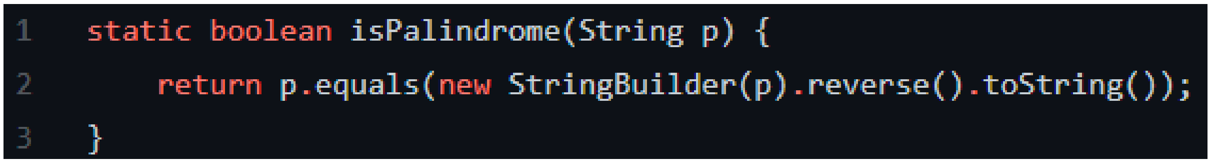
Sample Questions

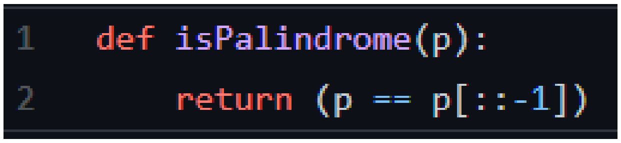
Sample Question 1. A palindrome is a string that is spelt the same forwards and backwards. The following code tries to check if the given string is a palindrome or not. It does not work properly when the input is given as a mixed-case string.

Java:



```
1   static boolean isPalindrome(String p) {
2       return p.equals(new StringBuilder(p).reverse().toString());
3   }
```

Python:



```
1   def isPalindrome(p):
2       return (p == p[::-1])
```

How can we fix the code?

I. convert p to lowercase before line 2

II. convert p to uppercase before line 2

III. change condition on line 2 to
(p[::-1] == p) [python] or
(new StringBuilder(p).reverse().toString()).equals(p); [Java]

IV. change condition on line 2 to
(p != p[::-1]) [python] or
!p.equals(new StringBuilder(p).reverse().toString()); [Java]

A. Only I
B. Only II
C. Both I & II
D. All options work (I, II, III, IV)
E. None of the above.

Correct Answer: C, converting the case to uppercase or lowercase will ensure this works for mixed case strings.

Sample Question 2. What would be considered an efficient way of finding an element in a binary search tree (BST)?

I. Use a recursive find algorithm that goes down one branch of the BST

II. Use an iterative find algorithm that goes down one branch of the BST

III. Use a breadth-first search algorithm

IV. Use a depth-first search algorithm

A. I & II

B. Only I
C. Only II
D. All of I & II & III & IV
E. None of the Above

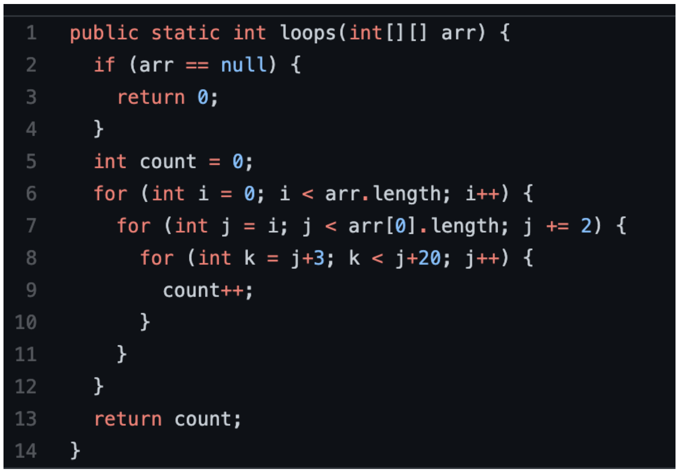
Correct answer A. While all of them can get the answer, but only I and II are efficient. While iteration is most efficient, even recursive tree-walk down a branch is very efficient with log(N) traversals.

