

1. What is the output of the following program?

```
1: randomList = [0, 2, 'a']
2: for entry in randomList:
3:     print("The entry is", entry)
4:     r = 1/int(entry)
5:     print("1/",entry,"is",r)
```

- 1) The program prints 1/0 is infinite and finishes without any error.
- 2) The program prints 1/0 is ZeroDivisionError and finishes without any error.
- 3) The program prints The entry is 0 and finishes without any error.
- 4) The program prints The entry is 0 and finishes with an ZeroDivisionError error.
- 5) The program does not print anything and finishes without any error.

2. What is the output of the following code fragment?

```
1: a = ['Hello',' World', 2017]
2: print(a[0])
```

- 1) Hello World 2017
- 2) Hello
- 3) H
- 4) HW2
- 5) None of the above choices is correct.

3. What is the output of the following code fragment?

```
1: for x in [1,2,3]:
2:     print(x, end=' ')
```

- 1) xxx
- 2) 123
- 3) H
- 4) HW2
- 5) None of the above choices is correct.

Use the following code fragment to answer the next 3 questions.

```
1: def magic(str):
2:     length = len(str)
3:     if length > 2:
4:         tmp = ''
5:         for i in range(length-3,length):
6:             tmp = tmp+str[i]
7:             if tmp == 'ing':
8:                 str = str+'ly'
9:             else:
10:                str = str+'ing'
11:     return str
12: print(magic(input()))
```

4. What is the output when the user's input string is "Ting"?

- 1) Ting
- 2) Tly
- 3) Tingly
- 4) gniT
- 5) None of the above choices is correct.

5. What is the output when the user's input string is "ab"?

- 1) ab
- 2) ably
- 3) abingly
- 4) baing
- 5) None of the above choices is correct.

6. What is the output when the user's input string is "Love"?

- 1) Loving
- 2) Lovely
- 3) Love
- 4) evoLing
- 5) None of the above choices is correct.

7. What is the output of the following code fragment?

```
1: def num(seq):
2:     x = seq[0]
3:     for a in seq:
4:         if a < x:
5:             x = a
6:     return x
7: print(num([1, 2, -8, 0]))
```

- 1) 1
- 2) 2
- 3) -8
- 4) 0
- 5) None of the above choices is correct.

8. What is the correct Python expression to represent the following mathematical expression?

$$\frac{(6+3) \times 2a}{5} - m$$

- 1) (6 + 3) * 2a / 5 - m
- 2) 6 + 3 * 2 * a / 5 - m
- 3) ((6 + 3) * 2a / 5) - m
- 4) (6 + 3 * 2 * a) / 5 - m
- 5) (6 + 3) * 2 * a / 5 - m

9. Which choice correctly calculates the expression:

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3}$$

1)

```
total = 0
for x in [1, 2, 3]:
    total = total + 1/x
print(total)
```

2)

```
total = 0
for x in range(3):
    total = total + 1/x
print(total)
```

3)

```
a = [1, 2, 3]
for x in a:
    x = 1/x
    total = sum(a)
print(total)
```

4)

```
a = [1, 2, 3]
a = 1/a
total = sum(a)
print(total)
```

5)

```
a = [1, 2, 3]
for x in range(3):
    a[x] = a[x]/x
    total = sum(a)
print(total)
```

10. What is the output of the following code fragment?

```
1: n = 4
2: i = 0
3: while i < n:
4:     print(i,n-i)
5:     i = i+1
```

1) 0 4
1 3
2 2
3 1

2) 0 4
1 3
2 2
3 1
4 0

3) 0 0
1 1
2 2
3 3
4 4

4) 4 0
3 1
2 2
1 3

5) 0 1
1 2
2 3
3 4
4 5

11. What should the user input if he/she wants the following code to print **This is it.**

```
1: x = int(input())
2: y = int(input())
3: if x > y:
4:     print("No, no.")
5: else:
6:     if x < 15:
7:         print("This is it.")
8:     else:
9:         print("Na, Na.")
```

1) 10
7

2) 100
90

3) 10
10

4) 50
55

5) 20
40

12. Which choice is NOT a valid statement?

- 1) `is-prime = False`
- 2) `Class = int("10")`
- 3) `function = input("Enter number: ")`
- 4) `string = "hello"*3`
- 5) All of the above choices

13. Consider the following statements:

A: `print(2*3)`

B: `print(2**3)`

C: `print(power(2,3))`

D: `import math`
`print(math.power(2,3))`

E: `import Math`
`print(Math.power(2,3))`

What is the one to compute “two, raised to the third power”, (i.e., 2^3)?

