1. Recursive Method Analysis (`ct` method)

Method Behavior:

The `ct` method prints "Starting nn", recursively calls itself with n/3n / 3 if n>0n > 0, and prints "Middle nn" after the recursive call.

Outputs:

a) **`ct(13)`**
Output:

- Starting 13
- Starting 4
- Starting 1
- Starting 0
- Middle 1
- Middle 4
- Middle 13

b) **`ct(3)`**
Output:

- Starting 3
- Starting 1
- Starting 0
- Middle 1
- Middle 3

c) **`ct(0)`**
Output:

- Starting 0

4. Recursive Method with Modulo (`ct` method with `%`)

Method Behavior:

The method prints nn, recursively calls itself with n/3n / 3 if nn is odd, and n/2n / 2 if nn is even, until n>0n > 0 fails.

Outputs:

a) `ct(13)`
Output:

- 13
- 4
- 2
- 1
- 0

b) `ct(14)`
Output:

- 14
- 7
- 2
- 1
- 0

c) `ct(15)`
Output:

- 15
- 5
- 1
- 0

5. Recursive Method for Digits (`ct` method for digits)

Method Behavior:

The method recursively calls itself with n/10n / 10 until n>0n > 0 fails and prints the remainder n%10n \% 10 during each recursive return. This prints the digits of nn in the order they appear.

Outputs:

a) `ct(13)`
Output:

- 1
- 3

b) `ct(124)`
Output:

- 1
- 2
- 4

c) `ct(21785)`
Output:

- 2
- 1
- 7
- 8
- 5

General Purpose:

This method prints the digits of the given number from left to right.

6. Recursive Method with String Input (`whatzItDo`)

Method Behavior:

The method reads user input recursively until a period (`.`) is entered. It then prints the characters in reverse order as recursion unwinds.

Outputs:

a) **Input: T, E, S, T, .**
Output:

- T
- S
- E
- T

General Purpose:

The method reads characters from the user, stopping at `.`, and prints them in reverse order.

7. Stability of Sorting Algorithms

Stable vs. Unstable Sort:

- **Stable Sort:** Preserves the relative order of elements with equal keys.
- **Unstable Sort:** May change the relative order of elements with equal keys.

Sorting Algorithms:

a) **Selection Sort:**
Unstable. Swapping during selection can alter the order of equal keys.
Example:
Before Sorting: [(Jon,19),(Tom,19)][(Jon, 19), (Tom, 19)]
After Sorting: [(Tom,19),(Jon,19)][(Tom, 19), (Jon, 19)]

b) **Merge Sort:**
Stable. Merging preserves the order of equal keys.
Example:
Before Sorting: [(Jon,19),(Tom,19)][(Jon, 19), (Tom, 19)]
After Sorting: [(Jon,19),(Tom,19)][(Jon, 19), (Tom, 19)]