Supplementary material

Computational LEGO Technic Design

Table of Contents

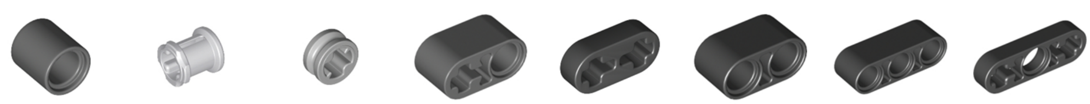
This supplementary material includes the following parts.

Part A - Details on the Brick Set and Connection Mechanisms

Part A.1 - Details on the brick set

Length (mm)	16.0	24.0	32.0	40.0	40.0	48.0	56.0	64.0
Width (mm)	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8
Height (mm)	4.8	4.8	4.8	4.8	4.8	4.8	4.8	4.8
Mass (g)	0.26	0.43	0.60	0.66	0.73	0.94	1.05	1.18

Figure 1. Basic information of axles in the brick set.



Length (mm)	7.4	7.4	7.4	15.5	15.5	15.5	23.3	23.3
Width (mm)	7.4	7.4	7.4	7.4	7.4	7.4	7.4	7.4
Height (mm)	8.0	8.0	4.0	8.0	4.0	8.0	8.0	4.0
Mass (g)	0.17	0.14	0.10	0.46	0.33	0.40	0.70	0.45

Length (mm)	31.3	39.3	47.2	55.2	71.1	87.1	103.0	119.1
Width (mm)	7.4	7.4	7.4	7.4	7.4	7.4	7.4	7.4
Height (mm)	4.0	8.0	4.0	8.0	8.0	8.0	8.0	8.0
Mass (g)	0.55	1.21	0.80	1.74	2.59	2.80	3.30	4.00

Length (mm)	31.2	23.3	46.2	39.3	65.4
Width (mm)	15.4	23.3	27.1	23.3	20.7
Height (mm)	8.0	4.0	8.0	8.0	8.0
Mass (g)	1.43	0.73	2.00	1.77	2.63

Figure 2. Basic information of beams in the brick set.

(Page to Cont.)

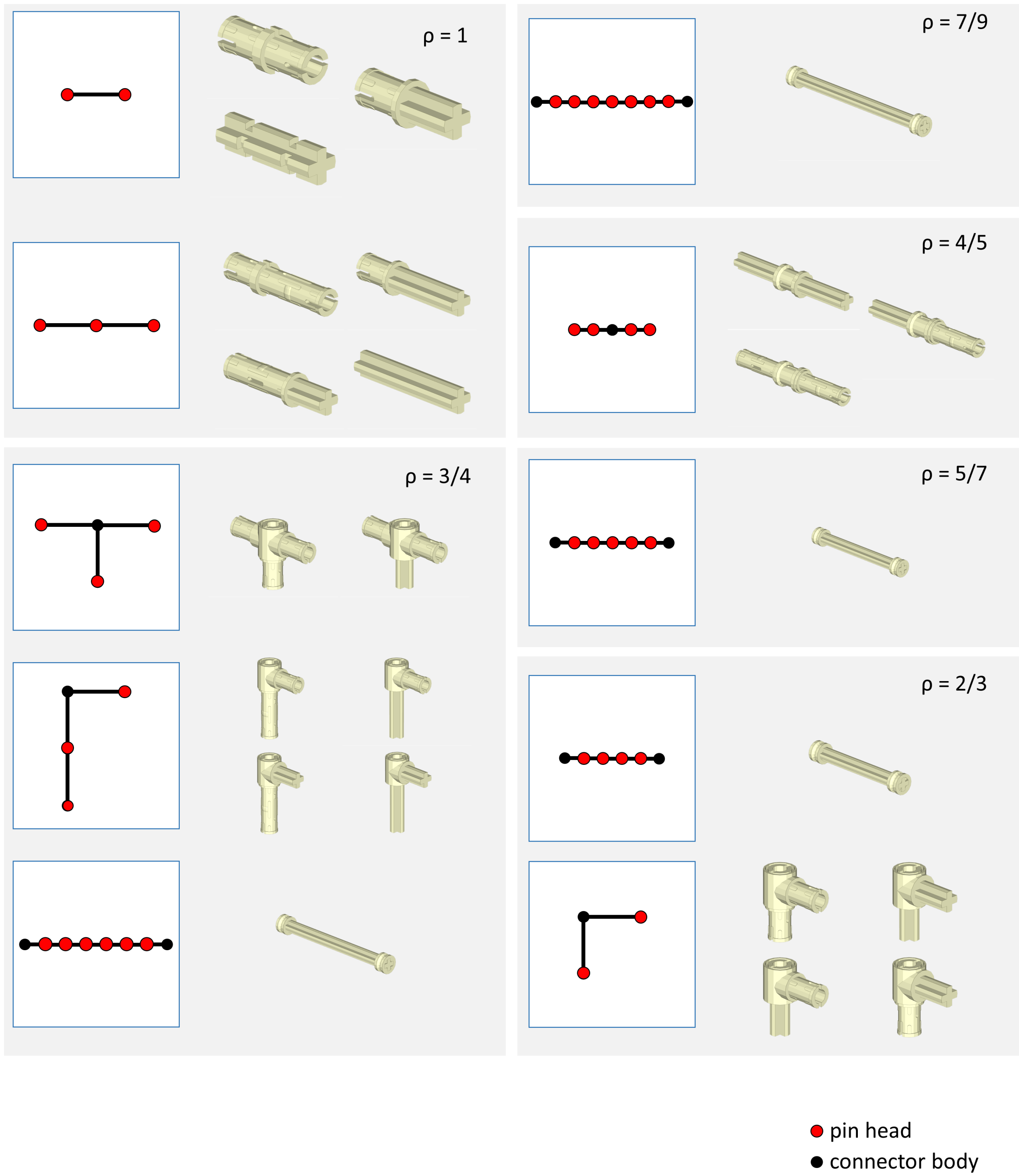
Length (mm)	15.4	15.4	23.3	23.3	23.3
Width (mm)	7.4	7.4	7.4	7.4	7.4
Height (mm)	8.0	8.0	8.0	8.0	8.0
Mass (g)	0.40	0.39	0.50	0.64	0.33

Figure 3. Basic information of connectors in the brick set.

Length (mm)	16.0	16.0	24.0	24.0	24.0
Width (mm)	4.8	4.8	4.8	4.8	4.8
Height (mm)	4.8	4.8	4.8	4.8	4.8
Mass (g)	0.16	0.26	0.31	0.35	0.32

Figure 4. Basic information of pins in the brick set.

Part A.2 - Details on the Connection mechanisms



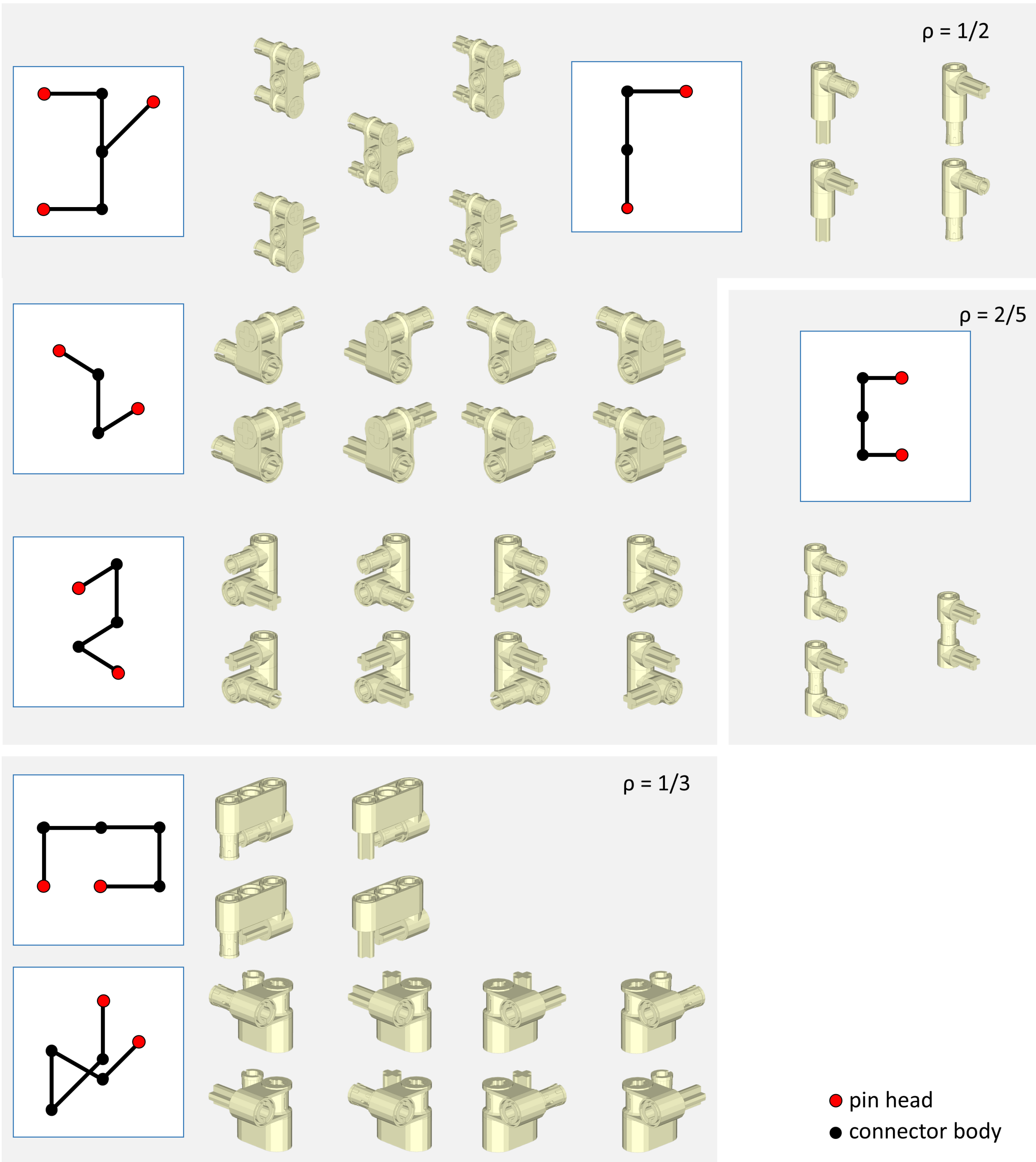


Figure 5. All connection mechanisms supported by our system. We group them by the pin head ratio (ρ), and show their line graphs, with red and black dots to indicate the pin head and connector body, respectively.

Part B - Exploring the Solution Space – How many ways of forming a 9x9 square

In this part, we explore the immenseness of the search space for LEGO Technic constructions. Here, we try to build a simple shape, which is a 9×9 square, by finding out how many ways that we may build the square from straight beams with simple pins; see Figure 6 for some examples. Typically, we only need beams of length 2, 3, 4, 5, 6, 7 and 9.

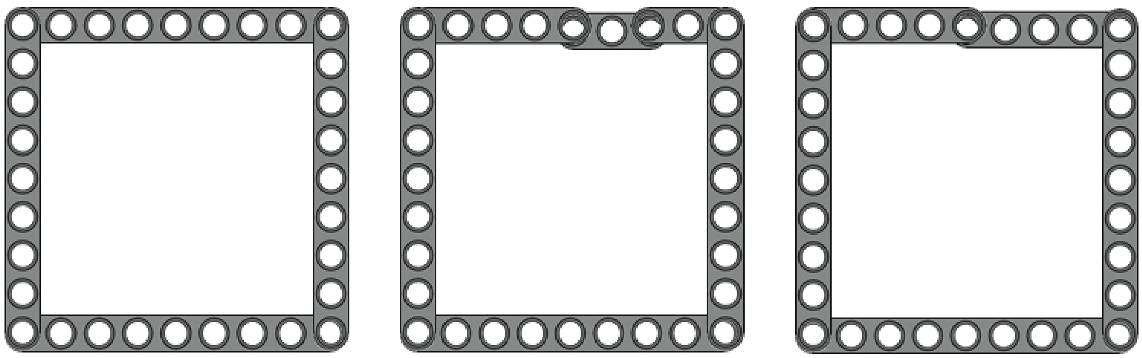


Figure 6. Three example ways of building a 9x9 square.

