Problem

Let $n \in \mathbb{N}$ be an even number, say $n = 2m$. Also let $\mathbb{I}_n = \{1, 2, 3, \cdots, n\}$.
Let $f : \mathbb{I}_n \to \mathbb{I}_n$ be a bijection; in other words, it represents a permutation of length n.
If $i \in \mathbb{I}_n$ such that $f(i) = i$, then i is called a *fixed point* of f.
If $i \in \mathbb{I}_n$ such that $f(i) = n + 1 - i$, then i is called a *mirror point* of f.

Let X_n denote the number of permutations of length n with no fixed points and no mirror points. Find the value of X_n.

Solution

We wish to apply the Principle of Inclusion-Exclusion (PIE).
Let our universal set be S_n, the set of all permutations of length n; hence $|S_n| = n!$.
For all $i \in \mathbb{I}_n$, we define $A_i = \{f \in S_n | f(i) = i\}$, and $B_i = \{f \in S_n | f(i) = n + 1 - i\}$.
In other words, A_i is the set of all permutations for which i is a fixed point, and B_i is the set of all permutations for which i is a mirror point.

We note the following lemma, which will help us to make suitable cases later on:
Lemma: For any $i, j \in \mathbb{I}_n$ such that $i + j = n + 1$, $|A_i \cap B_i| = |A_i \cap B_j| = 0$.
Proof: It suffices to note that n is even, hence no element can be a fixed point as well as a mirror point of the same permutation.
(If n is odd, then only the middle element $\frac{n+1}{2}$ can be a fixed point as well as a mirror point. The reader is encouraged to suitably modify this solution to work for odd values of n as well.)

By PIE, what we want to calculate is:

$$X_n = |S| - |A_1 \cup A_2 \cup \cdots \cup A_n \cup B_1 \cup B_2 \cup \cdots \cup B_n| = |S| + \sum_{t=1}^{2n} (-1)^t E_t \tag{1}$$

where E_t denotes the sum of the sizes of all t-set intersections from the A_i, B_j families.
In other words, $E_t = \sum |A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k} \cap B_{j_1} \cap B_{j_2} \cap \cdots \cap B_{j_l}|$, where the summation is taken over all $1 \le i_1 < i_2 < \cdots < i_k \le n$ and $1 \le j_1 < j_2 < \cdots < j_l \le n$ such that $k + l = t$; including the possibility that either of k, l could be zero.

To calculate E_t, all such t-set intersections $I = |A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k} \cap B_{j_1} \cap B_{j_2} \cap \cdots \cap B_{j_l}|$ can be partitioned into the following cases:

Case 1: There exist some $1 \le r \le k$ and $1 \le s \le l$, such that $i_r = j_s$, or $i_r + j_s = n + 1$
Due to the above lemma, we note that $|A_{i_r} \cap B_{j_s}| = 0$; hence $|I| = 0$.

Case 2: There do not exist any $1 \le r \le k$ and $1 \le s \le l$, for which $i_r = j_s$ or $i_r + j_s = n + 1$.
For any permutation f that belongs to the t-set intersection I as defined above, the following values of f are forced:
$f(i_r) = i_r$ for all r from 1 to k, and $f(j_s) = n + 1 - j_s$ for all s from 1 to l.
All the above values in the domain and codomain are distinct, which is consistent with f being a bijection.
All the remaining $n - k - l = n - t$ values of f can be chosen in $(n - t)!$ ways; so $|I| = (n - t)!$.

Now we only need to count the t-set intersections that belong to case 2.
Noting that $n = 2m$, we partition the $2n$ sets of A_i, B_j families into m groups of 4 sets each: $\{A_1, A_n, B_1, B_n\}, \{A_2, A_{n-1}, B_2, B_{n-1}\}, \cdots$
$\cdots, \{A_i, A_{n+1-i}, B_i, B_{n+1-i}\}, \cdots, \{A_m, A_{m+1}, B_m, B_{m+1}\}$
As per the constraint imposed by case 2, no two A_i, B_j from the same group can be used in any given t-set intersection. So we can use at most 2 out of the 4 elements from each group.