- 1) Only A
- 2) Only B
- 3) B, C, and D
- 4) B, C, and E
- 5) None of the above choices

14. What is the output of the following program if the user enters **190342** as an input?

```
1: num = input("Enter a number: ")
2: a = 0
3: b = 0
4: for i in range(len(num)):
5:     mod = num[i] % 2
6:     if mod > 0:
7:         a = a+1
8:     else:
9:         b = b+1
10: print (a,b)
```

- 1) 0 0
- 2) 3 3
- 3) 5 1
- 4) An error occurs at line 5.
- 5) Two errors occur at lines 4 and 5.

15. What is the output of the following program if the user enters **6** as an input?

```
1: my_list = [1,2,3,5,8,13,15,20,25,29]
2: n = int(input("Choose a number: "))
3: new_list = []
4: for i in my_list:
5:     if i < n:
6:         new_list.append(my_list[i])
7: print(new_list)
```

- 1) []
- 2) [1, 2, 3, 5]
- 3) [2, 3, 5, 13]
- 4) [1, 2, 3, 5, 8, 13]
- 5) The program has an error.

16. What is the output of the following program?

```
1: def list_call(lst):
2:     return [lst[1], lst[len(lst)-1]]
3:
4: lst = [1,2,3,4,5,6]
5: lst = list_call(lst)
6: print(lst)
```

- 1) []
- 2) 2, 6
- 3) [2, 6]
- 4) [[2, 6]]
- 5) The program has an error.

17. Suppose that we have a text file, containing 4 lines:

data.txt

```
Hello my
World !
Computer
Programming
```

What is the output of the following program?

```
1: lines = open('data.txt').read()
      .splitlines()
2: for i in range(len(lines)-1,0,-1):
3:     if i%2 == 0:
4:         print(lines[i])
5:     else:
6:         pass
```

- 1) Programming
World !
- 2) Computer
Hello my
- 3) Programming
World
my
- 4) Computer
World
- 5) Computer

18. We want to write a Python program to draw a triangle using * characters as shown below:

```
----*
---***
__*****
_*****
*****
```

Which are the correct expressions to be filled into the blanks (A) and (B) below?

```
1: for i in range(5):
2:     dashes = '-'*(____(A)____)
3:     stars = '*'*(____(B)____)
4:     print(dashes + stars)
```

- 1) (A): 5-i
(B): 2*i-1
- 2) (A): i
(B): 2*i
- 3) (A): 4-i
(B): 2*i+1
- 4) (A): i
(B): 2*i+1
- 5) (A): 4-i
(B): 2*(4-i)+1

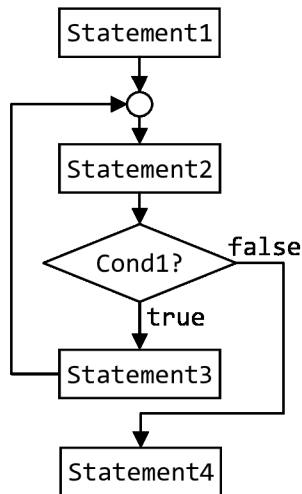
19. Let the list **X** be defined with the statement:

```
X = [1,2,3,4,5]
```

Which statement creates another list **Y** containing [2,6,10]?

- 1) Y = [x*2 for x in X]
- 2) Y = [x*2 for x in X if x%2 == 0]
- 3) Y = [x*2 for x in X if x%2 == 1]
- 4) Y = [x for x in X if x>2]
- 5) Y = [x for x in X*2 if x%2 == 1]

Use the flowchart below to answer the following 2 questions.



20. Which choice is correct about the flowchart?

- 1) **Statement2** gets executed only once.
- 2) **Statement2** gets executed at least once.
- 3) **Statement3** gets executed only once.
- 4) **Statement3** gets executed at least once.
- 5) **Statement4** gets executed only once.