Building one side of the square. First of all, we try to find out all the ways of forming one side of the square. Figure 7 shows six example ways of building one side of the square from the beam set.

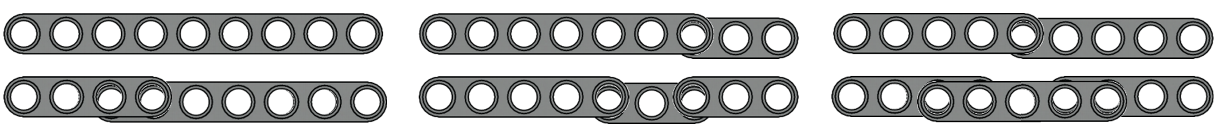


Figure 7. Six example ways of building one side of the square.

To simplify the problem for counting, we only consider layouts, where (i) all the beams in the whole layout are located within two layers (see Figure 4 in main paper); and (ii) adjacent beams are connected only by "one" holes (see the top examples in Figure 7) or only by "one-or-two" holes (see all examples in Figure 7). If we do not consider these restrictions, the search space would be too large for counting.

To get the exact number of ways of forming one side of the square, we wrote a program that enumerating all possible ways from an empty beam set to build a complete side in a depth-first manner, while the layout fulfills the above two restrictions. Figure 8 shows the results. Note further that due to layering, we intentionally enumerate only the layouts whose endpoints are single-layered rather than double-layered; in this way, when we later connect four of these layouts into a square, we can still maintain two-layer-only corners in the generated squares.

Among the generated layouts, some contains even number of beams and some contains odd number of beams. When a layout has an **odd** number of beams, the beam(s) at its two endpoints are in the **same layering level**, while when a layout has an **even** number of beams, the beam(s) at its two endpoints are at **different layering levels**.

One-hole-only connections

#Beams	1	2	3	4	5	6	7	8
Counts	1	5	21	35	35	21	7	1

One-or-two-holes-only connections

#Beams	1	2	3	4	5	6	7	8
Counts	1	9	61	129	129	61	13	1

Figure 8. Number of one-side layouts for one-hole-only (top) and one-or-two-holes-only connections (bottom).

- For the case of one-hole-only connections, there are 1 + 21 + 35 + 7 = **64** layouts with odd number of beams, and 5 + 35 + 21 + 1 = **62** layouts with even number of beams.
- For the case of one-or-two-hole-only connections, there are 1 + 61 + 129 + 13 = **204** layouts with odd number of beams, and 9 + 129 + 61 + 1 = **200** layouts with even number of beams.

Building the square. Next, we are ready to enumerate the 9x9 squares by assembling these one-side layouts. Here, we have the following possible ways of taking the one-side layouts to form a square:

First, we consider the case of one-hole-only connections:

- If all four sides of the squares are built from the one-side layouts of the same layering level at endpoints, we have 64^4 cases.
- If two adjacent sides use the one-side layouts of the same layering level and the other two use the one-side layouts of different layering level, then we have $64^2 \times 62^2$ cases.
- Two opposite sides use layouts of the same layering level and the other pair of opposite sides use layouts of different layering level, so we have $64^2 \times 62^2$ cases.
- All four sides use the one-side layouts of different layering levels, so we have 62^4 cases.

Therefore, we have **63,043,600** cases altogether.

For the case of one-or-two-hole-only connections, we can use the same calculation and find that the total number of cases is computed by 204^4 + $204^2 \times 200^2 \times 2$ + 200^4, there are **6,661,171,456** cases altogether.

Remarks. Note that in the above calculations, we did not ignore the cases that are equivalent under rotation and reflection. However, we also did not count the layouts whose adjacent beams connect with more than two holes, layouts whose beams span more than two layers, and layouts that contain L-shaped beams. Hence, the calculation certainly demonstrates the scale, or the immenseness, of the search space for LEGO Technic constructions, even for such a simple 9x9 model.

Part C - Alternatives of the Layout Modification Operator

Before we finalize the procedure for the layout modification operator presented in Section 5.2 in the main paper, we have explored the following alternative designs and evaluated their efficiency and performance in the solution search framework:

- First, in step (iii) of the layout modification operator, we have tried to choose candidate placements with equal probability and also tried to remove only the beams that cover (but not touch) the selected edge in step (ii). The former case has a time performance issue, since the solution search process becomes too slow, even for simple models. For the latter case, the modification becomes so small that we can hardly find any good solution that minimizes the objective function.

- On the other hand, we tried to remove more nearby beams in step (ii) by considering more neighboring beams further away from the selected edge. However, the local modification becomes too large that the search process can hardly converge.

- Furthermore, we tried to divide our operator into two separate operators, i.e., one to remove beams and the other to add beams, then in our framework, we apply them at random. However, the intermediate beam layout becomes mostly incomplete as a result. Although we further tried to add another term in the objective function to explicitly maximize the model completeness, it is still hard for the search process to converge.

Part D - Details on the Balance, Stress, and Assemblability analysis

Our tool provides further analysis on the LEGO Technic model generated by the search framework:

Balance analysis Our tool stores the mass and center of gravity (CG) of each brick in the supported brick set, so it can compute the CG of the generated model and check if the CG falls within the support polygon (2D convex hull of the contact points on the ground) when projected along the gravity [**???**]. Hence, we can estimate if the generated model can stand on its own; see Figure 14 (top) in the main paper for an example.

Stress analysis We employ ANSYS R19.0 (Academic) Mechanical, a professional software, to analyze the stress distribution on the generated LEGO Technic models. To do so, our tool can automatically mesh and script the generated model as input to ANSYS by representing it as an ANSYS beam structure with polyethylene as the material: density $950kg/m^3$ and Young's modulus 1.1GPa. By further defining the supporting ground, weight, and external force (if any), we can use ANSYS to compute the combined stress in the structure and generate stress plots; see Figure 14 (bottom) in the main paper for example results. Note that if there are insufficient ground supports or the structure cannot bear the external forces, ANSYS will not output analysis results.

Assembly analysis If a model is assemblable, we should be able to progressively disassemble it into individual bricks. Hence, starting from a complete assembly, our tool iteratively looks for a brick that can be taken out from the model without colliding with the bricks that are still remained in the model. Note that sometimes, if no single brick can be removed, our tool will look for a group of removable bricks, which are usually related to the connection mechanisms. For example, if we insert the U-shaped mechanism (see the ρ=2/5 group in Figure 5 in the main paper) to the layout, we have to unplug the whole U-shaped mechanism (a group of three bricks) before disassembling individual bricks in the group. By then, we can continue to remove more bricks from the assembly.

By this means, our tool can determine the assemblability of a generated model (i.e., self-collision in the assembly process) and produce an assembly sequence that further respects the model's symmetry. In the end, our tool also generates scripts for rendering model assembly videos using the KeyShot software (see Figure 9 below) and for producing LEGO-style assembly instructions using the LPub3D software [**?**]; see the last part in this Supplementary document for examples.

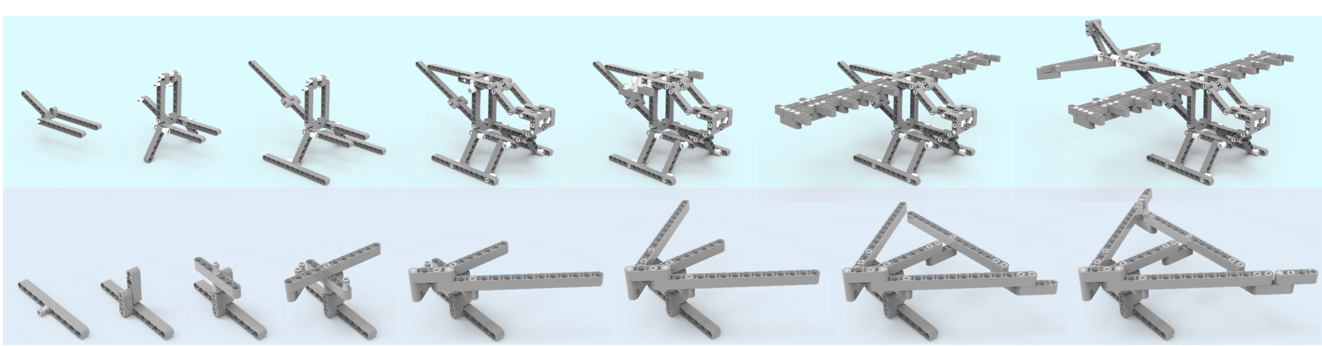


Figure 9. Snapshots taken from the generated assembly sequence of AIRPLANE (top) and FLYING_KITE (bottom).

Part E - More Results on Robustness of Our Method to Brick set

Figures 10 and 11 show results produced by our method by iteratively removing the longest beams, until only a single beam remains in the brick set. Figure 12 shows results produced by our method by iteratively removing the longest straight beams from the brick set, until only two L-shaped beams remain.

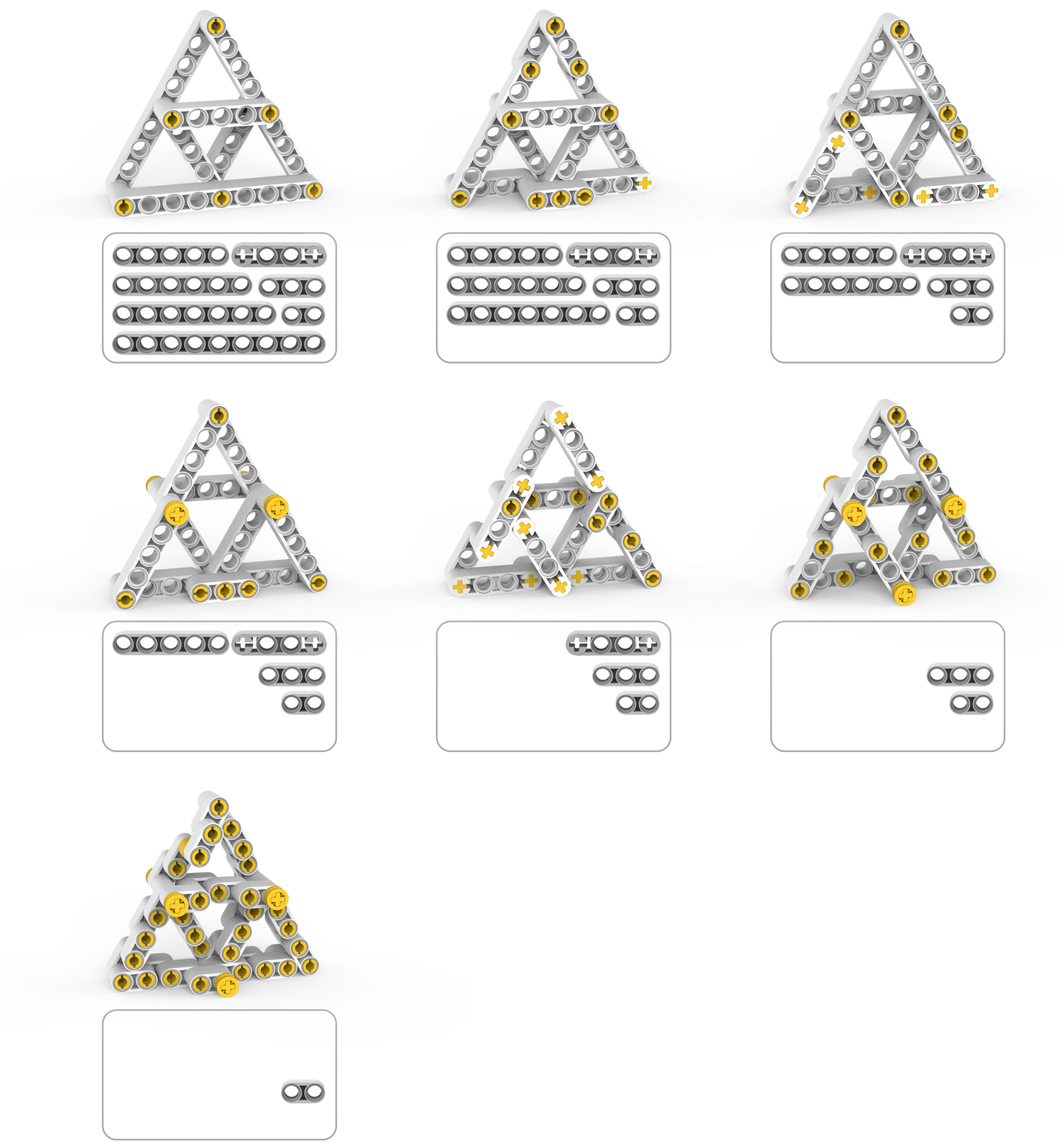


Figure 10. Nested triangles generated using different brick sets.

