Q1 Proof
A structure is a circle subdivided into n equal pieces with a value at each subdivision point also called a seat, a transformation of a structure can be seen as the rotation of of that circle by some a(360/n) where a is an integer such that the structure “1234” when rotated 90 degrees clockwise gives “4123”. the Equivalent transformation between two combinatorial structures is really an equivalent relation.Two Combinatorial Structures are considered equivalent when the first can be transformed to look as the second with respect to its weights by rotation as such if we have a structures A = (W1, W2, W3, W4), B = (W4, W1, W2, W3) and C = (W2, W3, W4, W1)
A = A is true because A can be transformed to A by a rotation of 0° or 360° clockwise or anticlockwise hence the relation is reflexive
Then A = B and B = A is true because A can be transformed to B by a rotation of 90° clockwise and B can be transformed to A by a rotation of 90° anti clockwise hence the relation is symmetric
And finally if A = B and B = C then A = C. this is true because A can be transformed to B by a rotation of 90° clockwise and B can be transformed to C by a rotation of 180° clockwise and A can be transformed to C by a rotation of 90° anti clockwise and so they are all equivalent. Since the transformation between combinatorial structures is Reflexive, Symmetric and Transitive then it is an equivalent relation.

Q2 Algorithm:
Take in input for both structures
rotate the first structure till all forms are found
store the different forms of the first structure in a list
check to see if the second structure is in that list,
if it is; both structures are equivalent.
if it is not; the structures are not equivalent.
print the results

Q3 Algorithm:

Following question 2
in the list containing all forms of a structure the canonical form is the one with the greatest value
get the form with the greatest value from the list using the Max function

Q4 Algorithm:
a structure is a combination of numbers
generate structures by generating random numbers and joining them together
generate all structures of n such that each structure of n has a length n and contains only values in the range [2,(n-1)]
store all the possible structures of n in a list, the total number of structures of n should be (n-2)^n
an equivalence class is a group of structures that are rotations of each other, you can also say they are structures with the same canonical form

for each equivalence class of all structures of n, there is only 1 unique Canonical form
to find the number of equivalence classes of all structures of n
loop through all structures of n and for each structure, add it's canonical form to a list only if it is not yet in the list
finally count the number of elements in the list with canonical forms, that will give you the number of equivalence classes among all structures of n

Q5

n	# of equivalence classes
2	0
3	1
4	6
5	51
6	700
7	11165