21. Which Python code segment is corresponding to the flowchart?

1) `Statement1`
`Statement2`
`while Cond1:`
 `Statement3`
`Statement4`

2) `Statement1`
`Statement2`
`while not Cond1:`
 `Statement3`
`Statement4`

3) `Statement1`
`while True:`
 `Statement2`
 if `Cond1:`
 `break`
 `Statement3`
`Statement4`

4) `Statement1`
`while True:`
 `Statement2`
 if not `Cond1:`
 `break`
 `Statement3`
`Statement4`

5) `Statement1`
`Statement2`
if not `Cond1:`
 `Statement4`
else:
 `Statement3`

22. Given two sequences, $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_n)$. The mean squared error (MSE) between A and B is defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^n (a_i - b_i)^2$$

Suppose A and B are already defined as NumPy arrays, which expression correctly computes the MSE above?

- 1) `sum((A-B)**2)/n`
- 2) `sum(A**2-B**2)/n`
- 3) `(1/n)*sum(A-B)**2`
- 4) `(1/n)*(sum(A)-sum(B))**2`
- 5) `(sum(A-B)/n)**2`

23. Consider the following Python shell session:

```
>>> import numpy as np
>>> data = np.loadtxt("data.txt", delimiter=",")
>>> data.size
6
>>> data.shape
(2, 3)
>>> data[1]
array([ 56., 27., 61.])
>>> data.T[1]
array([ 17., 27.])
```

Which contents of the file `data.txt` would give the above results?

1) `17,17,79`
`56,27,61`
`32,26,88`

2) `56,27,61`
`17,27,90`

3) `56,17`
`27,27`
`61,90`

4) `34,56`
`17,27`
`83,61`

5) `34,17,83`
`56,27,61`

24. Given a Python shell session:

```
>>> list(range(____(A)____))
[5, 3, 1, -1, -3, -5]
```

Which choice should be used to fill in the blank (A) to get the result shown?

- 1) 5,-5
- 2) 5,-5,-1
- 3) 5,-6,2
- 4) 5,-5,-2
- 5) 5,-6,-2

25. What is the output of the following program.

```
1: a = [1,2,3]
2: b = a + 4
3: print(b[2])
```

- 1) 2
- 2) 3
- 3) 4
- 4) 6
- 5) An error occurred.

26. What is the output of the following program.

```
1: a = [1,0,2,0,3,0,4,0]
2: i = 1
3: sum = 0
4: while i < 5:
5:     sum = sum+a[i]
6:     if i > 3:
7:         break
8:     i = i+1
9: print(sum)
```

- 1) 2
- 2) 3
- 3) 5
- 4) 6
- 5) 9

27. What is the output of the following program.

```
1: a = [[0,1,2],[0,1,2,3],[0,1,2,3,4]]
2: sum = 0
3: for i in range(1,3):
4:     for j in range(2,4):
5:         sum = sum+a[i][j]
6: print(sum)
```

- 1) 6
- 2) 9
- 3) 10
- 4) 16
- 5) Error occurred. Index out of range.

28. What choice filled in blank (A) will make the program output 6?

```
1: def M(x):
2:     sum = 0
3:     for i in range(len(x)):
4:         sum = sum+i
5:     return sum
6:
7: a = [[0],[1],[1,2],[1,2,3]]
8: ____ (A) ____
```

- 1) print(M(a))
- 2) print(M(a[1]))
- 3) print(M(a[2]))
- 4) print(M(a[3]))
- 5) print(M(a[4]))

Use the following code fragment to answer the next 2 questions.

A:

```
sum = 0
for i in range(0,11):
    sum = sum+i
print(sum)
```

B:

```
sum = 0
for i in range(1,10,1):
    sum = sum+i
print(sum)
```

C:

```
sum = 0
for i in range(10,0,-1):
    sum = sum+i
print(sum)
```

D:

```
sum = 0
for i in range(11,1,-1):
    sum = sum+i
print(sum)
```

29. Which pair of programs give the same output?

- 1) A and C
- 2) A and D
- 3) B and C
- 4) B and D
- 5) All code fragments give different outputs.