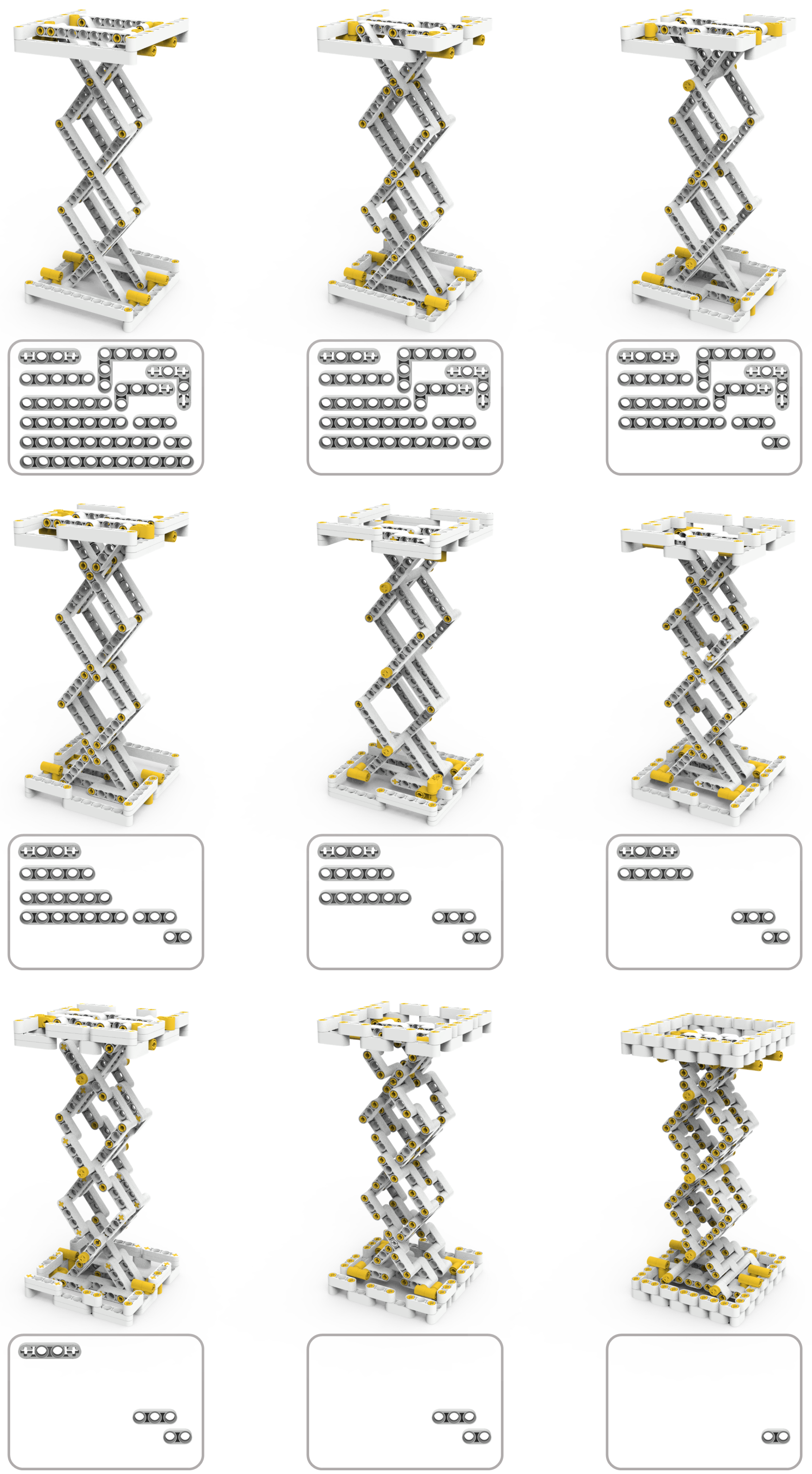


Figure 11. The LIFTER model generated using different brick sets.

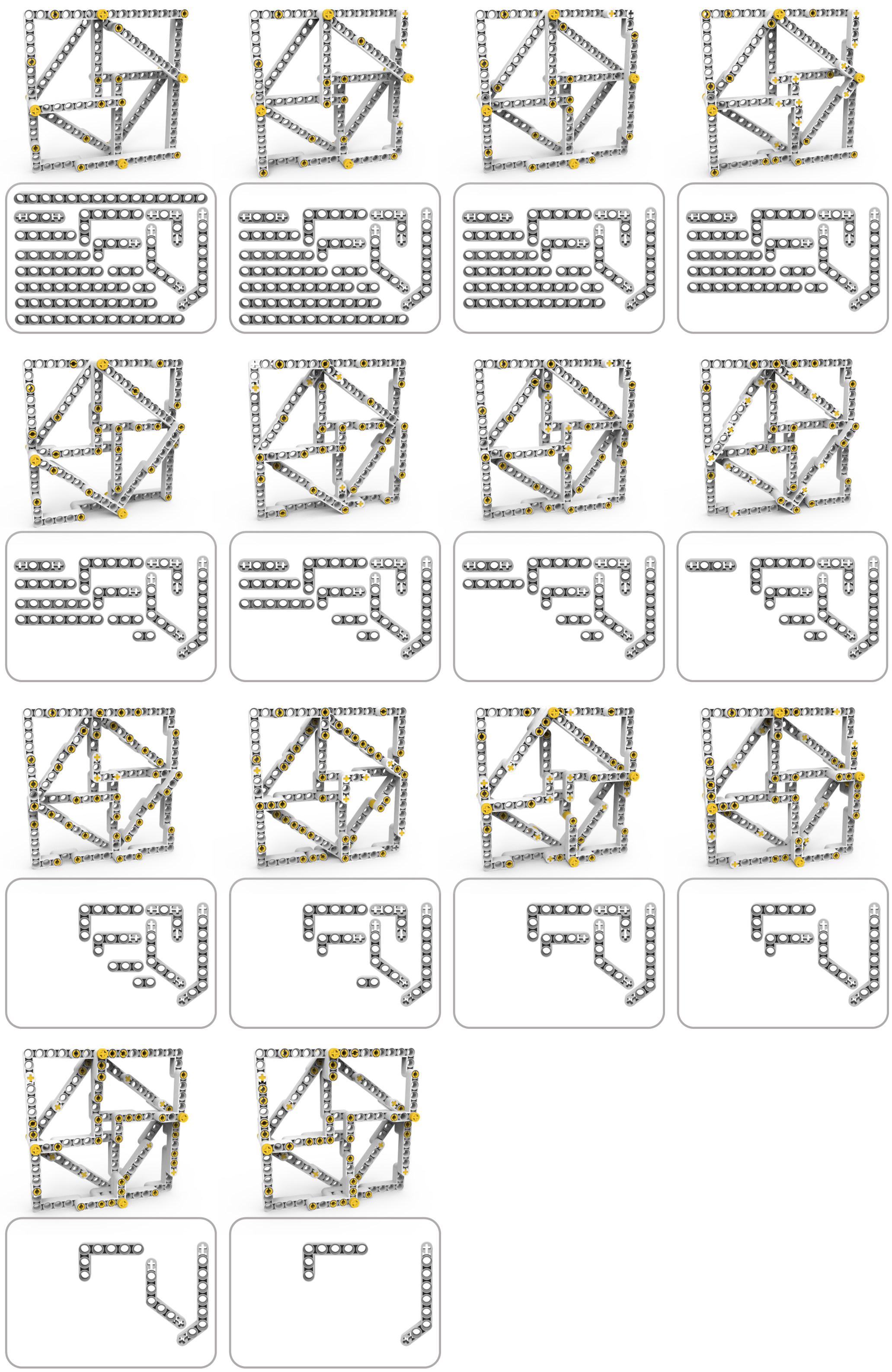
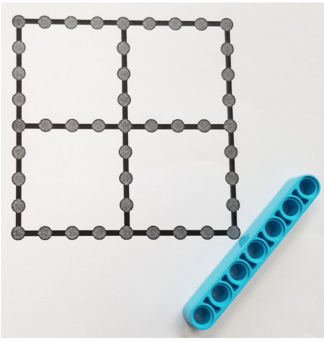


Figure 12. Windmill patterns generated using different brick sets.

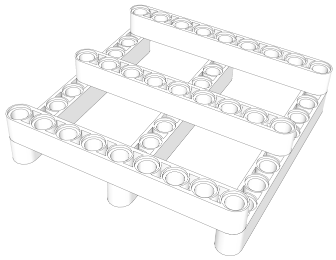
Part F - Details on the Human Performance

Two-dimensional grids designed by our method, as well as by the ten participants (see the “manual design” paragraph in Section 6 of the main paper). Here, we show the designed model, number of beams, number of gaps, and time taken to construct each of the model.