Sample Question 3. How many nodes are there in a fully balanced binary tree of depth 3?
A. 3
B. 7
C. 15
D. An indeterminate number between 0 and 16
E. None of the above

Correct answer B, i.e. 4+2+1

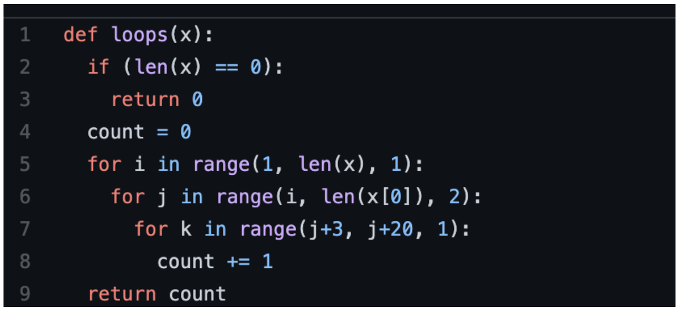
Sample Question 4. What is the asymptotic O(.) time-complexity of the following code snippet? Assume that the input parameter to the loops function is a 2D-array of integers.

Java



```java
public static int loops(int[][] arr) {
  if (arr == null) {
    return 0;
  }
  int count = 0;
  for (int i = 0; i < arr.length; i++) {
    for (int j = i; j < arr[0].length; j += 2) {
      for (int k = j+3; k < j+20; j++) {
        count++;
      }
    }
  }
  return count;
}
```

Python



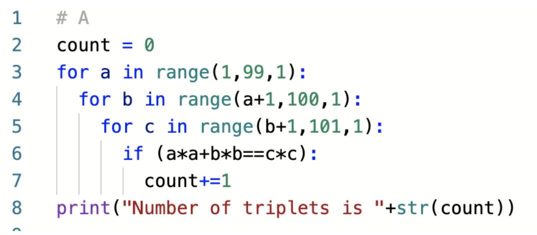
```python
def loops(x):
  if (len(x) == 0):
    return 0
  count = 0
  for i in range(1, len(x), 1):
    for j in range(i, len(x[0]), 2):
      for k in range(j+3, j+20, 1):
        count += 1
  return count
```

A. O(n^2)
B. O(n^3)
C. O(n*log n)
D. O(n)
E. none of the above

Correct answer: A.

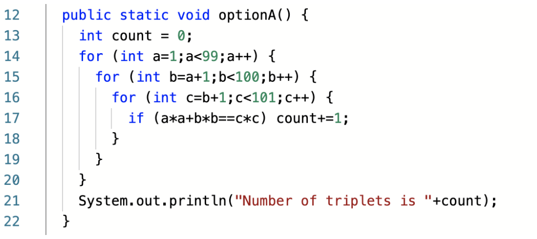
Sample Question 5. A pythagorean triplet (a,b,c) of three positive integers has the property that $a^2 + b^2 = c^2$. Which of the following code samples will print the number of pythagorean triplets satisfying the inequality 1 <= a < b < c <=100

<table>
<tr><th>Option</th><th>Python</th><th>Java</th></tr>
<tr><td>A</td><td>



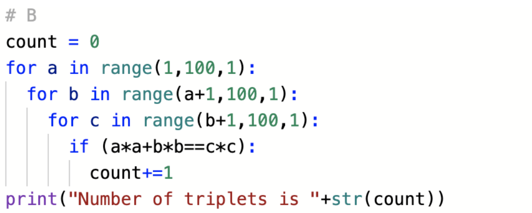
```
# A
count = 0
for a in range(1,99,1):
    for b in range(a+1,100,1):
        for c in range(b+1,101,1):
            if (a*a+b*b==c*c):
                count+=1
print("Number of triplets is "+str(count))
```

</td><td>



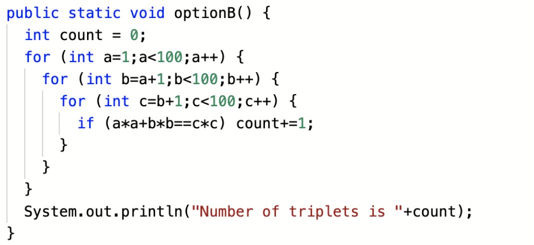
```
public static void optionA() {
  int count = 0;
  for (int a=1;a<99;a++) {
    for (int b=a+1;b<100;b++) {
      for (int c=b+1;c<101;c++) {
        if (a*a+b*b==c*c) count+=1;
      }
    }
  }
  System.out.println("Number of triplets is "+count);
}
```

