

1. What is the output of the following Python program

```
list_A = [x for x in range (10) if x % 2 == 0]
list_B = list_A
for i in range(1, 8, 2):
    list_B.append(i)
list_C = [x for x in range(16) if x not in list_A]
print(list_C)
```

A. [1, 3, 5, 7, 9, 10, 11, 12, 13, 14, 15]

B. [2, 4, 6, 8]

C. [9, 10, 11, 12, 13, 14, 15]

D. [2, 4, 6, 8, 10, 12, 14]

E. [10, 12, 14]

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

```
my_list = [0] * 10
index = 7

for i in range(4):
    index = (index * 3 + 3) % 10
    my_list[index] = 1

print( my_list )
```

A. [0, 0, 0, 1, 0, 1, 0, 1, 1, 0]

B. [0, 0, 0, 0, 1, 0, 1, 1, 1, 0]

C. [0, 0, 0, 0, 1, 1, 0, 1, 0, 1]

D. [0, 0, 0, 0, 1, 1, 0, 1, 1, 0]

E. None of the above

3. L_1 is a line and the two points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ are lying on it.
 L_2 is another line and the two points $p'_1 = (x'_1, y'_1)$ and $p'_2 = (x'_2, y'_2)$ are lying on it. If:

$$\frac{y_2 - y_1}{x_2 - x_1} * \frac{y'_2 - y'_1}{x'_2 - x'_1} = -1$$

then the two lines L_1 and L_2 are orthogonal (perpendicular) on each other.

Which of the following Python functions returns true if the input lines are orthogonal?

I.

```
def orthogonalCheck(line1, line2):
    n1 = float(line1.getP2().getY() - line1.getP1().getY())
    d1 = float(line1.getP2().getX() - line1.getP1().getX())
    n2 = float(line2.getP2().getY() - line2.getP1().getY())
    d2 = float(line2.getP2().getX() - line2.getP1().getX())
    if(n1/d1 * n2/d2 == -1):
        return True
    return False
```

II.

```
def orthogonalCheck(line1, line2):
    n1 = float(line1.getCenter().getY() - line1.getP1().getY())
    d1 = float(line1.getCenter().getX() - line1.getP1().getX())
    n2 = float(line2.getP2().getY() - line2.getCenter().getY())
    d2 = float(line2.getP2().getX() - line2.getCenter().getX())
    if(n1/d1 * n2/d2 == -1):
        return True
    return False
```

III.

```
def orthogonalCheck(line1, line2):
    n1 = float(line1.getP2().getY() - line2.getP1().getY())
    d1 = float(line1.getP2().getX() - line2.getP1().getX())
    n2 = float(line2.getP2().getY() - line1.getP1().getY())
    d2 = float(line2.getP2().getX() - line1.getP1().getX())
    if(n1/d1 * n2/d2 == -1):
        return True
    return False
```

IV.

```
def orthogonalCheck(line1, line2):
    n1 = float(line1.getP1().getY() - line1.getP2().getY())
    d1 = float(line1.getP1().getX() - line1.getP2().getX())
    n2 = float(line2.getP1().getY() - line2.getP2().getY())
    d2 = float(line2.getP1().getX() - line2.getP2().getX())
    if(n1/d1 * n2/d2 == -1):
        return True
    return False
```

V.

```
def orthogonalCheck(line1, line2):
    n1 = float(line1.getP1().getX() - line1.getP2().getX())
    d1 = float(line1.getP2().getY() - line1.getP1().getY())
    n2 = float(line2.getP2().getX() - line2.getP1().getX())
    d2 = float(line2.getP2().getY() - line2.getP1().getY())
    if(n1/d1 * n2/d2 == 1):
        return True
    return False
```

- A. I
- B. I, II, IV, V**
- C. I, IV
- D. III, V
- E. None of the above.

4. Which of the following functions draws on the *GraphWin* object "win" a square of side length equals "length" and having the upper left points "pt1"?