Let us count the sub-case in which we use exactly 2 elements from some r groups, and exactly 1 element from some s groups; where $2r + s = t$.
We can choose the first r groups in $\binom{m}{r}$ ways, and the next s groups in $\binom{m-r}{s}$ ways.
For each of the first r groups, we can use either both elements of the A_i family, or both elements of the B_i family, in that group.
For each of the next s groups, we can use any one of the 4 elements in that group.
Thus the number of t-set intersections in this sub-case is $\binom{m}{r}\binom{m-r}{s}2^r 4^s = \binom{m}{r}\binom{m-r}{t-2r}2^{2t-3r}$.

Summing up the above sub-cases over all valid values of r, and noting that each such t-set intersection is of size $(n-t)!$, we get:

$$E_t = (n-t)! \sum_{r=0}^{\lfloor \frac{t}{2} \rfloor} \binom{m}{r} \binom{m-r}{t-2r} 2^{2t-3r} \tag{2}$$

We note that if the two binomial coefficients on the RHS are non-zero, then $m \geq r$ and $m - r \geq t - 2r$; which forces $t \leq 2m = n$. In other words, $E_t = 0$ for all $t > n$; which makes sense because we cannot force more than n constraints on a function having n inputs.
The above formula also gives $E_0 = n!$ which is the size of our universal set S_n.

We can now combine (1) and (2) to write the formula for X_n, which is the number of n-length permutations with no fixed points and no mirror points:

$$X_n = \sum_{t=0}^{n} (-1)^t (n-t)! \sum_{r=0}^{\lfloor \frac{t}{2} \rfloor} \binom{m}{r} \binom{m-r}{t-2r} 2^{2t-3r}$$

By exchanging the order of summation, and other simplifications, we can rewrite this as:

$$X_{2m} = \sum_{s=0}^{m} \binom{m}{s} 2^{m-s} F_1(s)$$

where $F_1(s) = \sum_{k=0}^{s} (2s-k)! \binom{s}{k} (-4)^k$.

Similarly, the reader is encouraged to derive the following formula for the case when n is odd, say $n = 2m + 1$:

$$X_{2m+1} = \sum_{s=0}^{m} \binom{m}{s} 2^{m-s} (F_2(s) - F_1(s))$$

where $F_1(s)$ is the same as above, and $F_2(s) = \sum_{k=0}^{s} (2s-k+1)! \binom{s}{k} (-4)^k$.

It seems difficult to simplify these any further; the reader is encouraged to try the same.

Finally, we provide some initial values of X_n, along with the number of derangements D_n, as well as a sample implementation in Python to count D_n, X_n directly.

n	D_n	X_n
1	0	0
2	1	0
3	2	0
4	9	4
5	44	16
6	265	80
7	1854	672
8	14833	4752
9	133496	48768
10	1334961	440192

```python
def get_permutations(n, isValid):
  # use backtracking to generate all permutations recursively
  def perms_step(n, output, available, isValid):
    # output: permutation generated so far
    # available: elements available to append in this step
    # isValid is a function that takes inputs as n, i, p(i),
    # and returns whether it is ok to put element p(i) at position i
    if len(available) == 0:
      return [output]
    all_outputs = []
    for p_i in available:
      if isValid(n, len(output), p_i):
        newOutput = output.copy(); newOutput.append(p_i)
        newAvailable = available.copy(); newAvailable.remove(p_i)
        all_outputs += perms_step(n, newOutput, newAvailable, isValid)
    return all_outputs
  return perms_step(n, [], set(range(n)), isValid)


def isNotFixedPt(n, i, p_i):
  return p_i != i

def isNotMirrorPt(n, i, p_i):
  return p_i != n - 1 - i

def isNotFixedOrMirrorPt(n, i, p_i):
  return isNotFixedPt(n, i, p_i) and isNotMirrorPt(n, i, p_i)

def print_dn_xn_table(N):
  for n in range(1, N):
    D_n = len(get_permutations(n, isNotFixedPt))
    X_n = len(get_permutations(n, isNotFixedOrMirrorPt))
    print(f'n:{n: 3} D_n:{D_n: 9} X_n:{X_n: 9}')

if __name__ == '__main__':
  print_dn_xn_table(11)
```

mirror_pts.py