

Read all questions and answers carefully! Do not make any assumptions about the code other than those that are clearly stated.

1. Consider we have two lists, X and Y, containing integer values. We want to calculate Z, which contains for each element x in X, number of elements in Y smaller than x. (e.g., X = [5, 20, 14, 1, 7], Y=[2, 12, 9, 5, 4, 16], Z=[2, 6, 5, 0, 3])

Which of the following Python function(s) will do this job properly?

```
I def funcA(X,Y):
    Z=[]
    for x in X:
        counter = 0
        for y in Y:
            if y<x:
                counter += 1
        Z.append(counter)
    return Z
```

```
II def funcB(X,Y):
    Z=[]
    for x in X:
        counter = 0
        for y in Y:
            if x<y:
                counter += 1
        Z.append(counter)
    return Z
```

```
III def funcC(X,Y):
    Z=[]
    for x in X:
        temp = []
        for y in Y:
            if y<x:
                temp.append(1)
            else:
                temp.append(0)
        Z.append(sum(temp))
    return Z
```

```
IV def funcD(X,Y):  
    Z=[]  
    for x in X:  
        temp = []  
        for y in Y:  
            if x<y:  
                temp.append(1)  
            else:  
                temp.append(0)  
        Z.append(sum(temp))  
    return Z
```

- A. I
- B. II
- C. I and III
- D. II and IV
- E. I, II and IV

2. Which of the following function(s) will return the second biggest number in the given list? Assume the input list contains unique integer elements and length of it is bigger than 2. (e.g., func ([10,15,13,24,39,7,13]) should return 24)

```
I def funcA(X):  
    X.sort()  
    return X[-2]
```

```
II def funcB(X):  
    X.sort()  
    biggest = X[-1]  
    X.remove(biggest)  
    return X[-1]
```

```
III def funcC(X):  
    X.sort()  
    biggest = X[-1]  
    del X[biggest]  
    return X[-1]
```

```
IV def funcD(X):  
    X.sort()  
    biggest = X[0]  
    X.remove(biggest)  
    return X[-1]
```

- A. I
- B. I and II
- C. II and IV
- D. I, II and IV
- E. I, II, III and IV

3. How many times will the word "Purdue" be printed in the following Python program?

```
def func(n):  
    if n > 1:  
        func(n-1);  
    for i in range(n):  
        print ("Purdue")
```

```
func(10)
```

- A. 36
- B. 45
- C. 55
- D. 78
- E. 85

4. What is the purpose of the following Python program?

```
def func(n):  
    if n == 1:  
        return 1  
    else:  
        return func(n-1) + n  
  
print func(5)
```

- A. Sums up all the numbers between 1, $n - 1$ (both inclusive)
- B. Sums up all the numbers between 1, n (both inclusive)
- C. Sums up all the numbers between 1, $n + 1$ (both inclusive)
- D. Sums up all the numbers between 2, n (both inclusive)
- E. Sums up all the numbers between 2, $n + 1$ (both inclusive)

5. Consider we have a list of lists to compute a series of plus and minus operations, called `computationTree`. The formal recursive definition of `computationTree` is the following,

`computationTree` = [integer] or [computationTree, operation, computationTree]
`operation` = '+' or '-'

Example: `X` = [[[2], '+', [12]], '+', [[4], '+', [[5], '-', [9]]]] represents $(2 + 12) + (4 + (5-9))$

Which of the following Python function(s) evaluates these kind of computation trees correctly?

```
I def evaluateA(X):
    if len(X) == 1:
        return X[0]
    else:
        if X[1] == '+':
            return evaluate(X[0]) + evaluate(X[2])
        elif X[1] == '-':
            return evaluate(X[0]) - evaluate(X[2])
```

```
II def evaluateA(X):
    if len(X) == 1:
        return X[0]
    else:
        if X[1] == '+':
            return evaluate(X[0]) + evaluate(X[1])
        elif X[1] == '-':
            return evaluate(X[0]) - evaluate(X[1])
```

```
III def evaluateA(X):
    if len(X) == 1:
        return X[0]
    else:
        if X[1] == '+':
            return evaluate(X[2]) + evaluate(X[0])
        elif X[1] == '-':
            return evaluate(X[2]) - evaluate(X[0])
```