I.

```
def drawSquare (win, pt1, length):  
    pt2 = Point(pt1.getX()+length, pt1.getY())  
    pt3 = Point(pt1.getX(), pt1.getY()+length)  
    pt4 = Point(pt1.getX()+length, pt1.getY()+length)  
    Line(pt1, pt2).draw(win)  
    Line(pt1, pt3).draw(win)  
    Line(pt2, pt4).draw(win)  
    Line(pt3, pt4).draw(win)
```

II.

```
def drawSquare (win, pt1, length):  
    pt2 = Point(pt1.getX()+length, pt1.getY())  
    rect = Rectangle(pt1, pt2)  
    rect.draw(win)
```

III.

```
def drawSquare (win, pt1, length):  
    pt2 = Point(pt1.getX()+length, pt1.getY())  
    pt3 = Point(pt1.getX(), pt1.getY()+length)  
    pt4 = Point(pt1.getX()+length, pt1.getY()+length)  
    Rectangle(pt1, pt2).draw(win)  
    Rectangle(pt1, pt3).draw(win)  
    Rectangle(pt2, pt4).draw(win)  
    Rectangle(pt3, pt4).draw(win)
```

IV.

```
def drawSquare (win, pt1, length):  
    for j in range(length):  
        win.plot(pt1.getX(), pt1.getY()+j+1)  
        win.plot(pt1.getX()+j+1, pt1.getY())  
    for j in range(length):  
        win.plot(pt1.getX()+length, pt1.getY()+j+1)  
        win.plot(pt1.getX()+j+1, pt1.getY()+length)
```

A. I,II

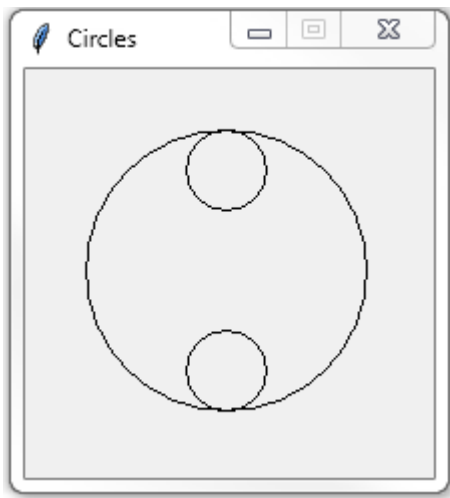
B. I

C. II

D. I, III, IV

E. None of the above.

5. What should be written at ? so that the output is as shown in the figure?



```
from graphics import *
def Main():
    win = GraphWin('Circles', 200, 200)
    Circle(Point(100,50), 20).draw(win)
    Circle(Point(100,150), 20).draw(win)
    cir = ?
    cir.draw(win)
    win.getMouse()
    win.close()
Main()
```

- A. Circle(Point(100,100), 50)
- B. Circle(Point(100,75), 100)
- C. Circle(Point(100,100), 30)
- D. Circle(Point(100,100), 70)**
- E. None of the above.

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

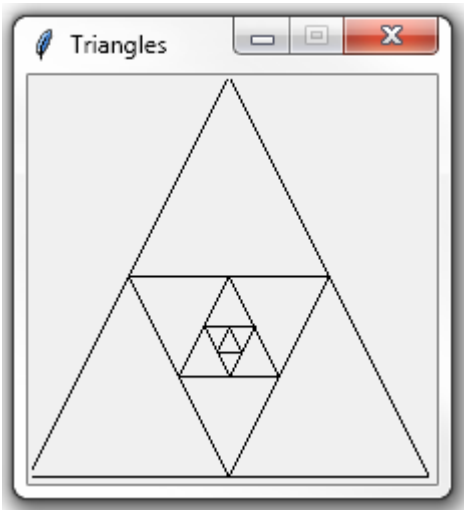
```
from graphics import *

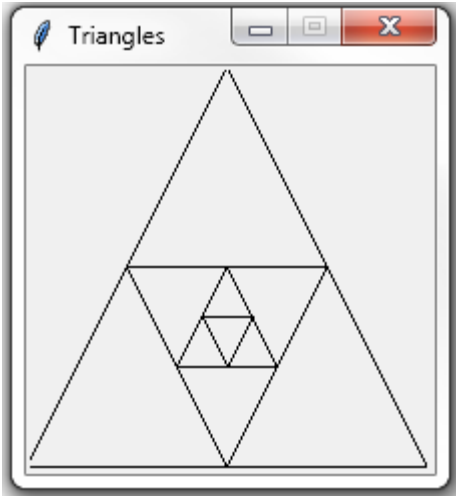
def triangles(win, p1, p2, p3, count):
    if(count):
        line1 = Line(p1,p2)
        line2 = Line(p1,p3)
        line3 = Line(p2,p3)
        line1.draw(win)
        line2.draw(win)
        line3.draw(win)
        triangles(win, line1.getCenter(), line2.getCenter(),
                line3.getCenter(), count-1)
    return

def Main():
    win = GraphWin('Triangles', 200, 200)
    p1 = Point(100,0)
    p2 = Point(0,200)
    p3 = Point(200,200)
    triangles(win, p1, p2, p3, 5)
    win.getMouse()
    win.close()

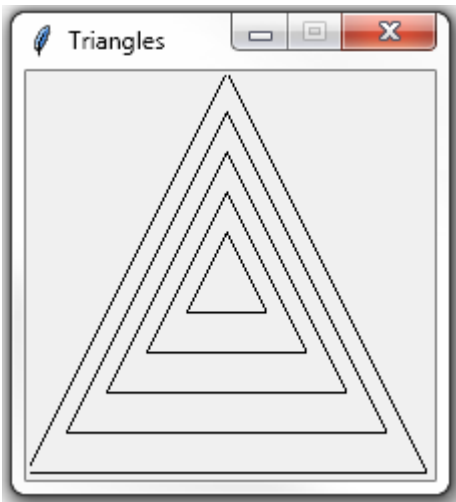
Main()
```

