```python
import sys
import itertools

def decompose(permutation):
    """
    Decomposes a permutation into disjoint cycles.
    """
    # Track our visited numbers
    visited = set()
    cycles = []

    # Track every single cycle
    for idx in xrange(len(permutation)):
        if (idx + 1) != permutation[idx] and not idx + 1 in visited:
            cycle = [idx + 1]
            nxt = permutation[idx]
            # Chase the cycle until it repeats
            while idx + 1 != nxt:
                cycle.append(nxt)
                nxt = permutation[nxt - 1]
            cycles.append(cycle)
            visited = visited.union(set(cycle))

    return cycles

def permutations(n):
    """
    Returns a list of all the permutations of 1..n
    """
    return itertools.permutations(range(1, n + 1))

if __name__ == '__main__':
    """
    Generates all elements of Sn
    """

    # num args and args
    num_args = len(sys.argv)
    if num_args != 2:
        print 'usage: python p5a.py <n>'
    else:
        perms = permutations(int(sys.argv[1]))
        for p in perms:
            print decompose(p)
```