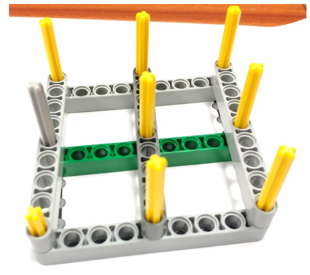
n=2

Input



#Beams:6
#Gaps:0
Total time:2.48s

Computer

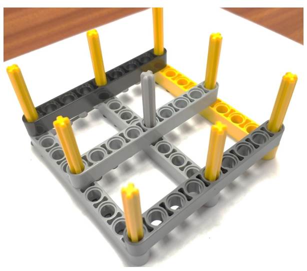


#Beams:6
#Gaps:0
Total time:45s

User #1

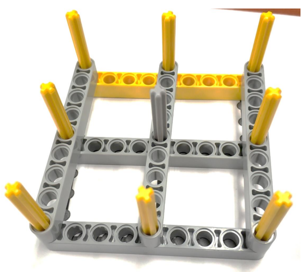
#Beams:6
#Gaps:0
Total time:27s

User #2



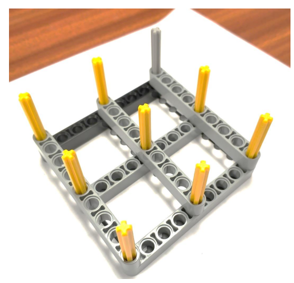
#Beams:6
#Gaps:0
Total time:194s

User #3



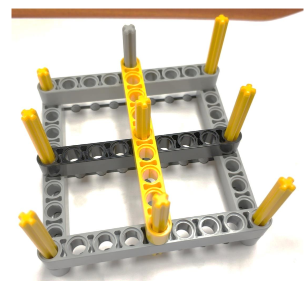
#Beams:6
#Gaps:0
Total time:45s

User #4



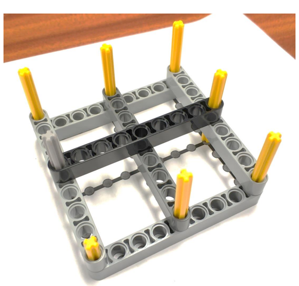
#Beams:6
#Gaps:0
Total time:20s

User #5



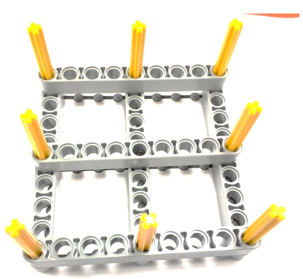
#Beams:6
#Gaps:0
Total time:60s

User #6



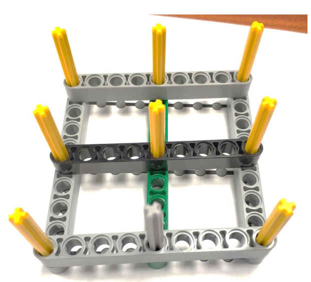
#Beams:6
#Gaps:0
Total time:200s

User #7



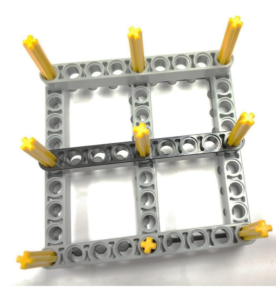
#Beams:6
#Gaps:0
Total time:20s

User #8



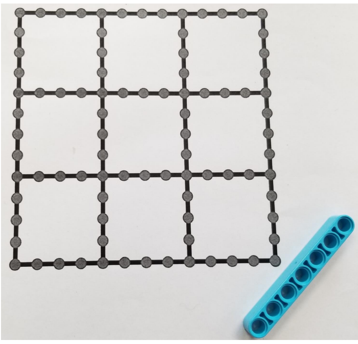
#Beams:6
#Gaps:0
Total time:20s

User #9



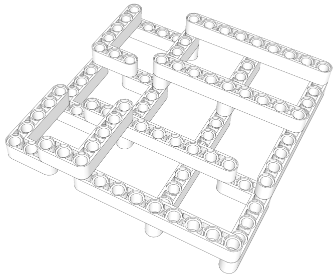
#Beams:6
#Gaps:0
Total time:25s

User #10



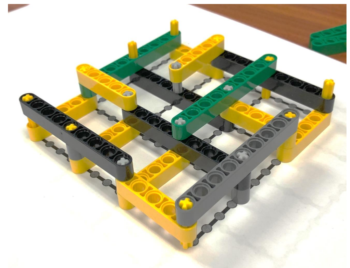
n=3

Input



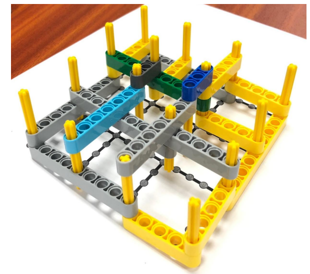
#Beams:15
#Gaps:0
Total time:5.84s

Computer



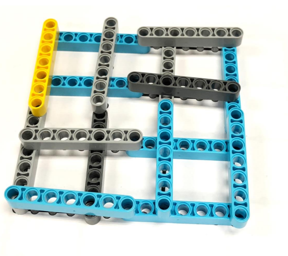
#Beams:17
#Gaps:3
Total time:23mins

User #1



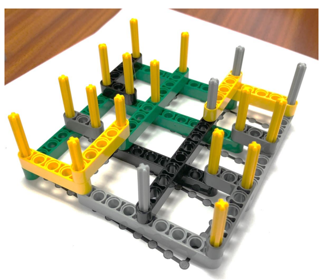
#Beams:18
#Gaps:4
Total time:23mins

User #2



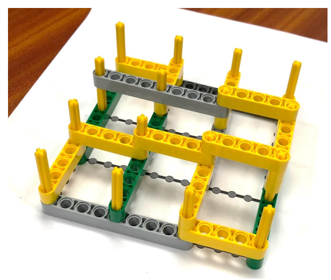
#Beams:16
#Gaps:0
Total time:19mins

User #3



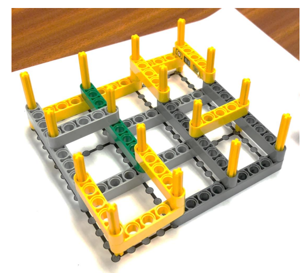
#Beams:16
#Gaps:0
Total time:40mins

User #4



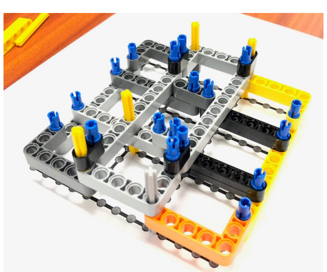
#Beams:20
#Gaps:3
Total time:9.5mins

User #5



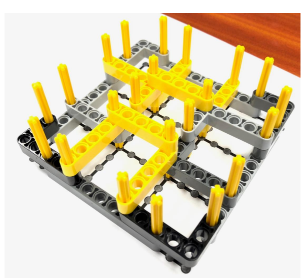
#Beams:16
#Gaps:0
Total time:8.9mins

User #6



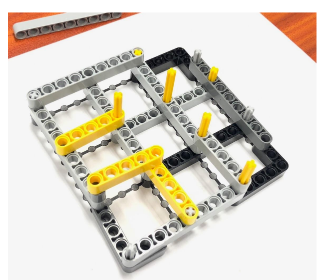
#Beams:20
#Gaps:0
Total time:35mins

User #7



#Beams:18
#Gaps:0
Total time:23mins

User #8

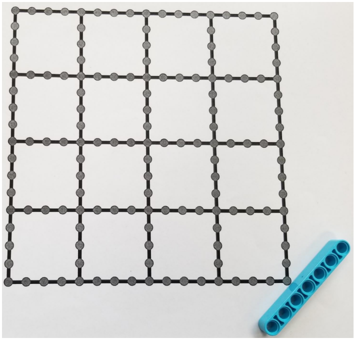


#Beams:15
#Gaps:1
Total time:12mins

User #9

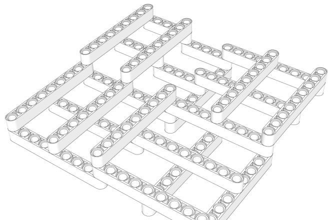
#Beams:25
#Gaps:0
Total time:23mins

User #10



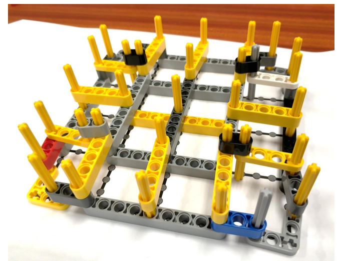
n=4

Input



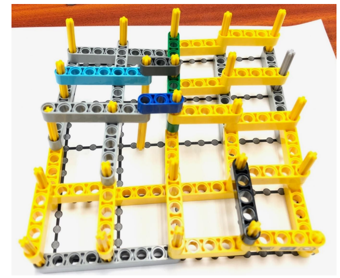
#Beams:22
#Gaps:0
Total time:8.63s

Computer



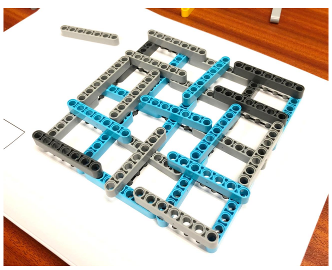
#Beams:38
#Gaps:4
Total time:19mins

User #1



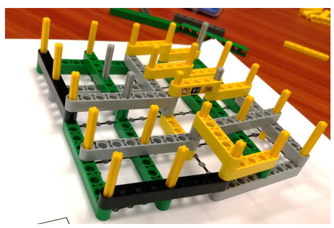
#Beams:32
#Gaps:9
Total time:9mins

User #2



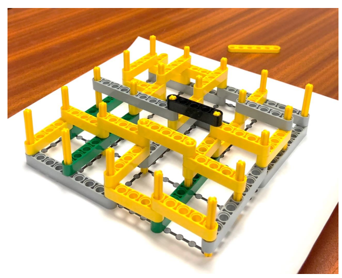
#Beams:30
#Gaps:0
Total time:9mins

User #3



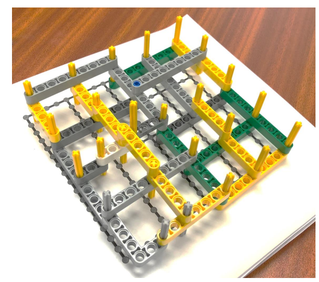
#Beams:25
#Gaps:0
Total time:22mins

User #4



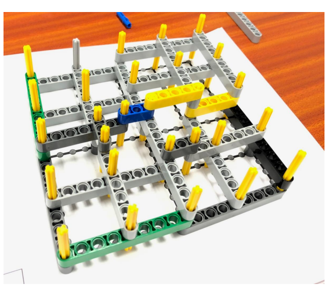
#Beams:34
#Gaps:5
Total time:6mins

User #5



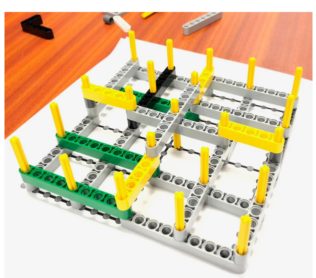
#Beams:27
#Gaps:2
Total time:28mins

User #6



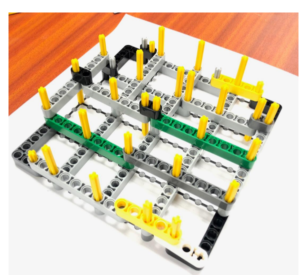
#Beams:24
#Gaps:0
Total time:60mins

User #7



#Beams:22
#Gaps:2
Total time:6mins

User #8



#Beams:28
#Gaps:0
Total time:21mins

User #9

#Beams:29
#Gaps:0
Total time:9mins

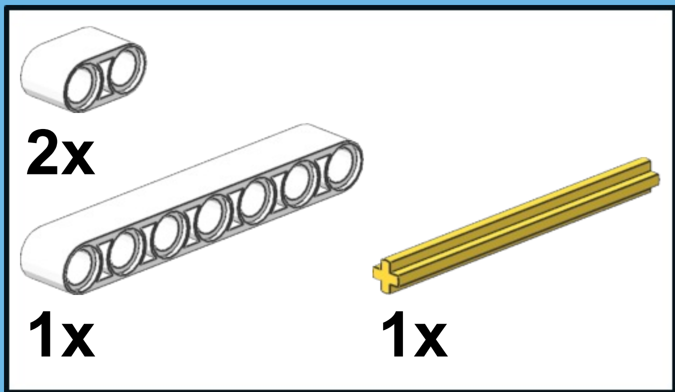
User #10

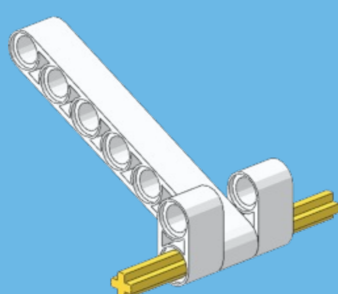
Part G - Example Assembly Instruction Manuals

Please turn over to see the example instruction manuals.

