1. What is the output of the following program?

```
 1: tlist = [
 2:     [1, 2],
 3:     [3, 4, 5],
 4:     [6],
 5:     [7, 8],
 6:     [9, 10, 11]
 7: ]
 8: clist = [len(r) for r in tlist]
 9: mcol = max(clist)
10: u = [
11:     r[-1]
12:     for r in tlist if len(r) == mcol
13: ]
14: print(u)
```

1) `[9, 10, 11]`
2) `[[5], [11]]`
3) `[2, 5, 6, 8, 11]`
4) `[5, 11]`
5) `[[3, 4, 5], [9, 10, 11]]`

2. What is the output of the following program?

```
1: s1 = '123'
2: s2 = s1
3: list1 = [1, 2, 3]
4: list2 = list1
5: s1 = '456'
6: list1 = [4, 5, 6]
7: print(s2, list2)
```

1) `123 [1, 2, 3]`
2) `123 [4, 5, 6]`
3) `456 [1, 2, 3]`
4) `456 [4, 5, 6]`
5) Nothing gets printed because there is an error when the program runs.

3. What is the output of the following program?

```
1: s1 = '123'
2: s2 = s1
3: list1 = [1, 2, 3]
4: list2 = list1
5: s1 = s1 + '4'
6: list1.append(4)
7: print(s2, list2)
```

1) `123 [1, 2, 3]`
2) `123 [1, 2, 3, 4]`
3) `1234 [1, 2, 3]`
4) `1234 [1, 2, 3, 4]`
5) Nothing gets printed because there is an error when the program runs.

4. What is the output of the following program?

```
1: s1 = '123'
2: s2 = s1
3: list1 = [1, 2, 3]
4: list2 = list1
5: s1[-1] = '4'
6: list1[-1] = 4
7: print(s2, list2)
```

1) `123 [1, 2, 3]`
2) `123 [1, 2, 4]`
3) `124 [1, 2, 3]`
4) `124 [1, 2, 4]`
5) Nothing gets printed because there is an error when the program runs.

5. Which choice is <u>true</u> for the following functions `xcolor()` and `ycolor()`?

```
def xcolor(code):
    if code == 0:
        color = 'black'
    if code == 1:
        color = 'red'
    if code == 2:
        color = 'blue'
    else:
        color = 'white'
    return color
```

```
def ycolor(code):
    if code == 0:
        color = 'black'
    elif code == 1:
        color = 'red'
    elif code == 2:
        color = 'blue'
    else:
        color = 'white'
    return color
```

1) For <u>every value</u> of **x**, the values returned from `xcolor(x)` and `ycolor(x)` are the same.
2) The values returned from `xcolor(0)` and `ycolor(0)` are different.
3) The values returned from `xcolor(2)` and `ycolor(2)` are different.
4) The values returned from `xcolor(3)` and `ycolor(3)` are different.
5) Choices 1), 2), 3), and 4) are all false.

6. Suppose we want to write a program that reads a text file and prints out only the longest lines of the file. For example, if the file contents are

```
12 45 78
123 56
abcde gh
123 5
```

The output will be

```
12 45 78
abcde gh
```

Which choice is a correct order of the following code segments **A**, **B**, **C**, **D**, and **E** that forms a complete program that does the task?

A:
```
maxlen = max([len(x) for x in llist])
```

B:
```
for line in llist:
    if len(line) == maxlen:
        maxlines.append(line)
```

C:
```
fname = input('Enter filename: ')
llist = open(fname).read().splitlines()
```

D:
```
for line in maxlines:
    print(line)
```

E:
```
maxlines = []
```

1) E, C, B, A, D
2) E, C, D, A, B
3) C, A, E, B, D
4) C, E, B, A, D
5) C, D, A, E, B

7. What is the output of the following program?

```
 1: xlist = [
 2:     ['Fictions', 'reveal'],
 3:     ['the', 'truth', 'that'],
 4:     ['reality', 'obscures']
 5: ]
 6: w = ''
 7: for x in xlist:
 8:     for c in x[0]:
 9:         w = w + c
10: print(w)
```

1) Ftr
2) Fictions
3) Fictionsreveal
4) Fictionsthereality
5) ['F', 't', 'r']

8. The function **arrow(n)** prints **n** lines that look like an arrow head. The following Python shell session shows the use of the function **arrow(n)**.

```
>>>arrow(6)
*
  *
    *
    *
  *
*
>>>arrow(5)
*
  *
    *
  *
*
>>>
```

Which choice is a correct order of the following code segments **A**, **B**, **C**, **D**, and **E** that forms a complete code for the function **arrow(n)**?

