7) a) Show that the order of a permutation p is the least common multiple of the length of its cycles.
b) What is the largest possible order of a permutation on 8 points? (In general, the largest order for n points is the value of the *Landau function* for n.)

8) Let $U \leq G$. For $g \in G$ we define $U^g = g^{-1}Ug = \{g^{-1}ug \mid u \in U\}$. Show that $U^g \leq G$.

9) Recall that we defined the group of *units modulo* n as $U(n) = \mathbb{Z}_n^* = \{1 \leq a \leq n \mid \gcd(a,n) = 1\}$, with operation being multiplication modulo n.
a) Calculate $5 \cdot 17^{-1}$ in $U(1234)$.
b) Determine a cyclic and a non-cyclic subgroup of $U(40)$.
c) Determine the multiplication table of $U(8)$. Is $U(8)$ a subgroups of $U(16)$? Explain!

10) Let $G = \langle (1,2,3,8)(4,5,6,7), (1,7,3,5)(2,6,8,4) \rangle$ be a group generated by two permutations.
a) Determine the elements of G and their orders. What is $|G|$?
GAP can help here:

```
gap> a:=(1,2,3,8)(4,5,6,7);
(1,2,3,8)(4,5,6,7)
gap> b:=(1,7,3,5)(2,6,8,4);
(1,7,3,5)(2,6,8,4)
gap> Print(a*b);
(1,6,3,4)(2,5,8,7)
gap> a*b;
(1,6,3,4)(2,5,8,7)
gap> Elements(Group(a,b));
[ (), ...
gap> Size(Group(a,b));
```

respectively:

```
a:=(1,2,3,8)(4,5,6,7);
b:=(1,7,3,5)(2,6,8,4);
Print(a*b);
Print(Elements(Group(a,b)));
```

on `https://tio.run/#gap`

in the standard version,
b) Determine all subgroups of G. (Hint: Every proper subgroup has at most 4 elements. If it contains an element of order 4 it therefore must be cyclic.)

11) Let $G = S_n$ and $a_i = (i, i+1)$ for $i = 1, \ldots, n-1$.
a) Let $g \in S_n$ with and $g(n) = i$. Show that $b = g \cdot a_i \cdot a_{i+1} \cdot \cdots \cdot a_{n-1}$ fulfills that $b(n) = n$, i.e. that $b \in S_{n-1}$.

b) Using the argument from a), show by induction over n that $S_n = \langle a_1, \ldots, a_{n-1} \rangle$.
c) Show that $a_i a_j = a_j a_i$ if $|i-j| \geq 2$ and $a_i a_j a_i = a_j a_i a_j$ if $|i-j| = 1$. (Note: These relations are called "braid relations", the name comes from representing the braiding of strings. As shown in the picture on the side, relations of this type hold for the "twists" of adjacent strings.
One can show that using these braid relations as well as relations $a_i^2 = e$, one can bring every word in S_n into a shortest form.)

a_1: Move left string over middle one,
a_2: Move middle string over right one,