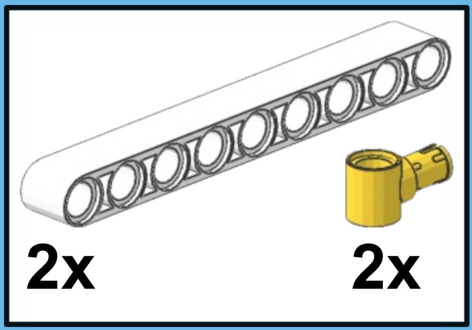

Computational LEGO Technic Design
Assembly Instructions:
Airplane
208 bricks
36cm x 31cm x 15cm

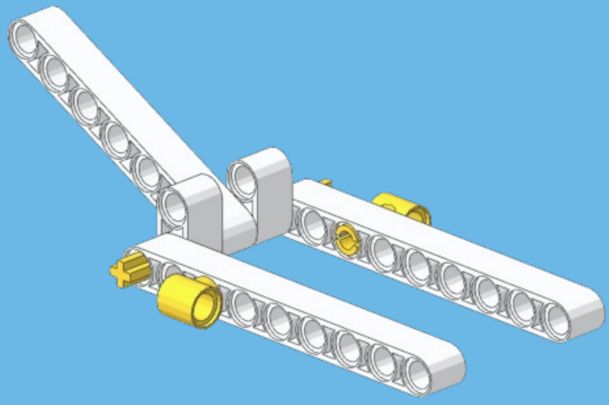
1



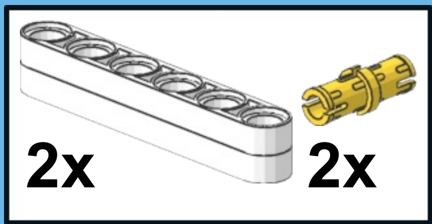


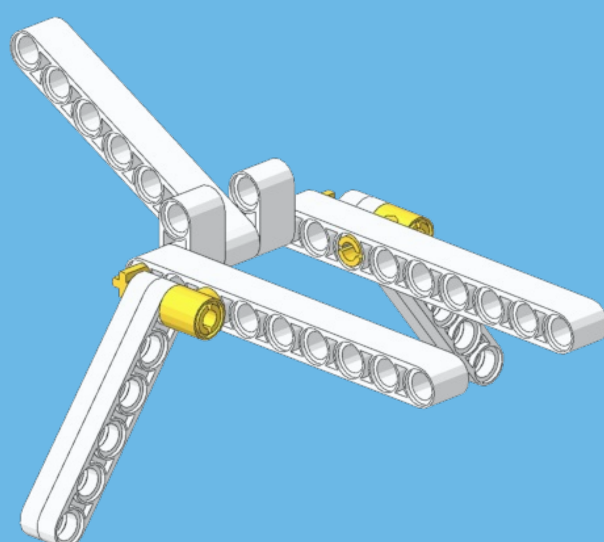
2



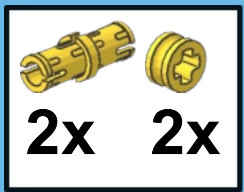


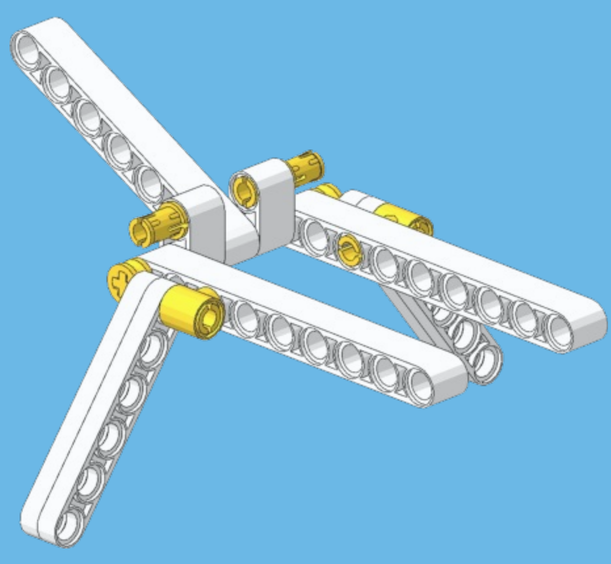