- A. I
- B. III
- C. I and II
- D. II and III
- E. I, II and III

6. Which of the following Python programs use recursion?

```
I def factorial(n):  
    rVal = 1  
    for i in range(2, n+1):  
        rVal = rVal * i  
    return rVal
```

```
II def factorial(n):  
    if n < 2:  
        return 1  
    else:  
        return n * factorial(n-1)
```

```
III def fibonacci(n):  
    if n < 2:  
        return 1  
    else:  
        return fibonacci(n-1) + fibonacci(n-2)
```

```
IV def fibonacci(n):  
    fiboArray = [1,1]  
    i = 2  
    while i <= n:  
        fiboArray.append( fiboArray[i-1] + fiboArray[i-2] )  
        i += 1  
    return fiboArray[n]
```

- A. I
- B. II and III
- C. II and IV
- D. I and IV
- E. I, II and IV

7. The following Python function, `doWork()`, takes a string, `s`, as an input and returns an integer. Which of the following is the purpose of `func()`?

```
def doWork(s):  
    if len(a) < 3:  
        return 0  
    if a[:3] == 'abc':  
        return 1 + doWork( a[1:] )  
    else:  
        return 0 + doWork( a[1:] )
```

- A. Count number of "abc" in the given string `s`
- B. Count number of "a" characters in the string
- C. Return the index of first occurrence of "a" characters in the string
- D. Return the index of first occurrence of "abc" in given string `s`
- E. Check whether "abc" occurs in the given string `s`

8. is an optimization technique used primarily to speed up computer programs by storing the results of expensive function calls.
- A. Recursion
 - B. Iteration
 - C. Memoization
 - D. Object oriented programming
 - E. Graphics library

9. Consider the list $a = [10,9,4,22,18,3]$. Which of the following statements is equivalent to:

a. `remove(22)`

A. `del a[a.index(22)]`

B. `del a[a.index(3)]`

C. `remove a[a.index(22)]`

D. `remove a[a.index(3)]`

E. `a.del(indexof(22))`

10. What is the output of the following Python program?

```
def main():
    myCount = Count()
    times = 0

    for i in range(0, 50):
        increment(myCount, times)

    print("myCount.count =", myCount.count, "times =", times)

def increment(c, times):
    c.count += 2
    times += 2

class Count:
    def __init__(self):
        self.count = 0

main()
```

- A. myCount.count = 100 times = 100
- B. myCount.count = 100 times = 0
- C. myCount.count = 0 times = 100
- D. myCount.count = 50 times = 50
- E. myCount.count = 0 times = 0

11. Analyze the following code:

```
class A:  
    def __init__(self, s):  
        self.s = s  
  
    def printX(self):  
        print(self.s)
```

```
a = A()  
a.printX()
```

- A. The program has an error because class A does not have a constructor
- B. The program has an error because s is not defined in print(s)
- C. The program runs fine and prints nothing
- D. The program has an error because the constructor is invoked without an argument
- E. B and D

12. An object is an instance of a
- A. Program
 - B. Method
 - C. Class
 - D. data
 - E. Function

13. What is the output of the following Python program?

```
class A:
    def __init__(self, i = 3):
        self.i = i

class B(A):
    def __init__(self, j = 2):
        super().__init__()
        self.j = j

def main():
    b = B()
    print(b.i, b.j)
```

```
main()
```

A. 2 3

B. 0 2

C. 3 2

D. 3 3

E. 3 0

14. What is the output of the following Python program?

```
class A:
    def __init__(self):
        self.i = 2

    def m(self):
        self.i = 10

class B(A):
    def m(self):
        self.i += 2
        return self.i

def main():
    b = B()
    print(b.m())

main()
```

- A. 2
- B. 4
- C. 10
- D. 12
- E. The program has an error.

15. What is the output of the following Python program?

```
class A:
    def __str__(self):
        return "A"

class B(A):
    def __str__(self):
        return "B"

class C(B):
    def __init__(self):
        super().__init__()

def main():
    b = B()
    a = A()
    c = C()
    print(a, b, c)
```

main()

- A. A A A
- B. B B B
- C. A B A
- D. A B B
- E. B A A

16. is a template, blueprint, or contract that defines objects of the same type.
- A. class
 - B. An object
 - C. A method
 - D. A data field
 - E. A program

17. What is the output of the following Python program?

```
class A:
    def __init__(self):
        self.setI(20)
        print("i from A is", self.i)

    def setI(self, i):
        self.i = 2 * i;

class B(A):
    def __init__(self):
        super().__init__()

    def setI(self, i):
        self.i = 4 * i;
```

```
b = B()
a = A()
```