A.

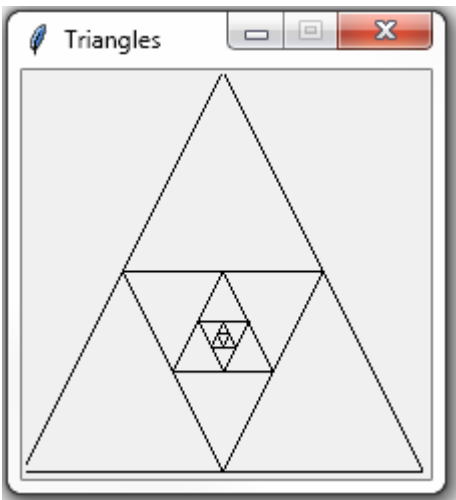




B.



C.



D.

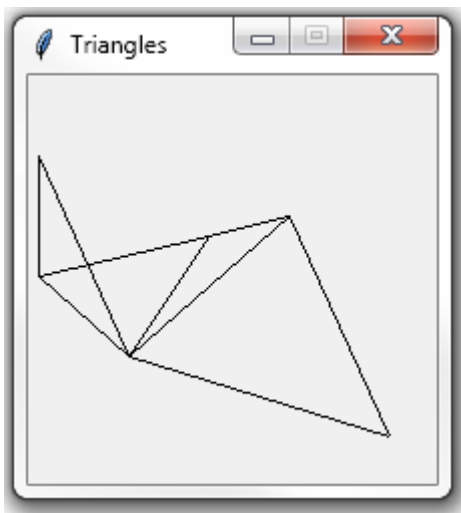
E. The function will never terminate.

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

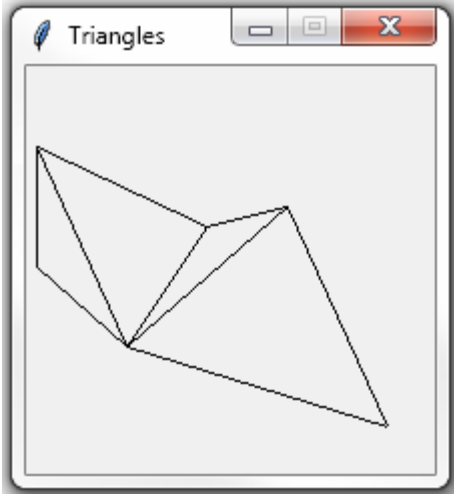
```
from graphics import *

def Main():
    win = GraphWin('Triangles', 200, 200)
    p1 = Point(5,40)
    p2 = Point(5,100)
    Line(p1, p2).draw(win)
    pointList = [Point(50,140), Point(90,80), Point(130,70), Point
        (180,180)]
    for p in pointList:
        if(p.getY()-p2.getY() > 0):
            Line(p1, p).draw(win)
            Line(p2, p).draw(win)
            p2 = p
        else:
            Line(p1, p).draw(win)
            Line(p2, p).draw(win)
            p1 = p
    win.getMouse()
    win.close()
```