3





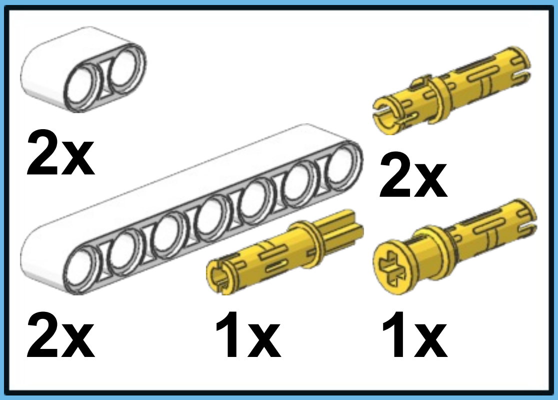
4

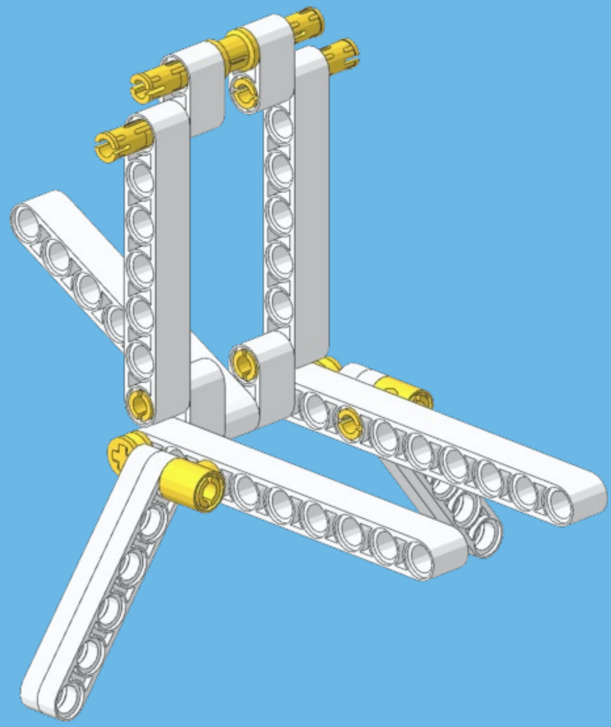




4

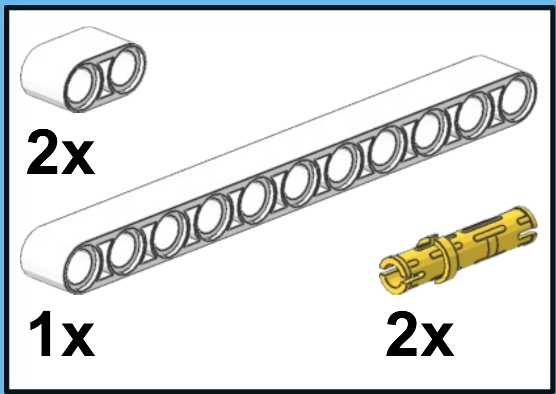
5

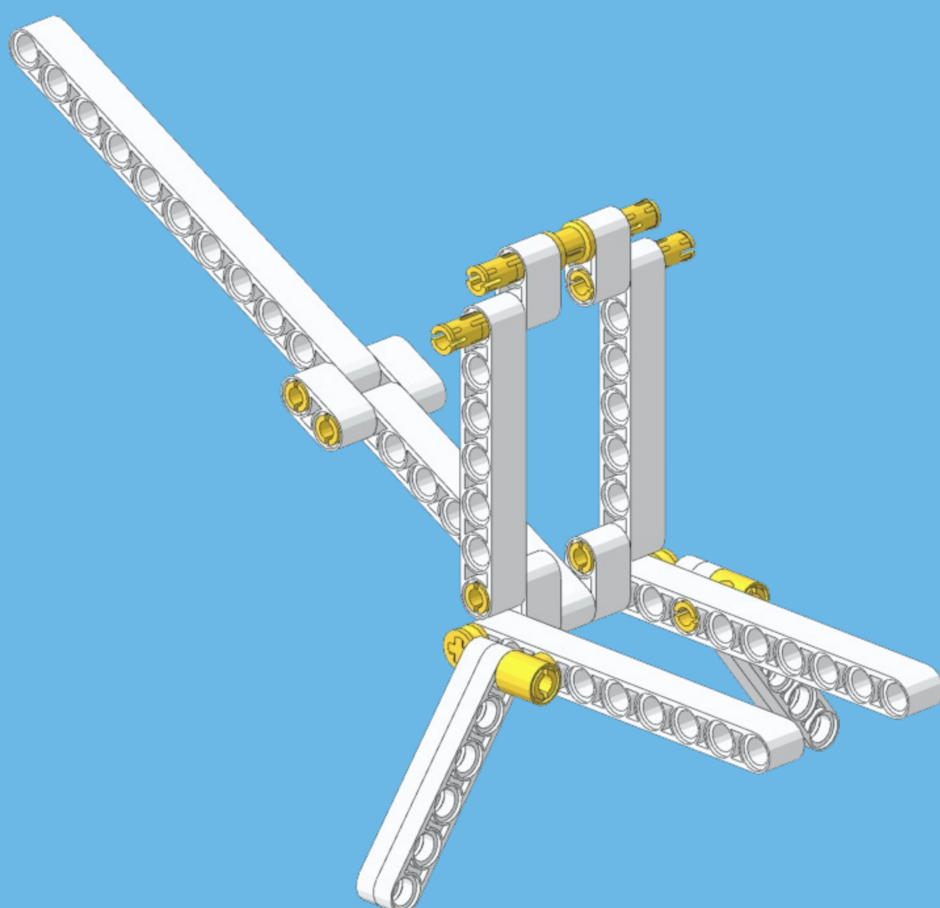




5

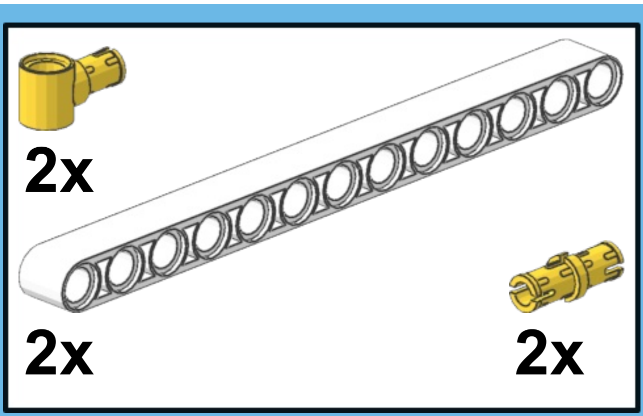
6

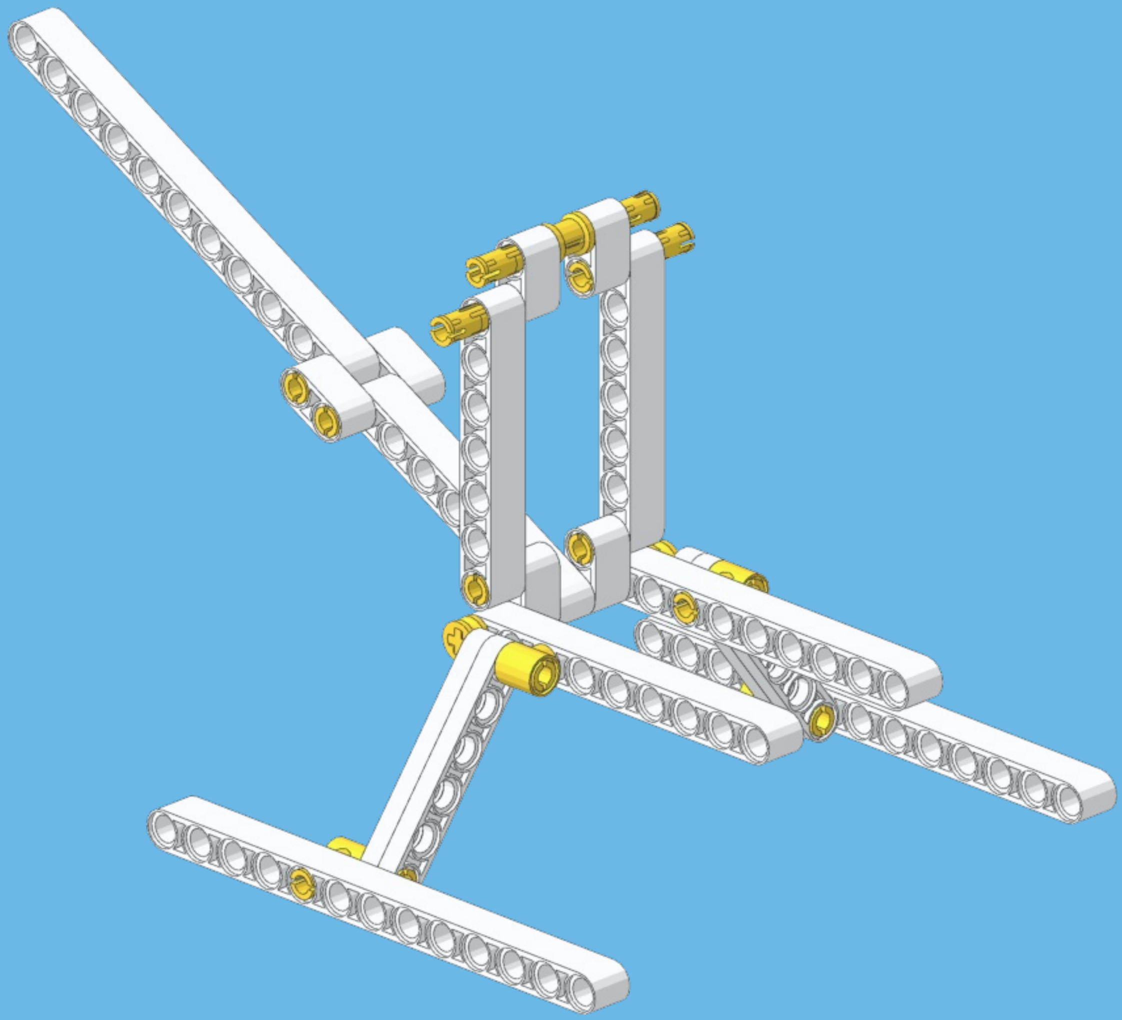




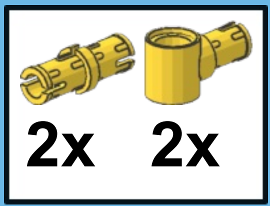
6

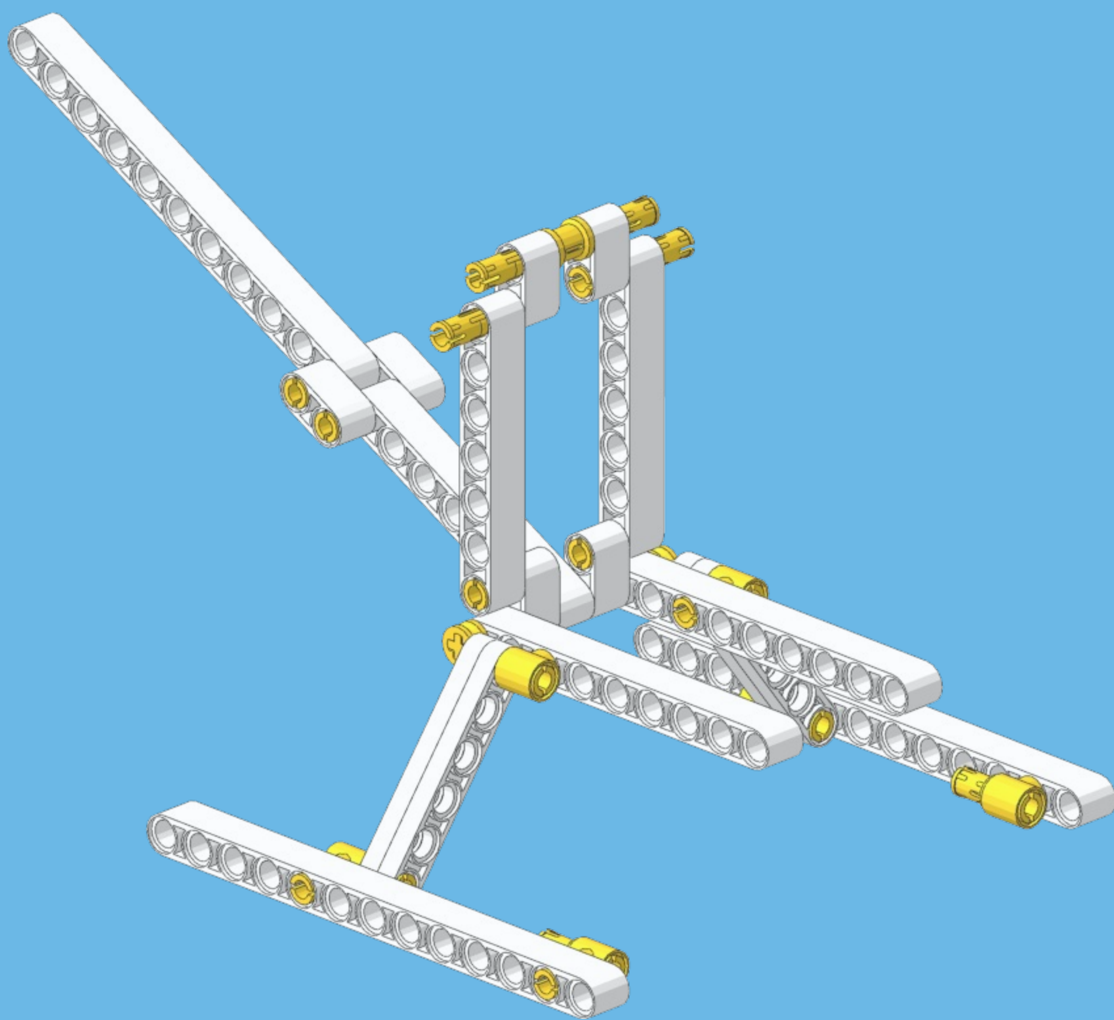