- A. i from A is 80
i from A is 40
- B. i from A is 20
i from A is 20
- C. i from A is 40
i from A is 80
- D. i from A is 40
i from A is 40
- E. i from A is 80
i from A is 80

18. Which of the following is a Python language feature that enforces information hiding?
- A. Encapsulation
 - B. Inheritance
 - C. Class
 - D. Polymorphism
 - E. Instantiation

19. Which of the following allows one class to be derived from another?
- A. encapsulation
 - B. inheritance
 - C. Object
 - D. Polymorphism
 - E. Instantiation

20. What is the value of the variable string after the execution of the following Python program?

```
def main():  
    i = 1  
    string = ""  
    while i < 3:  
        string = 'Hello '  
        string += ' world '  
        i = i + 1
```

main()

- A. Hello
- B. world
- C. Hello world
- D. Hello world Hello world Hellow world
- E. Hello world Hello world

21. Given the string `s="Programming is fun"`,

What is the output for the following:

```
print ( s [4:7] + s [9:] , s [:4] )  
print ( s *2)
```

- A. rammng is fun Prog
Programming is funProgramming is fun
- B. ramng is fun Prog
Programming is funProgramming is fun
- C. ramg is fun Progr
PProogrraammmiinngg iiss ffuunn
- D. rammng is fun Progr
Programming is funProgramming is fun
- E. ramng is fun Prog
PProogrraammmiinngg iiss ffuunn

22. What is the output of the following Python program?

```
myStr = "\t\tWorld\n"  
print (myStr.strip())
```

- A. \t\tWorld\n
- B. \tWorld\n
- C. \t\tWORLD\n
- D. World
- E. \t\tWorld

23. What is the output of the following Python program?

```
x = 0  
while x < 4:  
    x = x + 1  
x = x - 2  
  
print("x is", x)
```

- A. x is 0
- B. x is 1
- C. x is 2
- D. x is 3
- E. x is 4

24. What is the output of the following Python program?

```
sum = 0
item = 0
while item < 5:
    item += 1
    sum += item
    if sum >= 4: continue

print(sum)
```

- A. 15
- B. 16
- C. 17
- D. 18
- E. 19

25. What will be the output of the following:

```
x=1
s=0
while(x<10):
    s=s+x
    x=x+1
    if (x==5):
        break
else:
    print("The sum of first 9 integers:",s)
print("The sum of",x,'numbers is:',s)
```

- A. The sum of first 9 integers: 0
- B. The sum of first 9 integers: 45
The sum of 10 numbers is: 45
- C. The sum of 5 numbers is: 10
- D. The sum of 10 numbers is: 45
- E. The sum of 5 numbers is: 15

26. What is the output of the following python program?

```
from graphics import *  
win = GraphWin ("My window", 300, 300)  
rect = Rectangle (Point(20,20), Point(80,80))  
rect.draw (win)  
for i in range (10) :  
    rect.move (3,5)  
print(rect.getCenter().getX(), rect.getCenter().getY())
```

- A. Error (getCenter() cannot be used on a Rectangle object)
- B. 90.0 100.0
- C. 50.0 80.0
- D. 80.0 100.0
- E. 110.0 130.0

27. You are given these lines:

```
cir = Circle(Point(100,100), 50)
rect = Rectangle(Point(50,50), Point(100,100))
line = Line(Point(50,50), Point(80,80))
point = Point(50, 100)
```

which of the following lines does not work properly?

- I cir.getX().getCenter()
 - II rect.getCenter().getX()
 - III line.getX()
 - IV point.getX()
 - V point.getCenter()
- A. I and V
 - B. only III
 - C. I and III
 - D. II, III
 - E. I, III and V

28. How many times will "I like CS177" be printed?

```
def func(a,b):  
    for x in range(a):  
        for y in range(b):  
            print("I like CS177")
```

```
func(4,3)
```

- A. 1
- B. 2
- C. 4
- D. 7
- E. 12

29. What is the output of the following Python program?

```
myString = "Onomatopoeia"  
print (myString[3:10]+myString[8:11])
```

- A. matopoeioeia
- B. matopoeei
- C. matopoeioei
- D. matopoeoei
- E. atopoeoei

30. Which of the following is not a class within the graphics library:

- A. Rectangle
- B. Square
- C. Point
- D. Polygon
- E. Line

31. What is the output of the following Python program?

```
myList = [1, 5, 5, 5, 5, 5, 1]
max = myList[0]
indexOfMax = 0
for i in range(1, len(myList)):
    if myList[i] > max:
        max = myList[i]
        indexOfMax = i

print(indexOfMax)
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5

32. What is the output of the following Python program?