</td></tr>
<tr><td>B</td><td>



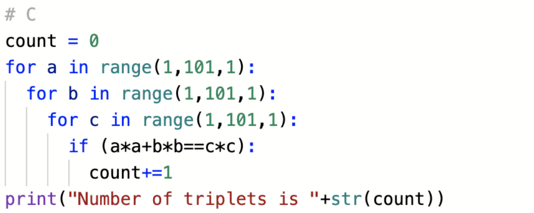
```
# B
count = 0
for a in range(1,100,1):
    for b in range(a+1,100,1):
        for c in range(b+1,100,1):
            if (a*a+b*b==c*c):
                count+=1
print("Number of triplets is "+str(count))
```

</td><td>



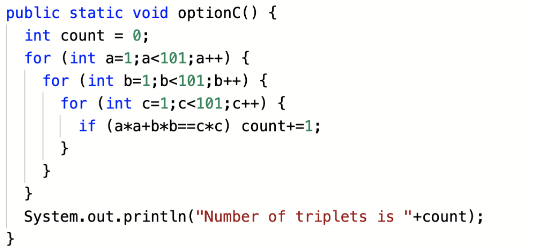
```
public static void optionB() {
  int count = 0;
  for (int a=1;a<100;a++) {
    for (int b=a+1;b<100;b++) {
      for (int c=b+1;c<100;c++) {
        if (a*a+b*b==c*c) count+=1;
      }
    }
  }
  System.out.println("Number of triplets is "+count);
}
```

</td></tr>
<tr><td>C</td><td>



```
# C
count = 0
for a in range(1,101,1):
    for b in range(1,101,1):
        for c in range(1,101,1):
            if (a*a+b*b==c*c):
                count+=1
print("Number of triplets is "+str(count))
```

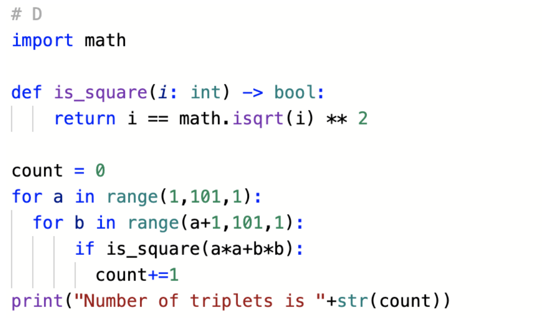
</td><td>



```
public static void optionC() {
  int count = 0;
  for (int a=1;a<101;a++) {
    for (int b=1;b<101;b++) {
      for (int c=1;c<101;c++) {
        if (a*a+b*b==c*c) count+=1;
      }
    }
  }
  System.out.println("Number of triplets is "+count);
}
```

</td></tr>
</table>

<table>
<tr>
<td>D</td>
<td>

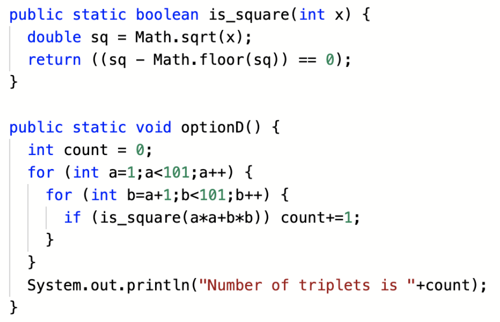


```python
# D
import math

def is_square(i: int) -> bool:
    return i == math.isqrt(i) ** 2

count = 0
for a in range(1,101,1):
  for b in range(a+1,101,1):
      if is_square(a*a+b*b):
        count+=1
print("Number of triplets is "+str(count))
```

</td>
<td>



```java
public static boolean is_square(int x) {
  double sq = Math.sqrt(x);
  return ((sq - Math.floor(sq)) == 0);
}

public static void optionD() {
  int count = 0;
  for (int a=1;a<101;a++) {
    for (int b=a+1;b<101;b++) {
      if (is_square(a*a+b*b)) count+=1;
    }
  }
  System.out.println("Number of triplets is "+count);
}
```

</td>
</tr>
<tr>
<td>E</td>
<td colspan="2">None of the above will print the right answer</td>
</tr>
</table>

Correct answer: A