A:
```
def arrow(n):
```

B:
```
    if n%2 != 0:
        print(k*' ' + '*')
```

C:
```
    for i in range(k):
        print(i*' ' + '*')
```

D:
```
    k = n//2
```

E:
```
    for i in range(k-1, -1, -1):
        print(i*' ' + '*')
```

1) A, D, C, B, E
2) A, D, B, C, E
3) A, D, C, E, B
4) A, D, B, E, C
5) A, D, E, B, C

9. Given a **lst** variable, what is the value of **lst[lst[lst[lst[4]]]]**?

```
lst = [3, 5, 4, 2, 1, 0]
```

1) **1**
2) **2**
3) **3**
4) **4**
5) It generates an error because list cannot be referenced this way.

10. Which one is a <u>Python list</u> of integers from 1 to 10?

    1) `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`
    2) `range(1, 10)`
    3) `range(11)`
    4) `[1, 2, 3, ..., 10]`
    5) More than one correct answer.

11. What is NOT a property of the following array?

    ```
    import numpy as np
    A = np.array([0, 1, 2, 3, 4, 5])
    ```

    1) `A.len`
    2) `A.ndim`
    3) `A.shape`
    4) `A.size`
    5) All of the above is correct.

12. Which choice can be filled in the blank in order to print the number from 0 to 4 to the screen?

    ```
    1:  i = 0
    2:  for _____:
    3:      print(i)
    4:      i = i+1
    ```

    1) `c in "01234"`
    2) `c in "12345"`
    3) `n in range(1, 6)`
    4) `k in range(5)`
    5) All of the above can be used.

13. Which one does the exact same thing as the following **if**-statement?

    ```
    1:  n = int(input())
    2:  if n > 10:
    3:      print(n)
    ```

    1)
    ```
    n = int(input())
    while n > 10:
        print(n)
    ```

    2)
    ```
    n = int(input())
    while n > 10:
        print(n)
        break
    ```

    3)
    ```
    n = int(input())
    while true:
        print(n)
        break
    ```

    4)
    ```
    n = int(input())
    count = 0
    while n > 10 and count <= 1:
        print(n)
        count = count + 1
    ```

    5) More than one correct answer.

14. What is the value of **c**?

    ```
    1:  import numpy as np
    2:  a = np.array([1, 2, 3, 4])
    3:  b = np.array([4, 5, 6, 7])
    4:  c = a + b
    ```

    1) `[5, 7, 9, 11]`
    2) `[1, 2, 3, 4, 4, 5, 6, 7]`
    3) `[[1, 2, 3, 4], [4, 5, 6, 7]]`
    4) `[[1, 4], [2, 5], [3, 6], [4, 7]]`
    5) `[14, 25, 36, 47]`

15. Which choice correctly calculates the expression:

    $$f = \sum_{i=1}^{n} i^2$$

    1)
    ```
    f = 0
    for i in range(1,n+1):
        f = f + i**2
    ```

    2)
    ```
    f = sum([i**2 for i in range(1,n+1)])
    ```

    3)
    ```
    i2 = []
    for i in range(1,n+1):
        i2.append(i**2)
    f = sum(i2)
    ```

    4) Choices 1), 2), and 3) are correct.

    5) None of the above.

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

    ```
    1:  s = [1,2,3,4,5,6]
    2:  s = sum([x/2 for x in s])
    3:  print(s)
    ```

    1) `10.5`
    2) `11.5`
    3) `12.5`
    4) There is a syntax error.
    5) None of the above choices is correct.

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

    ```
    1:  s = ['Hello',' World']
    2:  my_sum = min([len(x) for x in s])
    3:  print(my_sum)
    ```

    1) `5`
    2) `6`
    3) `11`
    4) There is a syntax error.
    5) None of the above choices is correct.

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

```
1:  def myfind(s,t):
2:      index = 0
3:      r = []
4:      for i in s:
5:          if i == t[0]:
6:              r.append(index)
7:          index = index+1
8:      if len(r) == 0:
9:          print('Found nothing.')
10:     else:
11:         for j in r:
12:             print(j,end=' ')
13:
14: s = 'Que sera, sera!'
15: myfind(s,'sera')
```

1) sera sera
2) 4 10
3) Found nothing.
4) There is a syntax error.
5) None of the above choices is correct.

