**Medium-to-high level pseudocode questions on Functions**

**Category 1: Recursive Functions – Q1 to Q20**

**Q1. What will be the output of the following?**

function sum(n)

if n == 1

return 1

return n + sum(n - 1)

print(sum(5))

Answer will be 15

**Q2. Predict the result:**

function mystery(a, b)

if b == 0

return 0

return a + mystery(a, b - 1)

print(mystery(3, 4))

answer will be 12

**Q3. What is the output?**

function f(n)

if n == 0

return 0

else

return n + f(n - 2)

print(f(6))

Answer will be 12

**Q4. Output?**

function fun(x)

if x == 0

return

fun(x - 1)

print(x)

fun(3)

Answer will be 123

**Q5. Output of the code:**

function fact(n)

if n == 0

return 1

return n \* fact(n - 1)

print(fact(4))

**Q6. What is returned?**

function fib(n)

if n == 0

return 0

else if n == 1

return 1

return fib(n - 1) + fib(n - 2)

print(fib(6))

**Q7. Predict output:**

function fun(n)

if n <= 1

return n

return fun(n - 1) + fun(n - 3)

print(fun(5))

**Q8. Count the number of recursive calls for fun(3):**

function fun(n)

if n == 0

return

fun(n - 1)

fun(n - 1)

**Q9. What is the output?**

function sumDigits(n)

if n == 0

return 0

return (n mod 10) + sumDigits(n div 10)

print(sumDigits(1234))

**Q10. Trace the output:**

function reversePrint(n)

if n == 0

return

print(n)

reversePrint(n - 1)

reversePrint(3)

**Q11. Output?**

function foo(n)

if n <= 0

return 0

else if n == 1

return 1

return foo(n - 1) + foo(n - 2)

print(foo(4))

**Q12. What will be printed?**

function series(n)

if n == 1

return 1

return n \* series(n - 1)

print(series(5))

**Q13. How many times will print be called?**

function fun(n)

if n == 0

return

fun(n - 1)

fun(n - 1)

print(n)

fun(2)

**Q14. What is the result?**

function productOfDigits(n)

if n == 0

return 1

return (n mod 10) \* productOfDigits(n div 10)

print(productOfDigits(123))

**Q15. Predict the final output:**

function f(x)

if x < 1

return

f(x / 2==0)

print(x)

f(8)

**Q16. Output?**

function power(x, y)

if y == 0

return 1

return x \* power(x, y - 1)

print(power(2, 4))

**Q17. Result?**

function countDown(n)

if n == 0

return

print(n)

countDown(n - 1)

countDown(3)

**Q18. What will be returned?**

function altSum(n)

if n == 0

return 0

return n - altSum(n - 1)

print(altSum(5))

**Q19. What is the value of fun(3)?**

function fun(n)

if n == 1

return 1

return n \* fun(n - 1) + fun(n - 1)

print(fun(3))

Answer will be 24

**Q20. What does this compute?**

function doubleFactorial(n)

if n <= 0

return 1

return n \* doubleFactorial(n - 2)

print(doubleFactorial(5))