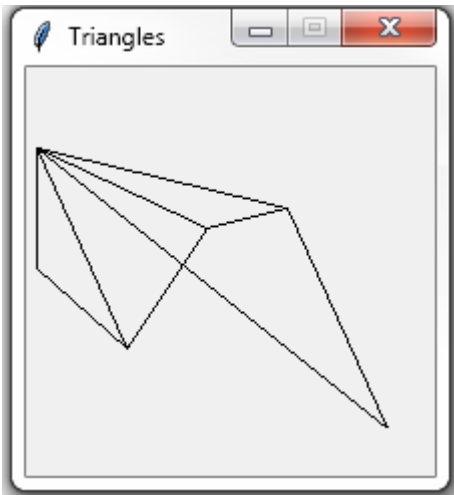
Main()



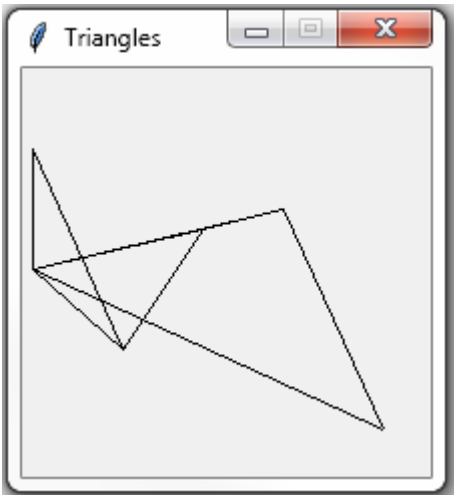
A.



B.



C.



D.

E. None of the above.

8. The following function takes two rectangle objects as inputs and returns True if these rectangles overlap with each other and returns False otherwise
What should be written at ? so that the function is correct?

```
def overlapRectangle(r1 , r2):  
    c1 = r1.getCenter()  
    c2 = r2.getCenter()  
    x1 = abs(r1.getP1().getX() - r1.getP2().getX())  
    y1 = abs(r1.getP1().getY() - r1.getP2().getY())  
    x2 = abs(r2.getP1().getX() - r2.getP2().getX())  
    y2 = abs(r2.getP1().getY() - r2.getP2().getY())  
    distX = abs(c1.getX()-c2.getX())  
    distY = abs(c1.getY()-c2.getY())  
    if(?):  
        return True;  
    return False;
```

- A. $\text{distX} \leq (x1/2+x2/2)$ or $\text{distY} \leq (y1/2+y2/2)$
B. **$\text{distX} \leq (x1/2+x2/2)$ and $\text{distY} \leq (y1/2+y2/2)$**
C. $\text{distX} \leq (x1+x2)$ or $\text{distY} \leq (y1+y2)$
D. $\text{distX} \leq (x1+x2)$ and $\text{distY} \leq (y1+y2)$
E. None of the above.

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

```
from graphics import *
def Main():
    win = GraphWin('Gfx', 200, 200)
    win.setCoords(0, 0, 200, 200)
    rec1 = Rectangle(Point(20,20),Point(120,120))
    print(rec1.getCenter().getX(), rec1.getCenter().getY())
    win.close()
```

Main()

- A. 130.0 70.0
- B. 70.0 130.0
- C. 130.0 130.0
- D. 70.0 70.0**
- E. None of the above.

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

```
class Node:
    def __init__(self, value):
        self.value = value
        self.left, self.right = None, None

def fun( node ):
    if node != None:
        fun( node.left )
        print(node.value)
        fun( node.right )

node1 = Node(10)
node2 = Node(20)
node3 = Node(30)
node4 = Node(40)
node5 = Node(50)

node1.left = node2
node1.right = node3
node2.left = node4
node2.right = node5

fun( node1 )
```

- A. 10
20
30
40
50
- B. 10
20
40
50
30
- C. 40
50
20
30
10

(Please see next page...)

- D. 40
 - 20
 - 50
 - 10
 - 30
- E. Error