30. Which pair of programs run with the same number of iterations in the for loop (จำนวนรอบในคำสั่ง for)?

- 1) A and B
- 2) B and C
- 3) C and D
- 4) D and A
- 5) None of the above pairs.

31. What is the print out of the following program?

```
1: name='Good Morning'
2: print(name[-len(name)])
```

- 1) G
- 2) M
- 3) d
- 4) g
- 5) None of the above

32. What is the print out of the following program?

```
1: import numpy as np
2: a = np.array([1,2,3,5,8])
3: b = np.array([0,3,4,2,1])
4: c = a+b
5: c = c*a
6: print(c[2])
```

- 1) 7
- 2) 10
- 3) 12
- 4) 21
- 5) 28

33. What is the print out of the following program?

```
1: import numpy as np
2: a = np.array([0,1,0])
3: a = a+3
4: b = a+3
5: print(a[1]+b[2])
```

- 1) 2
- 2) 8
- 3) 10
- 4) 14
- 5) None of the above

34. What is the print out of the following program?

```
1: numbers = [1,2,3,4]
2: numbers.append('01204111')
3: print(len(numbers))
```

- 1) 4
- 2) 5
- 3) 8
- 4) 12
- 5) None of the above

35. What needs to be filled in blank (A) at line 6 to get the output as shown at line 8?

```
1: >>> table
2: array([[ 1,  2,  3,  4],
3:        [ 5,  6,  7,  8],
4:        [ 9, 10, 11, 12],
5:        [13, 14, 15, 16]])
6: >>> table2 = (A)
7: >>> table2
8: [1, 6, 11, 16]
```

- 1) [table[i][i] for i in range(len(table))]
- 2) [table[i][i] for i in range(len(table[0]))]
- 3) [table[i][i] for i in range(len(table.shape))]
- 4) [table[i][i] for i in range(len(table.shape[0]))]
- 5) Both 1) and 2) are correct.

36. Consider the following list assignment:

```
a = [[1,2],[4,5],[7,8]]
```

which choice INCORRECTLY refers to a member of the above list?

- 1) a[0][0]
- 2) a[1][1]
- 3) a[1][2]
- 4) a[2][1]
- 5) More than one of the above choices

37. Consider the following array assignment:

```
import numpy as np
a = np.array([[1,2,3],[4,5,6]])
```

which choice INCORRECTLY refers to a member of the above array?

- 1) a[1][1]
- 2) a[1][2]
- 3) a[1][3]
- 4) a[2][2]
- 5) More than one of the above choices

38. Consider the following list assignment:

```
a = [1, 5, 8, 16, 5, 7, 2, 3]
```

which choice gives a different result from the others?

- 1) `print(a[1])`
- 2) `print(a[3]-a[len(a)-1])`
- 3) `print(a[-4])`
- 4) `print(a[3]-11)`
- 5) `print(a[int(4*(1/2))-1])`

39. Which '+' sign has a different meaning from the others?

- 1) `a = 1 + 2`
- 2) `b = 2.0 + 4.0`
- 3) `c = int('3') + int('6')`
- 4) `d = '4' + '8'`
- 5) `e = float('5') + float('10')`

40. Suppose user enters the values as in the following command lines.

```
n = int(input('Enter n: '))
m = int(input('Enter m: '))
```

Which choice gives the result of the following math series?

$$total = \sum_{i=m}^n \frac{1}{i}$$

- 1)

```
total = 0
for i in range(m,n+1):
    total = total + 1/i
```
- 2)

```
total = sum(
    [1/x for x in range(m,n+1)])
```
- 3)

```
nlist = list(range(m,n+1))
alist = [1/x for x in nlist]
total = sum(alist)
```
- 4)

```
import numpy as np
nlist = np.array(range(m,n+1))
total = sum(1/nlist)
```
- 5) All of the above answers are correct.

41. What is the output of the following code fragment?

```
1: list1 = [1,2,3]
2: list2 = 'Num'
3: print(list1+list2)
```

- 1) `[1,2,3, 'Num']`
- 2) `[1,2,3, 'N', 'u', 'm']`
- 3) `[1,2,3]`
- 4) `'Num'`
- 5) Compilation error

Use the following code fragment to answer the next 4 questions.