7





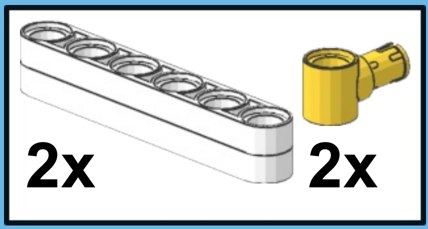
8

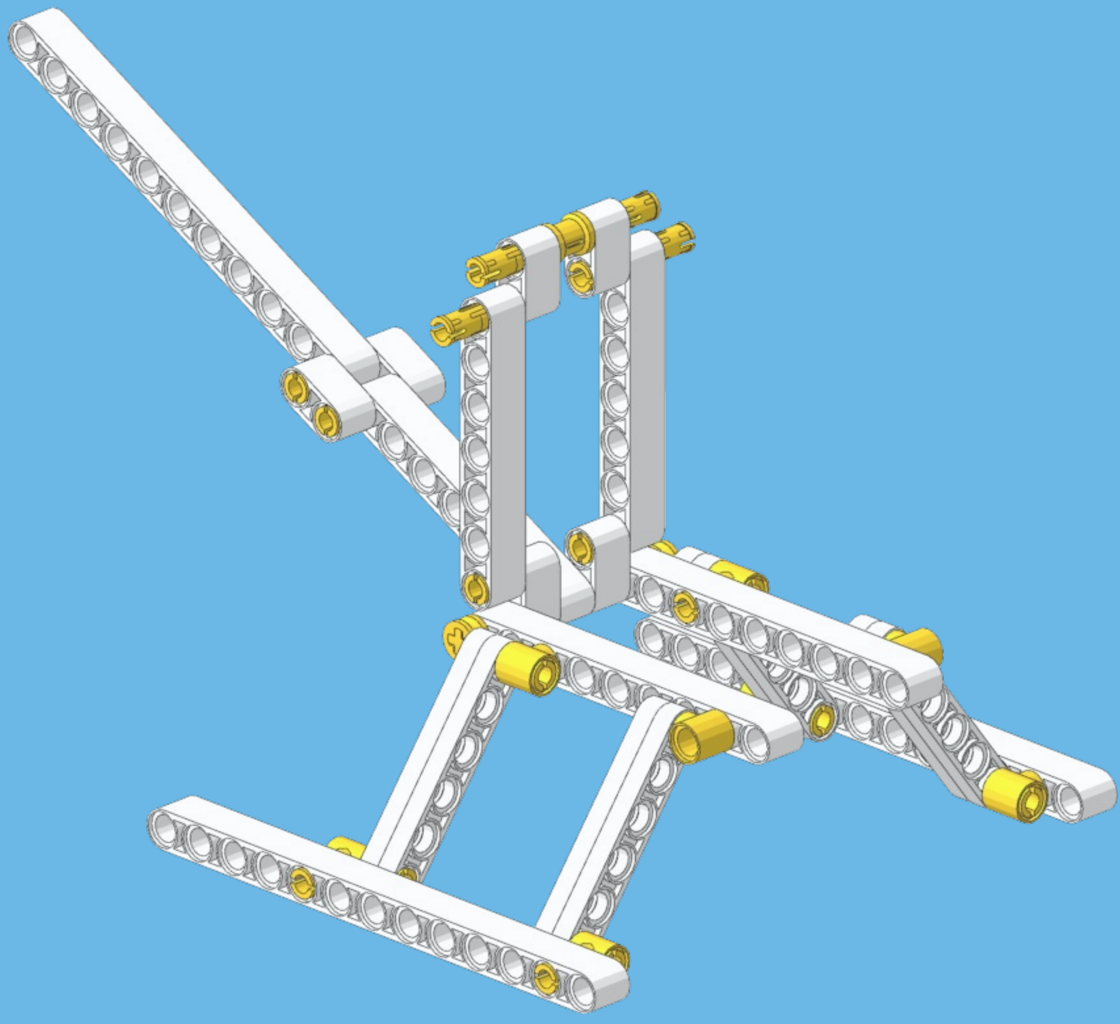




8

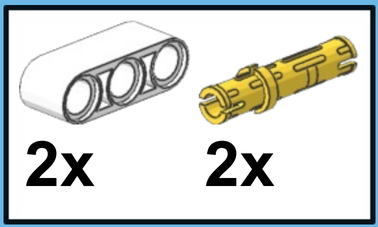
9

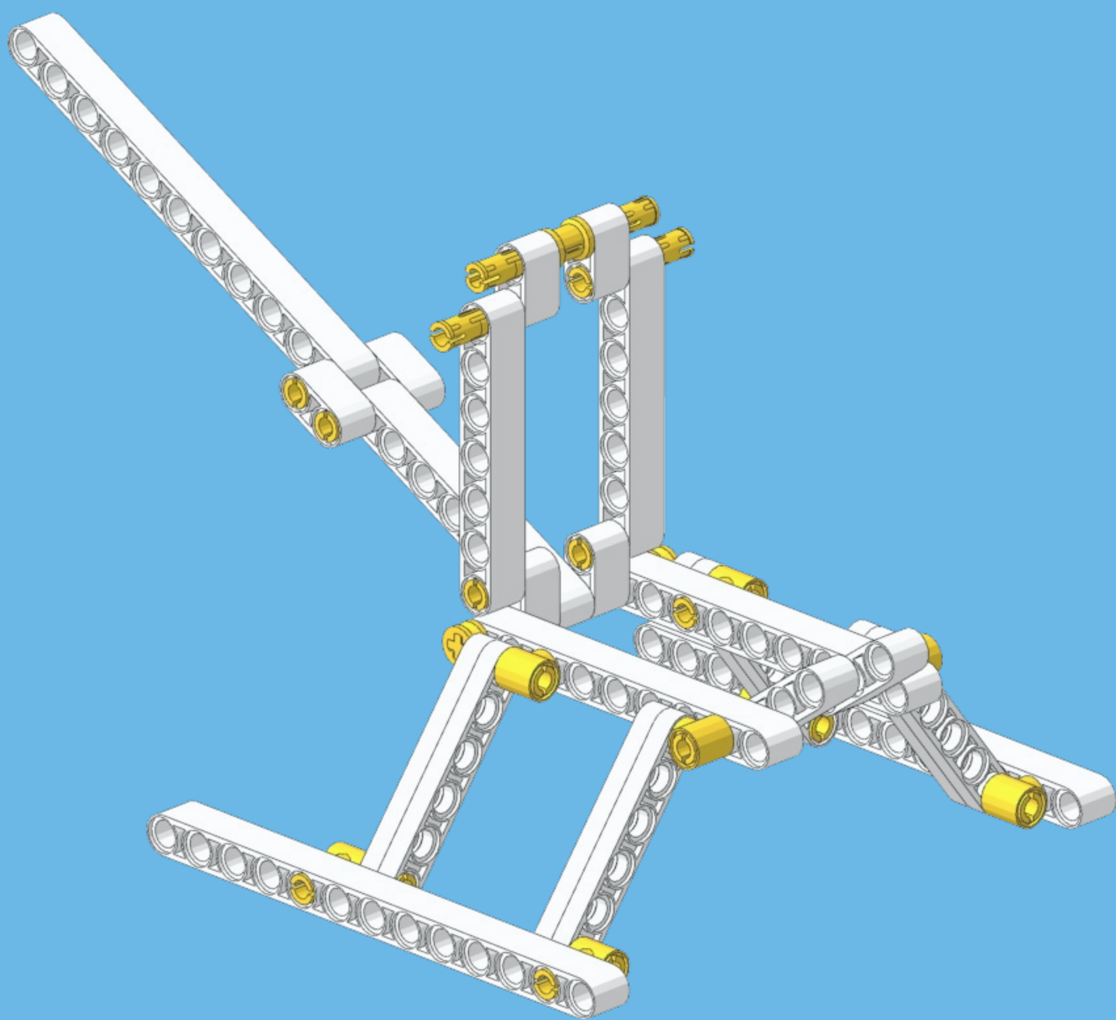




9

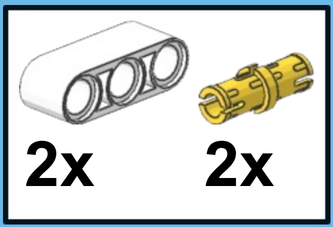
10

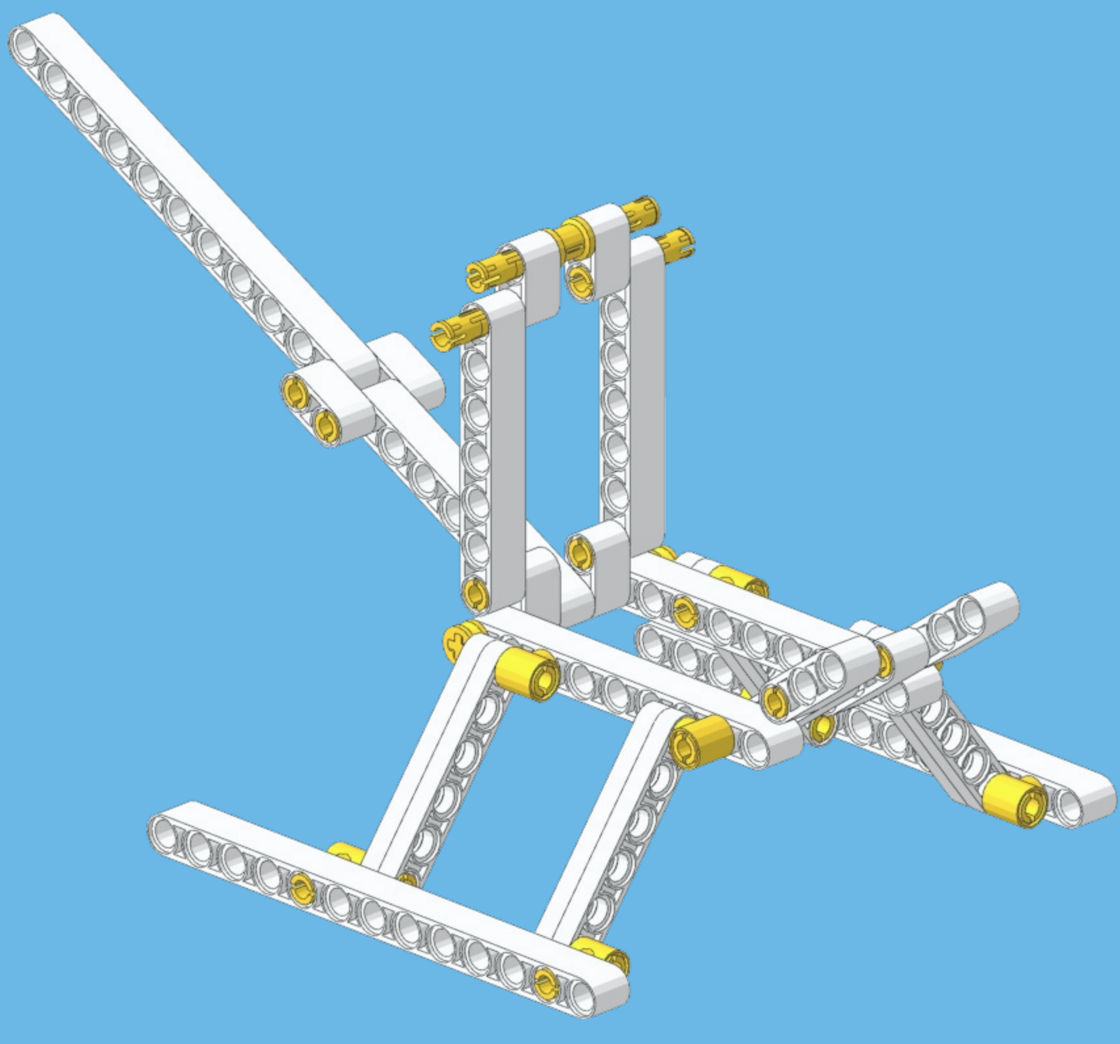




10

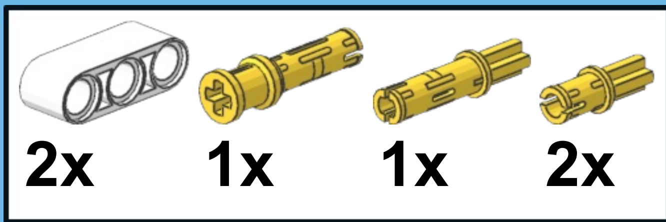
11