```
m = [[x, x + 1, x + 2] for x in range(0, 3)]  
print(m)
```

- 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]
- E. [[0, 1, 2], [0, 1, 2], [0, 1, 2]]

33. For what values of x, is the output "True"?

```
def lp(year):  
    return((year%4==0 and year%100 !=0) or year%400 == 0)
```

```
def main():  
    y = lp(x)  
    print(y)
```

```
main()
```

- A. 1990
- B. 2000
- C. 1116
- D. 1111
- E. None of the above

34. Assume myList is a list of positive numbers. What does this Python program do?

```
maxnum = -1
for i in range(len(mylist)) :
    if myList[i] in myList[i+1:] and maxnum < myList[i] :
        maxnum = myList[i]
print(maxnum)
```

- A. Find the largest duplicate element in myList, and output -1 if there is no duplicates
- B. Find the duplicate most times element in myList, and output -1 if there is no largest
- C. Find the largest and duplicate most times element in myList, and output -1 if there is any error
- D. Find the largest and duplicate most times element in myList, and -1 will never be returned
- E. None of the above is correct

35. What is the output of the following Python program?

```
names = ["Amar", "Bear", "Charlton", "Daman"]  
print names[-1][-1]
```

- A. A
- B. Daman
- C. Error
- D. n
- E. Dama

36. Which of the following statements uses list comprehension to create a list named `notodd` containing the positive, even integers up to and including `n`.

I `notodd = [i for i in range(2, n, 2)]`

II `notodd = [i for i in range(2, n+1, 2)]`

III `notodd = [i for i in range(2, n+1, 2): notodd.append(i)]`

IV `notodd = [i for i in range(2, n+1)if not i%2]`

V `notodd = [i for i in range(2, n+1)if i%2==0]`

VI `notodd = [i for i in range(2, n) if not i%2: notodd.append(i)]`

VII `notodd = [i for i in range(2, n) if i%2==0 notodd.append(i)]`

- A. I, III, VI
B. III, VI
C. II, IV, V
D. II, III, VI
E. II, IV, V, VII

37. What does the following Python program print?

```
x = "This sentence has: some punctuation!"  
for c in " !":  
    x.replace(c, ' ')  
print(x)
```

- A. This sentence has: some punctuation!
- B. This sentence has some punctuation
- C. This sentence has: some punctuation
- D. This sentence has some punctuation!
- E. None of the above is correct

38. What is the output of the following Python program?

```
S = set()  
T = {1, 0, 2, 3, 2, 3}  
U = {9, 5, 1, 4, 3}  
S.update(T)  
S.update(U)  
print(S)
```

- A. 1, 0, 2, 3, 2, 9, 5, 1, 4, 3
- B. 0, 1, 2, 3, 4, 5, 9
- C. 9, 5, 1, 4, 3
- D. 9, 5, 1, 4, 3, 3
- E. None of the above is correct

Read following instructions and answer Question 39 and 40:

Given two dictionaries, d1 and d2, create a new dictionary with the following property: for each entry (a, b) in d1, if there is an entry (b, c) in d2, then the entry (a, c) should be added to the new dictionary. For example, if d1 is 2:3, 8:19, 6:4, 5:12 and d2 is 2:5, 4:3, 3:9, then the new dictionary should be 2:9, 6:3 Associate the new dictionary with the variable d3

```
def mydictionaries(d1, d2):
    (*****1*****)
    for i in d1:
        if d1[i] in d2.keys():
            (*****2*****)
```

39. For (*****1*****) you should have:

- A. d3 = {}
- B. d1 = {}
- C. d1 = {}
d2 = {}
- D. d1 = {}
d2 = {}
d3 = {}
- E. d3 = {}
i = 0

40. For (*****2*****) you should have:

- I d3 [i] = d2[d1[i]]
 - II d3 [i] = d1[i]
 - III d3.update(i: d2[d1[i]])
 - IV d3.update(d2[i]: d2[d1[i]])
- A. I
 - B. I or II
 - C. I or II or III
 - D. I or III or IV
 - E. III or IV