Test cases proposed by **Bard** and implemented by **ChatGPT**-4:

```
Algorithm 2: Autogenerated Test Cases proposed by Bard.
Input: 1404 images, 15 MATLAB files
Output: Skeletonized images
1:Complex Data Structure - Linked List
2: Complex Algorithm - Quick Sort
3: Complex Library - NumPy
4: Error-Prone Code - Recursion
5: Multithreading
6: Modular code
```

Table of Contents

Test Case 1: Complex Data Structure - Linked List

```
class Node:
    def __init__(self, data=None):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def append(self, data):
        if not self.head:
            self.head = Node(data)
        else:
            current = self.head
            while current.next:
                current = current.next
            current.next = Node(data)
```

Test Case 2: Complex Algorithm - Quick Sort

```
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)
```

Test Case 3: Complex Library - NumPy

```
import numpy as np

# Create a 3x3 array of all zeros
a = np.zeros((3, 3))

# Create a 3x3 identity matrix
b = np.eye(3)

# Matrix multiplication
c = np.dot(a, b)
```

Test Case 4: Error-Prone Code - Recursion

```
def recursive_function(n):
    if n == 1:
        return 1
    else:
        return n * recursive_function(n-1)
```

Test Case 5: Multithreading

```
import threading

def print_numbers():
    for i in range(10):
        print(i)

def print_letters():
    for letter in 'abcdefghij':
        print(letter)

t1 = threading.Thread(target=print_numbers)
t2 = threading.Thread(target=print_letters)

t1.start()
t2.start()

t1.join()
t2.join()
```

Test Case 6: Modular code

These test cases cover a variety of complexities and challenges, including complex data structures, algorithms, libraries, recursion, and multithreading. They can be used to evaluate the performance and accuracy of the M-to-PY converter.