

1. The `\n` digraph forces the `print()` function to:
 - a. duplicate the character next to the digraph
 - b. stop its execution
 - c. output exactly two characters: `\` and `n`
 - d. break the output line
2. The meaning of the keyword parameter is determined by:
 - a. its position within the argument list
 - b. its connection with existing variables
 - c. its value
 - d. the argument's name specified along with its value
3. The value twenty point twelve times ten raised to the power of eight should be written as:
 - a. `20E12.8`
 - b. `20.12E8.0`
 - c. `20.12*10^8`
 - d. `20.12E8`
4. The `0o` prefix means that the number after it is denoted as:
 - a. hexadecimal
 - b. binary
 - c. decimal
 - d. octal
5. The `**` operator:
 - a. performs floating-point multiplication
 - b. performs exponentiation
 - c. does not exist
 - d. performs duplicated multiplication
6. The result of the following division:

$1 / 1$

 - a. cannot be predicted
 - b. is equal to 1
 - c. cannot be evaluated
 - d. is equal to 1.0
7. Only one of the following statements is false – which one?
 - a. the right argument of the `%` operator cannot be zero
 - b. the result of the `/` operator is always an integer value
 - c. multiplication precedes addition
 - d. the `**` operator uses right sided binding

8. Left sided binding determines that the result of the following expression is equal to:

```
1 // 2 * 3
```

- a. 0.16666666666666666
 - b. 0**
 - c. 0.0
 - d. 4.5
9. One of the following variables' names is illegal – which one?
- a. tTRUE
 - b. TRUE
 - c. true
 - d. True
10. The print() function can output values of:
- a. any number of arguments (including zero)**
 - b. not more than five arguments
 - c. just one argument
 - d. any number of arguments (excluding zero)

11. What is the output of the following snippet?

```
x=1  
y=2  
z=x  
x=y  
y=z  
print(x,y)
```

- a. 2 2
 - b. 1 2
 - c. 1 1
 - d. 2 1**
12. What is the output of the following snippet if the user enters two lines containing 2 and 4 respectively?

```
x=input()  
y=input()  
print(x+y)
```

- a. 6
- b. 4
- c. 2
- d. 24**

13. What is the output of the following snippet if the user enters two lines containing 2 and 4 respectively?

```
x=int(input())
y=int(input())
x=x//y
y=y//x
print(y)
```

- a. 8.0
- b. the code will cause a runtime error**
- c. 2.0
- d. 4.0

14. What is the output of the following snippet if the user enters two lines containing 2 and 4 respectively?

```
x=int(input())
y=int(input())
x=x/y
y=y/x
print(y)
```

- a. 4.0
- b. 8.0**
- c. the code will cause a runtime error
- d. 2.0

15. What is the output of the following snippet if the user enters two lines containing 11 and 4 respectively?

```
x=int(input())
y=int(input())
x = x % y
x = x % y
y = y % x
print(y)
```

- a. 4
- b. 2
- c. 3
- d. 1**

16. What is the output of the following snippet if the user enters two lines containing 3 and 6 respectively?

```
x=input()
y=int(input())
print(x*y)
```

- a. 666
- b. 36
- c. 18
- ☒ d. 333333

17. What is the output of the following snippet?

```
z = y = x = 1
print(x,y,z,sep='*')
```

- a. x y z
- b. 1 1 1
- c. x*y*z
- ☒ d. 1*1*1

18. What is the output of the following snippet?

```
x = 2 + 3 * 5.
print(X)
```

- ☒ a. 17.0
- b. 17
- c. 25.
- d. the snippet will cause an execution error

19. What is the output of the following snippet?

```
x = 1 / 2 + 3 // 3 + 4 ** 2
print(x)
```

- a. 17
- ☒ b. 17.5
- c. 8
- d. 8.5

20. What is the output of the following snippet if the user enters two lines containing 2 and 4 respectively?

```
x=int(input())
y=int(input())
print(x+y)
```

- a. 2
- b. 24
- c. 6**
- d. 4

21. An operator able to check whether two values are equal is coded as:

- a. =
- b. ===
- c. !=
- d. ==**

22. The value eventually assigned to x is equal to:

```
x = 1
x = x == x
```

- a. True**
- b. 0
- c. 1
- d. False

23. How many stars will the following snippet send to the console?

```
i = 0
while i <= 3 :
    i += 2
    print("*")
```