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

```
1:  s = [[75,34,55],[24,86,95]]
2:  my_sum = 0
3:  count = 0
4:  for r in range(len(s)):
5:      for c in s[r]:
6:          my_sum = my_sum+c
7:          count = count+1
8:  print(f'{my_sum/count}')
```

1) 61
2) 61.5
3) 62
4) There is a syntax error.
5) None of the above choices is correct.

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

```
1:  def print_above(s,v):
2:      for r in s:
3:          for c in r:
4:              if c > v:
5:                  print(c,end=' ')
6:
7:  s = [[75,84,64],[67,79,45],[58,74,79]]
8:  print_above(s,70)
```

1) 0 1 4 7 8
2) 64 67 45 58
3) 75 84 79 74 79
4) There is a syntax error.
5) None of the above choices is correct.

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

```
1:  s = [[75,84,64],[67,79,45],[58,74,79]]
2:  for r in range(len(s)):
3:      for i in range(len(s[r])):
4:          if s[r][i]<50:
5:              print(f'std#{r+1} ',end='')
6:              print(f'fails subj#{i+1}')
```

1) std#1 fails subj#2
2) std#2 fails subj#3
3) std#3 fails subj#1
4) There is a syntax error.
5) None of the above choices is correct.

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

```
1:  a = [1,2,3]
2:  b = [4,5,6]
3:  a.append(99)
4:  a.append(b)
5:  print(a,b)
```

1) [1, 2, 3, 99, 4, 5, 6]
2) [1, 2, 3, 99, [4, 5, 6]]
3) [1, 2, 3, 99, [4, 5, 6]] [4, 5, 6]
4) There is a syntax error.
5) None of the above choices is correct.

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

```
1:  def check(s):
2:      m = len(s)
3:      while True and False or True:
4:          m = m-1
5:          if m >= 0:
6:              if s[m] < 0:
7:                  print(f'{s[m]} ',end='')
8:                  break
9:
10: s = [-23,45,16,-7,12,-9,56]
11: check(s)
```

1) -23 -7 -9
2) -7
3) -23
4) -9
5) There is a syntax error.

24. What is NOT a valid variable name in Python?

1) _123
2) Dan_Adam_1
3) 9Iron_Man
4) _True_
5) _

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

```
1: Y = 5**1**5*2
2: print(f"{Y}",end=' ')
```

1) 10
2) 25
3) 50
4) 125
5) {Y}

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

```
1: if (5 > 9.5) and (9.5 < -9):
2:     print("Thanos",end="")
3: elif (9 > 0) or False:
4:     print("IronMan",end="")
5: else:
6:     print("CaptainAmerica",end="")
7: if not (10 > 6) or not (-6 < 10):
8:     print("Avengers",end="")
```

1) Thanos
2) IronMan
3) Avengers
4) CaptainAmerica
5) IronManAvengers

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

```
1: def f(v=2,t=1):
2:     return 2*v**t
3: def g(x=2):
4:     return f(x)
5: print(f()+g(x=3)+f(v=3))
```

1) 4
2) 8
3) 12
4) 16
5) 20

28. How many times is line 6 executed when the code runs?

```
1: i = 5
2: j = 4
3: for x in range(j):
4:    for y in range(i):
5:        if (y > 2):
6:            print("H")
7:        else:
8:            print("M")
9: print("B")
```

1) 4
2) 8
3) 12
4) 16
5) 20

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

```
1: M = "Let us sacrifice our today so"
2: M = M+" that our children can have"
3: M = M+"a better tomorrow."
4: print(len(M.split("et")))
```

1) 2
2) 3
3) 8
4) 13
5) 15

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

```
1: a = list(range(5))
2: b = list(range(4))
3: c = []
4: for x in a:
5:     c.append(x**1)
6: for x in b:
7:     c.append(x+1)
8: print(sum(c))
```

1) 12
2) 16
3) 20
4) 30
5) 40

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

```
1: mytxt = "rrrbbrbcccb"
2: i = [x for x in "mytxt"]
3: b = [2 for x in i if x == "b"]
4: r = [3 for x in "mytxt" if x == "r"]
5: k = [4 for x in "mytxt" if x == "c"]
6: print(sum(b)+sum(r)+sum(k))
```

1) 0
2) 16
3) 32
4) 64
5) Syntax errors

32. Which choice gives the different output?

1)
```
for i in range(1,10,2):
    print(i)
```

2)
```
for i in range(5):
    print((i+1)*2-1)
```

3)
```
for i in range(0,9,2):
    print(i+1)
```

4)
```
for i in range(1,5):
    print(2*i-1)
```

5) All of the above choices print the same output.