Complete the fragment of this program which aims to find the greatest common divisor (หารร่วมมาก) or gcd (ท.ร.ม.) of 8 and 6. For an example, the gcd of 12 and 8 is 4.

```
1: def find_gcd(n1,n2):
2:     found = False
3:     for n in range(____(A)____):
4:         if ____ (B) ____:
5:             ____ (C) ____
6:         if found:
7:             print(f"{n}")
8:
9: find_gcd(8,6)
```

42. Which choice should be filled in the blank (A)?

- 1) `n2,1,-1`
- 2) `n1,1`
- 3) `1,n2`
- 4) `1,n1`
- 5) `n1,0,-1`

43. Which choice should be filled in the blank (B)?

- 1) `n1%n2 == 0`
- 2) `n1%n == 0 and n2%n == 0`
- 3) `n2%n1 == 0`
- 4) `n2%n1 == 0 or n1%n2 == 0`
- 5) `n%n1 == 0 and n%n2 == 0`

44. Which choice should be filled in the blank (C)?

- 1) `break`
`found = false`
- 2) `break`
- 3) `found = True`
- 4) `found = True`
`break`
- 5) `pass`

45. What is the correct result printed on the screen?

- 1) 1
- 2) 2
- 3) 3
- 4) 4
- 5) 6

46. Which choice is NOT a valid expression?

- 1) `int(10)`
- 2) `float(10)`
- 3) `pow(10)`
- 4) `abs(10)`
- 5) All the above choices are invalid expressions.

47. Which choice gives a different answer from the others?

- 1)

```
word = "HelloWorld"
for ch in range(10):
    print(word[ch])
```
- 2)

```
word = "HelloWorld"
for ch in range(len(word)):
    print(word[ch])
```
- 3)

```
word = "HelloWorld"
for ch in word:
    print(word[ch])
```
- 4)

```
word = "HelloWorld"
for ch in word:
    print(ch)
```
- 5) All the above choices give the same answer.

48. Which choice has the same meaning (can work in the same manner) as the following "for loop"?

- 1:

```
for i in range(1,10,3):
    print(i)
```
- 1)

```
while True:
    i = 1
    if i > 10:
        break
    print(i)
    i = i + 3
```
- 2)

```
i = 1
while True:
    if i > 10:
        break
    print(i)
    i = i + 3
```
- 3)

```
while i < 10:
    i = 1
    print(i)
    i = i + 3
```
- 4)

```
i = 1
while i < 10:
    print(i)
    i = i + 3
```
- 5) None of the above choices is correct.

49. Which choice is the correct definition of function `circle_area()` so that the code fragment below can work properly?

```
1: import math
2: r = float(input("Enter a radius"))
3: area = circle_area(r)
4: print(f"Area is {area:.2f}")
```

- 1)

```
def circle_area(r):
    print(float(math.pi*r*r))
```
- 2)

```
def circle_area(r):
    print(math.pi*math.pow(r,2))
```
- 3)

```
def circle_area(r):
    return(int(math.pi*r*r))
```
- 4)

```
def circle_area(r):
    return(math.pi*r*r)
```
- 5) None of the above choices is correct.

50. What is the output of the following program?

```
1: def selector(inp, x, offset):
2:     str_len = len(inp)
3:     result = 0
4:     while offset < str_len:
5:         if inp[offset] == x:
6:             result = offset
7:             offset = offset + 1
8:     return result
9:
10: inp = "HelloWorld"
11: a = selector(inp, 'l', 3)
12: print(a)
```

- 1) 0
- 2) 2
- 3) 3
- 4) 8
- 5) 10

51. Which is the output of the following commands?

```
1: n = 0
2: for i in range(1,3):
3:     for j in range(i):
4:         n = n+j
5:     n = n-i
6: print(n)
```

- 1) 0
- 2) 1
- 3) -2
- 4) 2
- 5) No correct answer

52. Consider the following variable declaration:

```
a = [2,5,11,13,0,-5]
```

What is the value of `min(a)+max(a)+sum(a)`?