- a. two**
- b. one
- c. zero
- d. three

24. How many stars will the following snippet send to the console?

```
i = 0
while i <= 5 :
    i += 1
    if i % 2 == 0:
        break
    print("*")
```

- a. zero
- b. three
- c. two
- d. one**

25. How many hashes will the following snippet send to the console?

```
for i in range(1):
    print("#")
else:
    print("#")
```

- a. three
- b. zero
- c. two**
- d. one

26. How many hashes will the following snippet send to the console?

```
var = 0
while var < 6:
    var += 1
    if var % 2 == 0:
        continue
    print("#")
```

- a. three**
- b. zero
- c. two
- d. one

27. How many hashes will the following snippet send to the console?

```
var = 1
while var < 10:
    print("#")
    var = var << 1
```

- a. one
- b. four**
- c. eight
- d. Two

28. What value will be assigned to the x variable?

```
z = 10
y = 0
x = y < z and z > y or y > z and z < y
```

- ☒ a. True
- b. 1
- c. False
- d. 0

29. What is the output of the following snippet?

```
a = 1
b = 0
c = a & b
d = a | b
e = a ^ b
print(c + d + e)
```

- ☒ a. 2
- b. 0
- c. 3
- d. 1

30. What is the output of the following snippet?

```
lst = [3, 1, -2]
print(lst[lst[-1]])
```

- a. -2
- ☒ b. 1
- c. 3
- d. -1

31. What is the output of the following snippet?

```
lst = [1,2,3,4]
print(lst[-3: -2])
```

- ☒ a. [2]
- b. [2,3,4]
- c. []
- d. [2,3]

32. The second assignment:

```
vals = [0, 1, 2]  
vals[0], vals[2] = vals[2], vals[0]
```

- ☒ a. reverses the list
- ☐ b. doesn't change the list
- ☐ c. extends the list
- ☐ d. shortens the list

33. After execution of the following snippet, the sum of all vals elements will be equal to:

```
vals = [0, 1, 2]  
vals.insert(0,1)  
del vals[1]
```

- ☐ a. 5
- ☒ b. 4
- ☐ c. 3
- ☐ d. 2

34. Take a look at the snippet, and choose the true statement:

```
nums = [1,2,3]  
vals = nums  
del vals[1:2]
```

- ☐ a. nums is longer than vals
- ☐ b. the snippet will cause a runtime error
- ☐ c. vals is longer than nums
- ☒ d. nums and vals are of the same length

35. Which of the following sentences is true?

```
nums = [1,2,3]  
vals = nums[ -1: -2]
```

- ☒ a. nums is longer than vals
- ☐ b. the snippet will cause a runtime error
- ☐ c. nums and vals are of the same length
- ☐ d. vals is longer than nums

36. What is the output of the following snippet?

```
l1 = [1,2,3]
l2 = []
for v in l1:
    l2.insert(0,v)
print(l2)
```

- a. [3,3,3]
- b. [1,2,3]
- c. [3,2,1]**
- d. [1,1,1]

37. What is the output of the following snippet?

```
l1 = [1,2,3]
for v in range(len(l1)):
    l1.insert(1,l1[v])
print(l1)
```

- a. [1, 2, 3, 1, 2, 3]
- b. [1, 1, 1, 1, 2, 3]**
- c. [3, 2, 1, 1, 2, 3]
- d. [1, 2, 3, 3, 2, 1]

38. How many elements does the L list contain?

```
L = [i for i in range(-1,2)]
```

- a. one
- b. two
- c. three**
- d. four

39. What is the output of the following snippet?

```
T = [[3-i for i in range (3)] for j in range (3)]
s = 0
for i in range(3):
    s += T[i][i]
print(s)
```

- a. 6**
- b. 2
- c. 4
- d. 7

40. What is the output of the following snippet?

```
L = [[0, 1, 2, 3] for i in range(2)]  
print(L[2][0])
```

- a. 2
- b. 1
- ☒ c. the snippet will cause a runtime error
- d. 0

41. Which of the following lines properly starts a parameterless function definition?

- ☒ a. def fun():
- b. def fun:
- c. fun function():
- d. function fun():

42. A function defined in the following way:

```
def function(x=0):  
    return x
```

- a. must be invoked with exactly one argument
- b. may be invoked with any number of arguments (including zero)
- c. must be invoked without arguments
- ☒ d. may be invoked without any argument, or with just one

43. A built in function is a function which:

- a. is hidden from programmers
- b. has to be imported before use
- c. has been placed within your code by another programmer
- ☒ d. comes with Python, and is an integral part of Python