33. Prime number is a number that is only divisible by itself and 1 (except 1 is not prime number). What should be filled in the blanks **(A)** and **(B)** to make the function **is_prime()** correctly check whether a positive integer **n** is a prime number?

```
1:  import sys
2:  def is_prime(n):
3:      result = True
4:      if n <= 1:
5:          result = False
6:      else:
7:          i = 2
8:          while ___(A)___:
9:              if n % i == 0:
10:                 result = False
11:                 break
12:             ___(B)___
13:     return result
```

1) A: i > n
   B: i = i + 1
2) A: i < n
   B: result = True
3) A: i < n
   B: n = n - 1
4) A: i < n
   B: i = i + 1
5) A: i < n
   B: sys.exit()

34. Suppose that **lst** is a list variable containing 5 elements. Which statement can cause an error?

1) print(lst[5])
2) lst[0] = 'HELLO'
3) x = f'{lst[-5]}'
4) y = lst[len(lst)-1]
5) None of the above choices.

35. Which line of the following code fragment can cause an error?

```
1:  scores = []
2:  while True:
3:      n = input('Enter score: ')
4:      if n == '':
5:          break
6:      scores.append(n)
7:  if len(scores) > 0:
8:      avg = sum(scores)/len(scores)
9:      print('Average score: {avg:.2f}')
```

1) line 1
2) line 3
3) line 6
4) line 7
5) line 8

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

```
1:  word = 'Easy!!'
2:  result = ''
3:  for i in range(0,len(word)):
4:      for j in range(0,i):
5:          char = str(j)
6:          result = result + char
7:  print(result)
```

1) 01020304
2) 0102030405
3) 01012012301234
4) 001012012301234
5) 001012012301234012345

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

```
1:  text = 'hello world'
2:  result = ''
3:  for i in range(1,len(text),2):
4:      result = result + text[i]
5:  print(result)
```

1) hello world
2) hlowrd
3) el ol
4) ello world
5) The code fragment has an error.

38. What should be filled in the blanks **(A)** and **(B)** to make the following code fragment produce the output as shown below?

```
1:  w = input('Enter word: ')
2:  s = ' '
3:  ___(A)___
4:  ___(B)___
```

Sample output:

```
Enter word: word
   w
  o
 r
d
```

```
Enter word: lucky
    l
   u
  c
 k
y
```

1) A: for i in range(len(w)-1,-1,-1):
   B: print(f'{s*i}{w[i]}')
2) A: for i in range(len(w)-1,-1,-1):
   B: print(f'{s*i}{w[len(w)-i-1]}')
3) A: for i in range(len(w)-1,-1,-1):
   B: print(f'{s*i}{w[len(w)-i]}')
4) A: for i in range(len(w)-1,-1,-1):
   B: print(f'{s*i}{w[len(w)-i+1]}')
5) A: for i in range(len(w)-1,-1,-1):
   B: print(f'{s*i}{w[len(w)]}')

39. Given the input file "**numbers.txt**" containing data as follows:

```
23.2
51.3

99
28.9

81
```

Which code fragment creates a list of

[23.2, 51.3, 99.0, 28.9, 81.0] named **num**?

1)
```
s = open('numbers.txt').read()
l = split(s)
num = [float(x) for x in l if x != '']
```

2)
```
s = open('numbers.txt').read()
l = s.splitlines()
num = [float(x) for x in l if x != '']
```

3)
```
s = open('numbers.txt').read()
l = [x.split() for x in s if x != '']
num = float(l)
```

4)
```
s = open('numbers.txt').read()
num = []
for x in s:
    if x != '':
        num.append(float(x.split()))
```

5) All of the above choices.

40. From the following program that reads the csv file "scores.csv" as follows:

scores.csv

```
2,3,8,7
5,5,6,7
9,2,4,8
```

```
filename = 'scores.csv'
    (A)
nrow = len(csvdata)
print(csvdata[nrow-1])
```

Which choice can be filled in the blank **(A)** to make the program run without error?

