 3, 4]

11. How many elements are in m?
```
m = [[x, y] for x in range(0, 4) for y in range(0, 4)]
```
a) 8 b) 12 c) 16 d) 32

12. What will be the output?
```
values = [[3, 4, 5, 1], [33, 6, 1, 2]]
v = values[0][0]
for row in range(0, len(values)):
    for column in range(0,
len(values[row])):
        if v < values[row][column]:
            v = values[row][column]
print(v)
```
a) 3 b) 5 c) 6 d) 33

13.What will be the output?
```
values = [[3, 4, 5, 1], [33, 6, 1, 2]]
v = values[0][0]
for lst in values:
    for element in lst:
        if v > element:
            v = element
 print(v)
```
a) 1 b) 3 c) 5 d) 6

14. What will be the output?
```
values = [[3, 4, 5, 1 ], [33, 6, 1, 2]]
for row in values:
    row.sort()
    for element in row:
        print(element, end = " ")
    print()
```
a) The program prints two rows 3 4 5 1 followed by 33 6 1 2
b) The program prints on row 3 4 5 1 33 6 1 2
c) The program prints two rows 3 4 5 1 followed by 33 6 1 2
d) The program prints two rows 1 3 4 5 followed by 1 2 6 33

16. What is the output?
```
matrix = [[1, 2, 3, 4],
       [4, 5, 6, 7],
       [8, 9, 10, 11],
       [12, 13, 14, 15]]
for i in range(0, 4):
    print(matrix[i][1], end = " ")
```
a) 1 2 3 4
b) 4 5 6 7
c) 1 3 8 12
d) 2 5 9 13

17. What will be the output?
```
def m(list):
    v = list[0]
    for e in list:
      if v < e: v = e
    return v
values = [[3, 4, 5, 1], [33, 6, 1, 2]]
for row in values:
    print(m(row), end = " ")
```
a) 3 33 b) 1 1
c) 5 6 d) 5 33

18. What will be the output?
```
data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]
print(data[1][0][0])
```
a) 1 b) 2 c) 4 d) 5

19. What will be the output?
```
data = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]
def ttt(m):
    v = m[0][0]
    for row in m:
        for element in row:
           if v < element: v = element
    return v
print(ttt(data[0]))
```
a) 1 b) 2 c) 4 d) 5

20. What will be the output?
```
points = [[1, 2], [3, 1.5], [0.5, 0.5]]
points.sort()
print(points)
```
a) [[1, 2], [3, 1.5], [0.5, 0.5]] b) [[3, 1.5], [1, 2], [0.5, 0.5]] c) [[0.5, 0.5], [1, 2], [3, 1.5]]
d) [[0.5, 0.5], [3, 1.5], [1, 2]]

21. What is the output of the following?
```
k = [print(i) for i in my_string if i not in "aeiou"]
```
a) prints all the vowels in my_string
b) prints all the consonants in my_string
c) prints all characters of my_string that aren’t vowels
d) prints only on executing print(k)

22. What is the output of print(k) in the following?
```
k = [print(i) for i in my_string if i not in "aeiou"]
print(k)
```
a) all characters of my_string that aren’t vowels b) a list of Nones
c) list of Trues
d) list of Falses

23. What is the output of the following?

```
my_string = "hello world"
k = [(i.upper(), len(i)) for i in my_string]
print(k)
```

a) [(‘HELLO’, 5), (‘WORLD’, 5)] b) [(‘H’, 1), (‘E’, 1), (‘L’, 1), (‘L’, 1), (‘O’, 1), (‘ ‘, 1), (‘W’, 1), (‘O’, 1), (‘R’, 1), (‘L’, 1), (‘D’, 1)] c) [(‘HELLO WORLD’, 11)] d) none of the mentioned

24. Which of the following is the correct expansion of list_1 = [expr(i) for i in list_0 if func(i)] ?

a)

```
list_1 = []
for i in list_0:
    if func(i):
        list_1.append(i)
```

b)

```
for i in list_0:
    if func(i):
        list_1.append(expr(i))
```

c)

```
list_1 = []
for i in list_0:
    if func(i):
        list_1.append(expr(i))
```

d) none of the mentioned

25. What is the output of the following?

```
x = [i**+1 for i in range(3)]; print(x);
```

a) [0, 1, 2] b) [1, 2, 5] c) error, **+ is not a valid operator
d) error, ‘;’ is not allowed

26. What is the output of the following?

```
print([i.lower() for i in "HELLO"])
```

a) [‘h’, ‘e’, ‘l’, ‘l’, ‘o’] b) ‘hello’
c) [‘hello’] d) hello

27. What is the output of the following?

```
print([i+j for i in "abc" for j in "def"])
```

advertisements
a) [‘da’, ‘ea’, ‘fa’, ‘db’, ‘eb’, ‘fb’, ‘dc’, ‘ec’, ‘fc’] b) [[‘ad’, ‘bd’, ‘cd’], [‘ae’, ‘be’, ‘ce’], [‘af’, ‘bf’, ‘cf’]] c) [[‘da’, ‘db’, ‘dc’], [‘ea’, ‘eb’, ‘ec’], [‘fa’, ‘fb’, ‘fc’]] d) [‘ad’, ‘ae’, ‘af’, ‘bd’, ‘be’, ‘bf’, ‘cd’, ‘ce’, ‘cf’]