44. The fact that tuples belong to sequence types means:

- a. they can be modified using the del instruction
- b. they are actually lists
- c. they can be extended using the .append() method
- ☒ d. they can be indexed and sliced like lists

45. What is the output of the following snippet?

```
def f(x):  
    if x == 0:  
        return 0  
    return x + f(x - 1)  
  
print(f(3))
```

- a. the code is erroneous
- b. 6**
- c. 3
- d. 1

46. What is the output of the following snippet?

```
def fun(x):  
    x += 1  
    return x  
  
x = 2  
x = fun(x+1)  
print(x)
```

- a. 5
- b. 3
- c. 4**
- d. the code is erroneous

47. What code would you insert into the commented line to obtain the output that reads:

```
a  
b  
c  
  
dct = { }  
lst = ['a', 'b', 'c', 'd']  
for i in range(len(lst) - 1):  
    dct[lst[i]] = ( lst[i], )  
for i in sorted(dct.keys()):  
    k = dct[i]  
    # insert your code
```

- a. print(k["0"])
- b. print(k['0'])
- c. print(k)
- d. print(k[0])**

48. The following snippet:

```
def func(a,b):  
    return a ** a  
  
print(func(2))
```

- a. will output 4
- b. will output 2
- c. will return None
- ☒ d. is erroneous

49. The following snippet:

```
def func1(a):  
    return a ** a  
def func2(a):  
    return func1(a)*func1(a)  
print(func2(2))
```

- a. will output 2
- b. is erroneous
- c. will output 4
- ☒ d. will output 16

50. Which of the following lines properly starts a function using two parameters, both with zeroed default values?

- a. fun fun(a=0,b):
- ☒ b. def fun(a=0,b=0):
- c. fun fun(a,b=0):
- d. def fun(a=b=0):

51. Which of the following statements is false?

- a. The None value may not be used outside functions
- ☒ b. The None value cannot be used as an argument of arithmetic operators
- c. The None value can be compared with variables
- d. The None value can be assigned to variables

52. What is the output of the following snippet?

```
def fun(x):  
    if x % 2 == 0:  
        return 1  
    else:  
        return  
  
print(fun(fun(2)) + 1)
```

- a. 1
- b. 2
- c. None
- ☒ d. the code will cause a runtime error

53. What is the output of the following snippet?

```
def fun(x):  
    global y  
    y = x * x  
    return y  
  
fun(2)  
print(y)
```

- ☒ a. 4
- b. None
- c. the code will cause a runtime error
- d. 2

54. What is the output of the following snippet?

```
def any():  
    print(var + 1, end='')  
var = 1  
any()  
print(var)
```

- a. 11
- b. 12
- ☒ c. 21
- d. 22

55. Assuming that tuple is a correctly created tuple, the fact that tuples are immutable means that the following instruction:

```
tuple[1] = tuple[1] + tuple[0]
```

- a. can be executed if and only if the tuple contains at least two elements
- b. is fully correct
- c. is illegal**
- d. may be illegal if the tuple contains strings

56. What is the output of the following snippet?

```
list = ['Mary', 'had', 'a', 'little', 'lamb']
def list(L):
    del L[3]
    L[3] = 'ram'
print(list(list))
```

- a. ['Mary', 'had', 'a', 'ram']
- b. the snippet is erroneous return missing**
- c. ['Mary', 'had', 'a', 'little', 'lamb']
- d. ['Mary', 'had', 'a', 'lamb']

57. What is the output of the following snippet?

```
def fun(x,y,z):
    return x+2*y+3*z

print(fun(0,z=1,y=3))
```

- a. 0
- b. the snippet is erroneous
- c. 3
- d. 9**

58. What is the output of the following snippet?

```
def fun(inp=2,out=3):
    return inp * out
print(fun(out=2))
```

- a. 2
- b. 4**
- c. 6
- d. the snippet is erroneous

59. What is the output of the following snippet?

```
dct = { 'one':'two', 'three':'one', 'two':'three' }  
v = dct['one']  
for k in range(len(dct)):  
    v = dct[v]  
print(v)
```

- a. three
- b. one
- ☒ c. two
- d. ('one', 'two', 'three')

60. What is the output of the following snippet?

```
tup = (1, 2, 4, 8)  
tup = tup[1: -1]  
tup = tup[0]  
print(tup)
```

- a. the snippet is erroneous
- b. (2,)
- ☒ c. 2
- d. (2)