A:
```
import numpy as np
csvdata = np.loadtxt(filename,
            delimiter=",",ndmin=2)
```

B:
```
fp = open(filename)
text = fp.read()
lines = text.splitlines()
csvdata = []
for l in lines:
    l2 = l.split(",")
    csvdata.append(l2)
```

C:
```
text = open(filename).read()
lines = text.splitlines()
csvdata = [
    x.split(",")
    for x in lines
]
```

D:
```
csvdata = [
    x.split(",")
    for x in open(filename).read()
]
```

1) Only **A**
2) Only **A** and **B**
3) **A**, **B**, and **C**
4) **A**, **B**, **C**, and **D**
5) None of above choices is correct.

41. Which statement is NOT correct?
1) `For_class = 24**2 > 500`
2) `While = True or False`
3) `long_ranger = list(range(1,100,9))`
4) `list_p = [1,2.7,'KU']`
5) `combo = [['b','r','e','a','k'],else]`

42. Which program prints the string "**Enginboring**"?
1)
```
line = 'Engineering'
line[5-6] = 'bo'
print(line)
```

2)
```
line = 'Engineering'
line[5] = 'b'
line[6] = 'o'
print(str(line))
```

3)
```
line = 'Engineering'
fixline = [x for x in line]
fixline[5] = 'b'
fixline[6] = 'o'
print(str(fixline))
```

4)
```
line = 'Engineering'
fl = line.split('e')
fixline = fl[0]+'bo'+fl[-1]
print(str(fixline))
```

5)
```
line = 'Engineering'
i = 0
while i<len(line):
    if i == 5:
        print('b')
    elif i == 6:
        print('o')
    else:
        print(line[i])
    i = i + 1
```

43. What is the output of the following code fragment if **3** is given as the input?

```
1:  def f1(k):
2:      p = 0
3:      for i in range(k):
4:          j = i
5:          while True:
6:              p = p+i+j
7:              if j == 0:
8:                  break
9:              j = j - 1
10:     return p
11:
12: num = int(input())
13: ans = 0
14: for i in range(1,num+1):
15:     if i%2:
16:         ans = ans - 1
17:     else:
18:         ans = ans + f1(i)
19: print(f1(i),ans)
```

1) 1 12
2) 12 1
3) 12 3
4) 3 12
5) 12 9

44. If **s = 'Hi! How are you doing?'**, which choice gives **s1** as

[['Hi!', 'How', 'are', 'you', 'doing?']]?

A: `s1 = [s.split(" ")]`

B: `s1 = [s.split(" ") for s in s]`

C: `s1 = [x.split(" ") for x in s]`

1) Only **A**
2) Only **B**
3) Only **C**
4) Both **A** and **B**
5) Both **B** and **C**

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

```
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
```

45. Which choice gives the value of **21**?

1) `c[2]`
2) `c[0][2]`
3) `c[0][3]`
4) `c[0,3]`
5) Both 3) and 4) are correct.

46. What is the value of **c.shape**?

1) `(1,)`
2) `(1, 5)`
3) `(5)`
4) `(5,)`
5) None of the above

47. What needs to be filled in the blank **(A)** to get the output as shown on the last line?

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

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

48. Given two loops **A** and **B**,

Loop **A**:

```
num = int(input('Enter number: '))
while num < 0:
    num = int(input('Enter number: '))
```

Loop **B**:

```
1:  while True:
2:      num = int(input('Enter number: '))
3:      if _____:
4:          break
```

What is to put in line 3, so that both loops work the same?

1) `num != 0`
2) `num < 0`
3) `num <= 0`
4) `num > 0`
5) `num >= 0`

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

```
1:  s = 'hello'
2:  for i in range(0,len(s)):
3:      print(s[-i], end='')
```

1) holle
2) holl
3) olle
4) olleh
5) hello

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

```
1: x = [[6,5,4,3],[4,2,1,0],[9,7,8,6]]
2: s = 0
3: for index in range(len(x)):
4:     s = s + x[index][index+1]
5: print(s)
```

1) 12
2) 16
3) 19
4) 55
5) Error because index is out of range.