- 1) 8
- 2) 13
- 3) 26
- 4) 34
- 5) 44

53. Which is the output of the following program?

```
1: def add(a,b):
2:     a = a+10
3:     b = b+20
4:     i = 2
5:     j = 2
6:     add(i,j)
7:     print(i+j)
```

- 1) 4
- 2) 12
- 3) 14
- 4) 34
- 5) Compilation error

54. Which is the output of the following program?

```
1: a = [ 9, 8, 2, 5, 4 ]
2: b = []
3: for i in range(5):
4:     b.append(a[i]%5)
5: print(b)
```

- 1) [1.8, 1.6, 0.4, 1.0, 0.8]
- 2) [4, 3, 2, 0, 4]
- 3) [1, 1, 0, 1, 0]
- 4) [4, 3, -3, 0, -1]
- 5) No correct answer

55. Which is the output of the following program?

```
1: sum = 0
2: num1 = 25
3: num2 = 10
4: def find_sum():
5:     sum = num1+num2
6: find_sum()
7: print(sum)
```

- 1) 0
- 2) 10
- 3) 25
- 4) 35
- 5) No correct answer

56. Which is the output of the following program?

```
1: sum = 0
2: num1 = 25
3: num2 = 10
4: def find_sum():
5:     sum = num1+num2
6:     print(sum)
7: find_sum()
```

- 1) 0
- 2) 10
- 3) 25
- 4) 35
- 5) No correct answer

57. Which is the summation of members in each sub-list of list `a` to store in list `b`, and the output is as follows:

```
[3, 5, 7]
```

```
1) a = [[1,2],[2,3],[3,4]]
   b = []
   for i in range(3):
       x = 0
       for j in range(2):
           x = x+a[i][j]
       b.append(x)
   print(b)
```

```
2) a = [[1,2],[2,3],[3,4]]
   for i in range(3):
       x = 0
       for j in range(2):
           x = x+a[i][j]
       b.append(x)
   print(b)
```

```
3) a = [[1,2],[2,3],[3,4]]
   b = []
   for i in range(2):
       x = 0
       for j in range(3):
           x = x+a[i][j]
       b.append(x)
   print(b)
```

```
4) a = [[1,2],[2,3],[3,4]]
   b = []
   for i in range(3):
       x = 0
       for j in range(2):
           x = x+a[j][i]
       b.append(x)
   print(b)
```

- 5) No correct answer

58. How many iterations (จำนวนรอบ) will the following code fragment run before it stops?

```
1: while True:
2:     n = int(input('Enter an integer: '))
3:     if n < 0:
4:         break
5:     print('Binary number is',dec_to_bin(n))
6:     print('Bye!')
```

- 1) Infinite iterations: it will never stop because it's a forever Loop.
- 2) Ten iterations: **dec_to_bin()** means it will only accept 10 integer values.
- 3) Zero iteration: the **break** command will always stop the whole program.
- 4) Unknown: it depends on how **dec_to_bin()** handles integer values.
- 5) Unknown: it depends on the user's input.

59. What does the following code fragment do?

```
1: full_score = 100
2: raw_score = [31,56,73,49]
3: p = [x/full_score for x in raw_score]
4: NF = [1 for x in p if x >= 0.5]
5: print('The number is', len(p)-len(NF))
```

- 1) Find the maximum number of digits in the student's scores.
- 2) Find the minimum number of digits in the student's scores.
- 3) Find the average number of digits in the student's scores.
- 4) Count the number of students who score less than 50%.
- 5) Count the number of students who score 50% or more.

60. What does the following code fragment do?

```
1: a = []
2: b = []
3: raw_score = [31,56,73,49]
4: for x in raw_score:
5:     if (x%2 == 0):
6:         a.append(x)
7:     else:
8:         b.append(x)
9: print(a)
   print(b)
```

- 1) Separate students with score less than 50% into one list and the others into another list.
- 2) Separate students with odd raw score into one list and the others into another list.
- 3) Separate students in the front half into one list and the others into another list.
- 4) Separate the No. 1 student into the first list, the No. 2 student into the second list, the No. 3 student into the first list, the No. 4 student into the second list, etc.
- 5) Separate student(s) with maximum score into one list and the others into another list.