

1) Stack Trace for Recursive Function

Consider the following pseudo-code:

```
defn fib x = if x .lt. 2 then x else fib(x - 1) + fib(x - 2)
```

Show the function call stack (tree format or nested) during the evaluation of `fib(4)`.

2) Parameter Passing

Write a swap method that swaps two variables. Then test whether the original values are swapped outside the method.

3) Concurrency

Explain the difference between a process and a thread. How do they relate to concurrency?

4) Exception Handling

Write a method that asks the user for a number and divides 100 by it. Use exception handling to catch cases when:

- The input is not a number (ArgumentError)
- The input is zero (ZeroDivisionError)

Input: "abc"

```
Enter a number to divide 100 by: abc
```

```
Error: Please enter a valid number.
```

Input: 0

```
Enter a number to divide 100 by: 0
```

```
Error: Division by zero is not allowed.
```

Input: 4

```
Enter a number to divide 100 by: 4
```

```
Result: 100 / 4.0 = 25.0
```