51. Which choice is the same output as the given code fragment?

```
1: for i in range(5):
2:     print(i)
```

1)
```
i = 0
while i < 5:
    print(i)
```

2)
```
i = 0
while i <= 5:
    print(i)
```

3)
```
i = 0
while i < 5:
    print(i)
    i = i + 1
```

4)
```
i = 0
while i <= 5:
    print(i)
    i = i+1
```

5)
```
i = 1
while i <= 5:
    print(i)
```

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

```
1:  def foo(nums):
2:      ans = []
3:      index = 0
4:      while index <len(nums):
5:          if ((nums[index]%2 == 1) and
6:              (sum(ans) < 10)):
7:              ans.append(nums[index])
8:          elif (sum(ans) >= 10):
9:              return ans
10:         index = index + 1
11:
12: nums = [2,5,3,3,8,1,-1]
13: print(foo(nums))
```

1) [2, 5, 3]
2) [2, 8]
3) [5, 3, 1]
4) [5, 3, 3]
5) [5, 3, 3, 1]

53. If user respectively enters 3, 4, 5, 2, -1, what is the output of the following code fragment?

```
1: w = int(input('Enter width: '))
2: area_list = []
3: while w > 0:
4:     h = int(input('Enter height: '))
5:     area_list.append(w*h)
6: print(area_list)
```

1) [12, 10]
2) [12, 15, 6]
3) [12, 15, 6, -3]
4) [12, 20, 10, -2]
5) This code fragment results in an infinite loop.

54. Given the following code fragment:

```
1:  def printline(c):
2:      for j in range(c):
3:          print('#', end='')
4:      print()
5:
6:  def printshape(b):
7:      for j in ___(A)___:
8:          printline(j)
9:
10: def printshapes(a,b):
11:     for j in range(a):
12:         printshape(b)
13:
14:     (B)
15:     (C)
16: printshapes(p,n)
```

what are to put in the blanks (A), (B), and (C) to get output like this?

```
###
##
#
###
##
#
###
##
#
###
##
#
```

1) A: range(1,b+1,1)
   B: n = 4
   C: p = 3
2) A: range(1,b+1,1)
   B: n = 3
   C: p = 4
3) A: range(b,0,-1)
   B: n = 4
   C: p = 3
4) A: range(b,0,-1)
   B: n = 3
   C: p = 4
5) A: range(b,1,-1)
   B: n = 4
   C: p = 3

55. Given the following code fragment:

```
1:  import math
2:  import numpy as np
3:
4:  x = [6,9,8,7]
5:  y = [16,9,4,1]
6:  z = [0,0,0,0]
7:  for i in range(len(x)):
8:      z[i] = 2*x[i]+math.sqrt(y[i])-3
9:  print(np.array(z))
10:
11: xa = np.array(x)
12: ya = np.array(y)
13: print(_____)
```

Which choice to replace in line 13 and gives the same result as line 9?

1) `2*xa + math.sqrt(ya) - 3`

2) `2*xa + np.sqrt(ya) - 3`

3) `2*x + math.sqrt(y) - 3`

4) `2*x + np.sqrt(ya) - 3`

5) Choices 2) and 4) are correct.

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

```
1:  greet = "Sawaddee"
2:  print(greet[4])
```

1) w

2) a

3) d

4) e

5) s

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

```
1:  t = [8, 9, 10, 11, 12]
2:  print(t[-4])
```

1) 8

2) 9

3) 10

4) 11

5) 12

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

```
1:  ls = [1,2,3,4,5,6,7]
2:  x, i = len(ls), 1
3:  sum = 0
4:  while i < x:
5:      sum = sum + ls[i]
6:      i = i + 2
7:  print(sum)
```

1) 12

2) 15

3) 16

4) 20

5) 27

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

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

1) 6

2) 9

3) 18

4) 21

5) 27

60. What should be filled in the blanks **(A)** and **(B)** to make the program produce the output shown below?

```
1:  n = int(input("Enter n: "))
2:  for i in ___(A)___:
3:      for j in ___(B)___:
4:          print('x', end='')
5:      print()
```

Sample output:

```
Enter n: 5
x
xx
xxx
xxxx
xxxxx
```

1) A: `range(n)`
   B: `range(i+1)`

2) A: `range(n+1)`
   B: `range(n+i)`

3) A: `range(n)`
   B: `range(i)`

4) A: `range(i+1)`
   B: `range(n+1)`

5) A: `range(n+i)`
   B: `range(i)`