

SOURCE CODE 1. Zyklenzerlegung einer Permutation, die als Liste gegeben ist.

```
def cycle_decomposition(pi):  
    """Bestimme die (nicht-kanonische) Zyklenzerlegung einer Permutation"""  
    n = len(pi)  
    # "Buchführung": Liste der Elemente (MINUS 1), die noch nicht auf  
    # Zyklen "verteilt" wurden.  
    yet_to_consider = list(range(n))  
    # Rückgabewert: Liste von Zyklen (Listen); anfangs leer  
    list_of_cycles = []  
    while len(yet_to_consider):  
        # Erstes noch zu verteilendes Element (MINUS 1) ...  
        i = yet_to_consider[0]  
        # ... eröffnet einen neuen Zyklus:  
        new_cycle = [i+1]  
        # Dieses Element wird von der Liste der noch zu verteilenden  
        # Elemente gestrichen:  
        yet_to_consider = yet_to_consider[1:]  
        # Nun wird der neue Zyklus zusammengestellt:  
        while True:  
            pi_i = pi[i]  
            # Abbruchbedingung: Wenn wir wieder am Anfang des Zyklus'  
            # angekommen sind, ist der Zyklus fertig zusammengestellt  
            # und wird in die Liste der Zyklen aufgenommen.  
            if pi_i == new_cycle[0]:  
                list_of_cycles += [new_cycle]  
                break  
  
            # Ansonsten: Weitermachen! Der Zyklus wird um pi_i "verlängert" ...  
            new_cycle += [pi_i]  
            # und der pi_i entsprechende Index wird aus der Liste der noch  
            # zu behandelnden Elemente entfernt ...  
            yet_to_consider.remove(pi_i - 1)  
            # ... und wir machen weiter ...  
            i = pi_i - 1  
    return list_of_cycles
```

Die Funktion `cycle_decomposition` bestimmt für eine (als Liste) gegebene Permutation π die Zyklenzerlegung, als eine Liste von Listen.