28. What is the output of the following?

```
print([[i+j for i in "abc"] for j in "def"])
```

a) [‘da’, ‘ea’, ‘fa’, ‘db’, ‘eb’, ‘fb’, ‘dc’, ‘ec’, ‘fc’] b) [[‘ad’, ‘bd’, ‘cd’], [‘ae’, ‘be’, ‘ce’], [‘af’, ‘bf’, ‘cf’]] c) [[‘da’, ‘db’, ‘dc’], [‘ea’, ‘eb’, ‘ec’], [‘fa’, ‘fb’, ‘fc’]] d) [‘ad’, ‘ae’, ‘af’, ‘bd’, ‘be’, ‘bf’, ‘cd’, ‘ce’, ‘cf’]

29. What is the output of the following?

```
print([if i%2==0: i; else: i+1; for i in range(4)])
```

a) [0, 2, 2, 4] b) [1, 1, 3, 3] c) error
d) none of the mentioned

30. Which of the following is the same as list(map(lambda x: x**-1, [1, 2, 3]))?
a) [x**-1 for x in [(1, 2, 3)]] b) [1/x for x in [(1, 2, 3)]] c) [1/x for x in (1, 2, 3)] d) error

31. What is the output of the following?

```
elements = [0, 1, 2]
def incr(x):
    return x+1
print(list(map(elements, incr)))
```

a) [1, 2, 3] b) [0, 1, 2] c) error
d) none of the mentioned

32. What is the output of the following?

```
elements = [0, 1, 2]
def incr(x):
    return x+1
print(list(map(incr, elements)))
```

a) [1, 2, 3] b) [0, 1, 2] c) error
d) none of the mentioned

33. What is the output of the following?

```
x = ['ab', 'cd']
print(list(map(upper, x)))
```

a) [‘AB’, ‘CD’] b) [‘ab’, ‘cd’] c) error
d) none of the mentioned

34. What is the output of the following?

```
def to_upper(k):
    return k.upper()
x = ['ab', 'cd']
print(list(map(upper, x)))
```

a) [‘AB’, ‘CD’] b) [‘ab’, ‘cd’] c) none of the mentioned
d) error

35. What is the output of the following?

```
def to_upper(k):
    return k.upper()
x = ['ab', 'cd']
print(list(map(to_upper, x)))
```

a) [‘AB’, ‘CD’] b) [‘ab’, ‘cd’] c) none of the mentioned d) error

36. What is the output of the following?

```
def to_upper(k):
    k.upper()
x = ['ab', 'cd']
print(list(map(to_upper, x)))
```

a) [‘AB’, ‘CD’] b) [‘ab’, ‘cd’] c) none of the mentioned d) error

37. What is the output of the following?

```
x = ['ab', 'cd']
print(map(len, x))
```

a) [‘ab’, ‘cd’] b) [2, 2] c) [‘2’, ‘2’] d) none of these

38. What is the output of the following?

```
x = ['ab', 'cd']
print(list(map(len, x)))
```

a) [‘ab’, ‘cd’] b) [2, 2] c) [‘2’, ‘2’] d) none of these

39. What is the output of the following?

```
x = ['ab', 'cd']
print(len(map(list, x)))
```

a) [2, 2] b) 2   c) 4 d) none of these

40. What is the output of the following?

```
x = ['ab', 'cd']
print(len(list(map(list, x))))
```

a) 2 b) 4 c) error d) none of these

41 Python supports the creation of anonymous functions at runtime, using a construct called __________?
a) lambda b) pi c) anon d) None of these

42. What is the output of this program?

```
y = 6
z = lambda x: x * y
print z(8)
```

a) 48 b) 14 c) 64   d) None of these

43. What is the output of below program?

```
lamb = lambda x: x ** 3
print(lamb(5))
```

a) 15 b) 555 c) 125 d) None of these

44. Is Lambda contains return statements
a) True b) False

45. Lambda is a statement. a) True b) False

46. Lambda contains block of statements
a) True b) False

47. What is the output of below program?

```
def f(x, y, z): return x + y + z
f(2, 30, 400)
```

a) 432 b) 24000 c) 430 d) None of these

48.What is the output of below program?

```
def writer():
            title = 'Sir'
            name = (lambda x:title + ' ' + x)
            return name
who = writer()
who('Arthur')
```

a) Arthur Sir b) Sir Arthur
c) Arthur d) None of these

49. What is the output of this program?

```
L = [lambda x: x ** 2, lambda x: x ** 3,
      lambda x: x ** 4]
for f in L:
   print(f(3))
```

a) 27 81 343 b) 6 9   12
c) 9 27 81 d) None of these

50. What is the output of this program?

```
min = (lambda x, y: x if x < y else y)
min(101*99, 102*98)
```

a) 9997 b) 9999 c) 9996 d) None of these