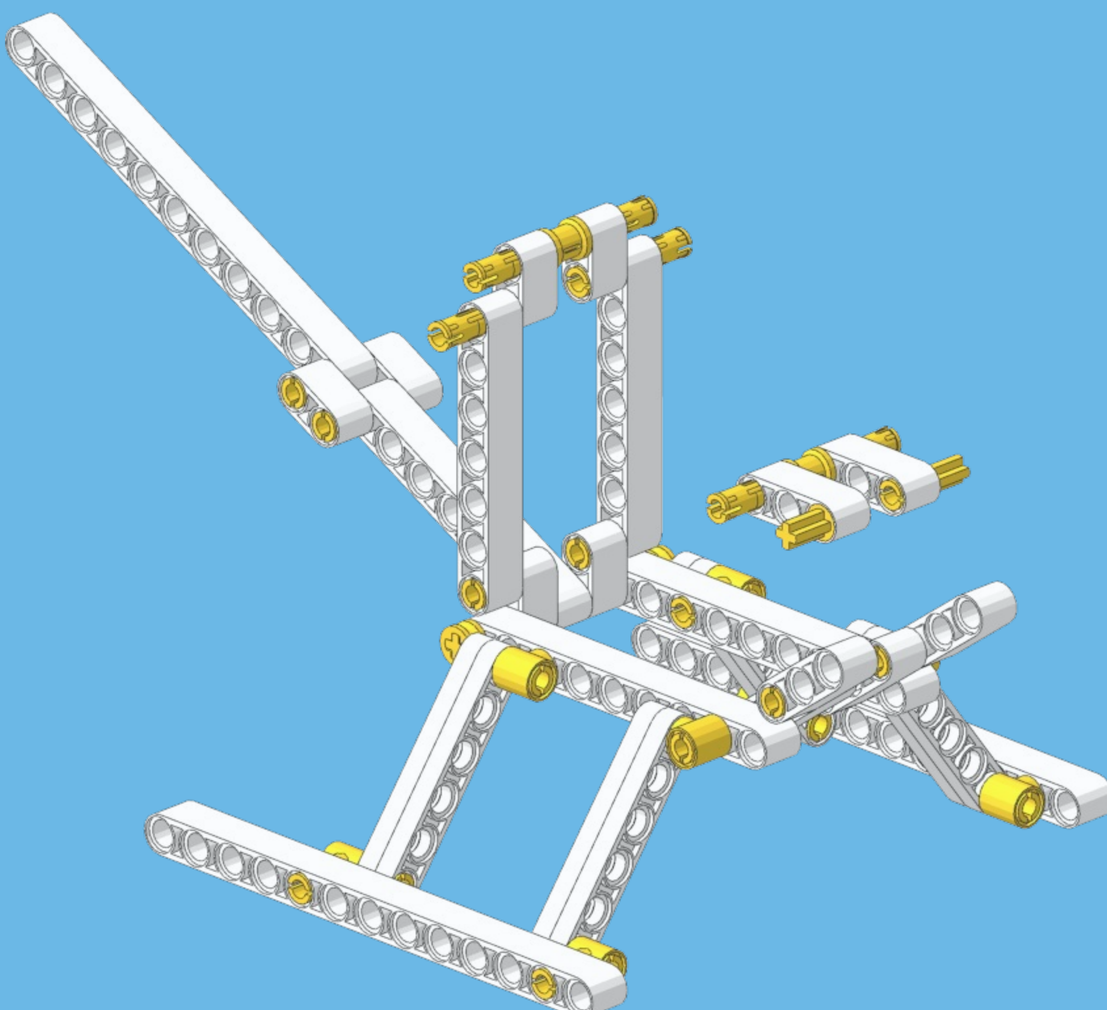




11

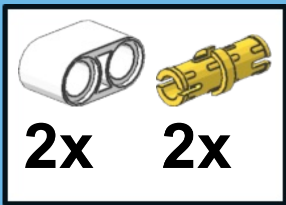
12

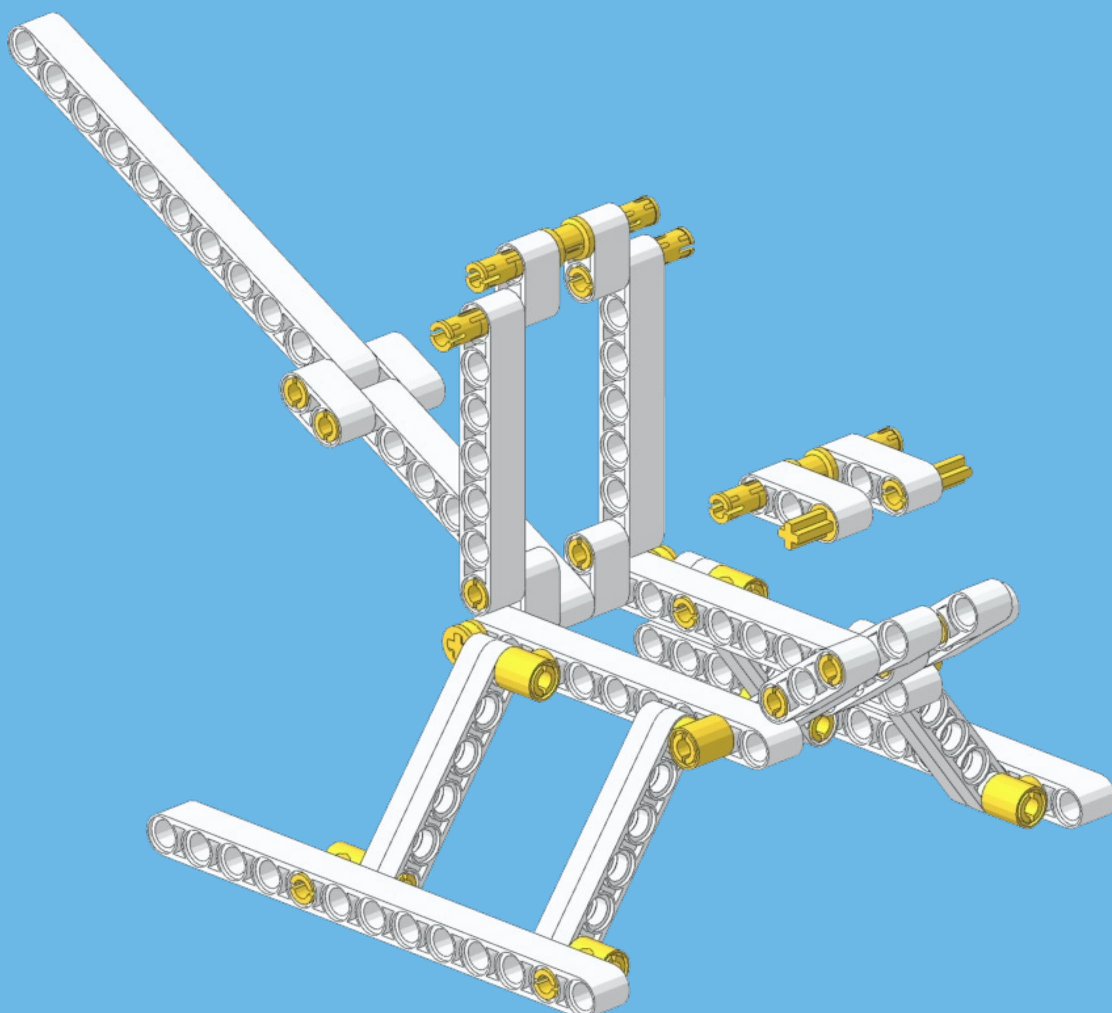




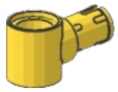
12

13



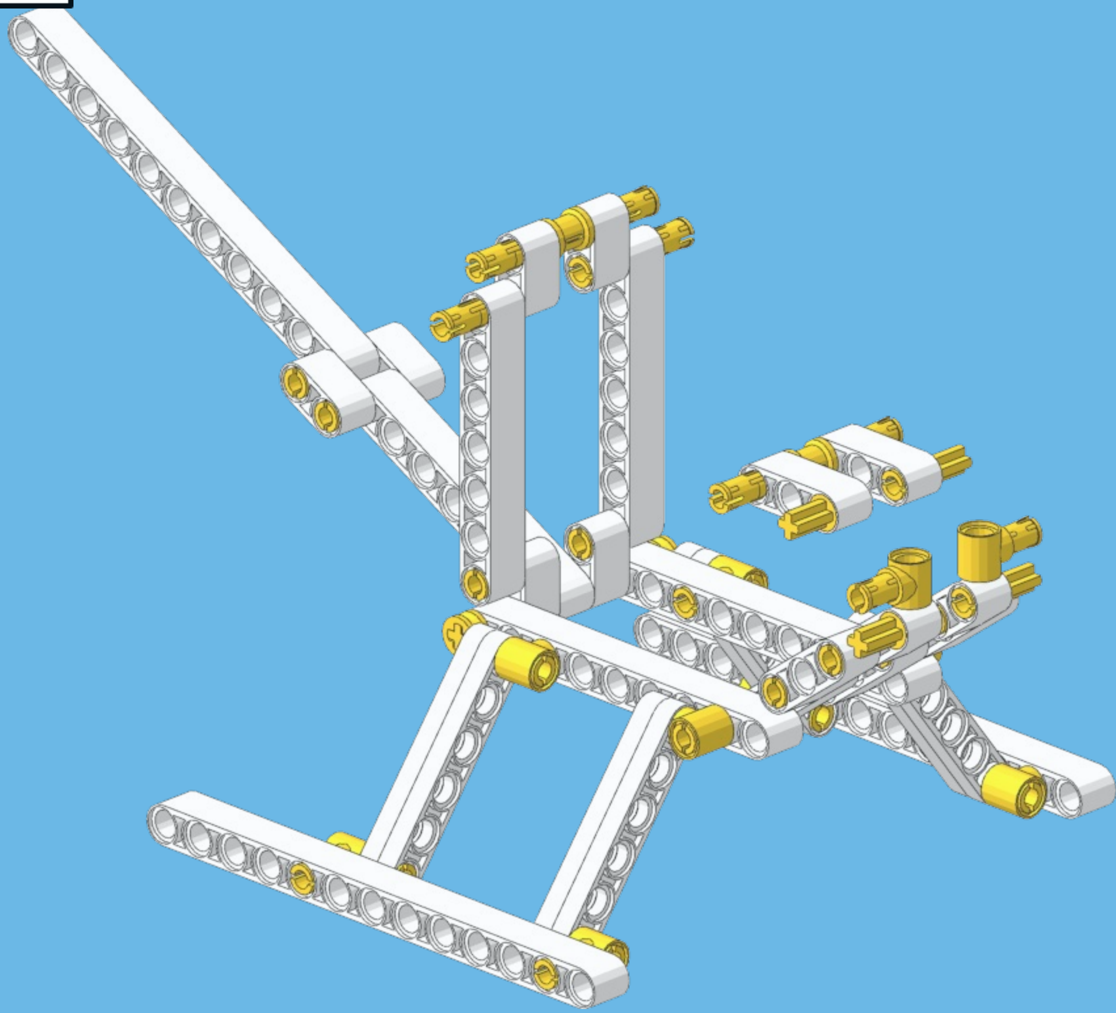


14



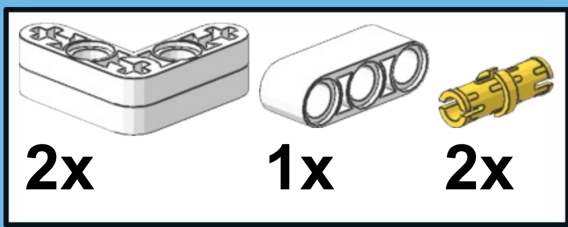
2x

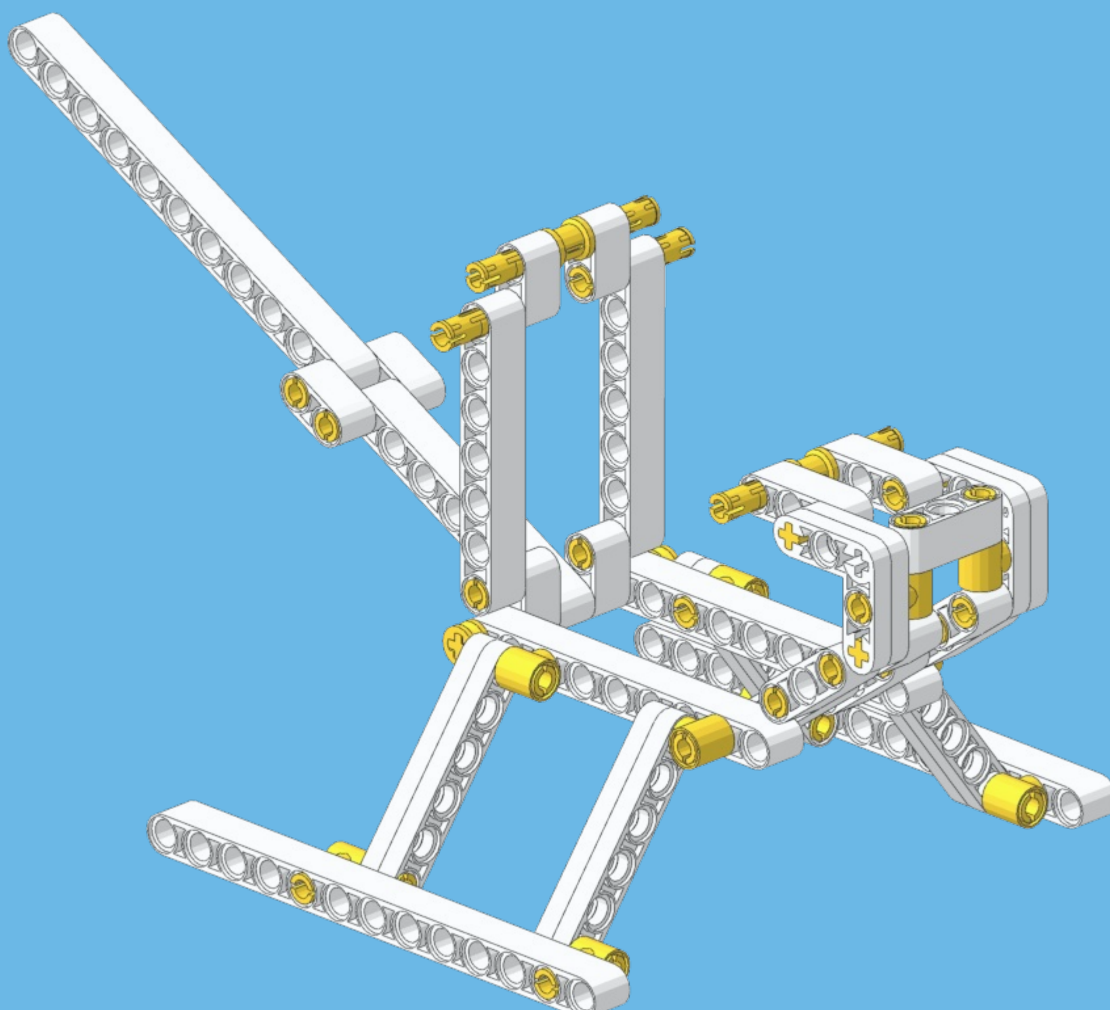
2x



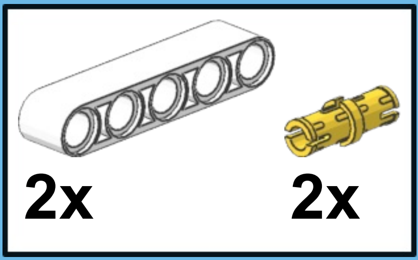
14

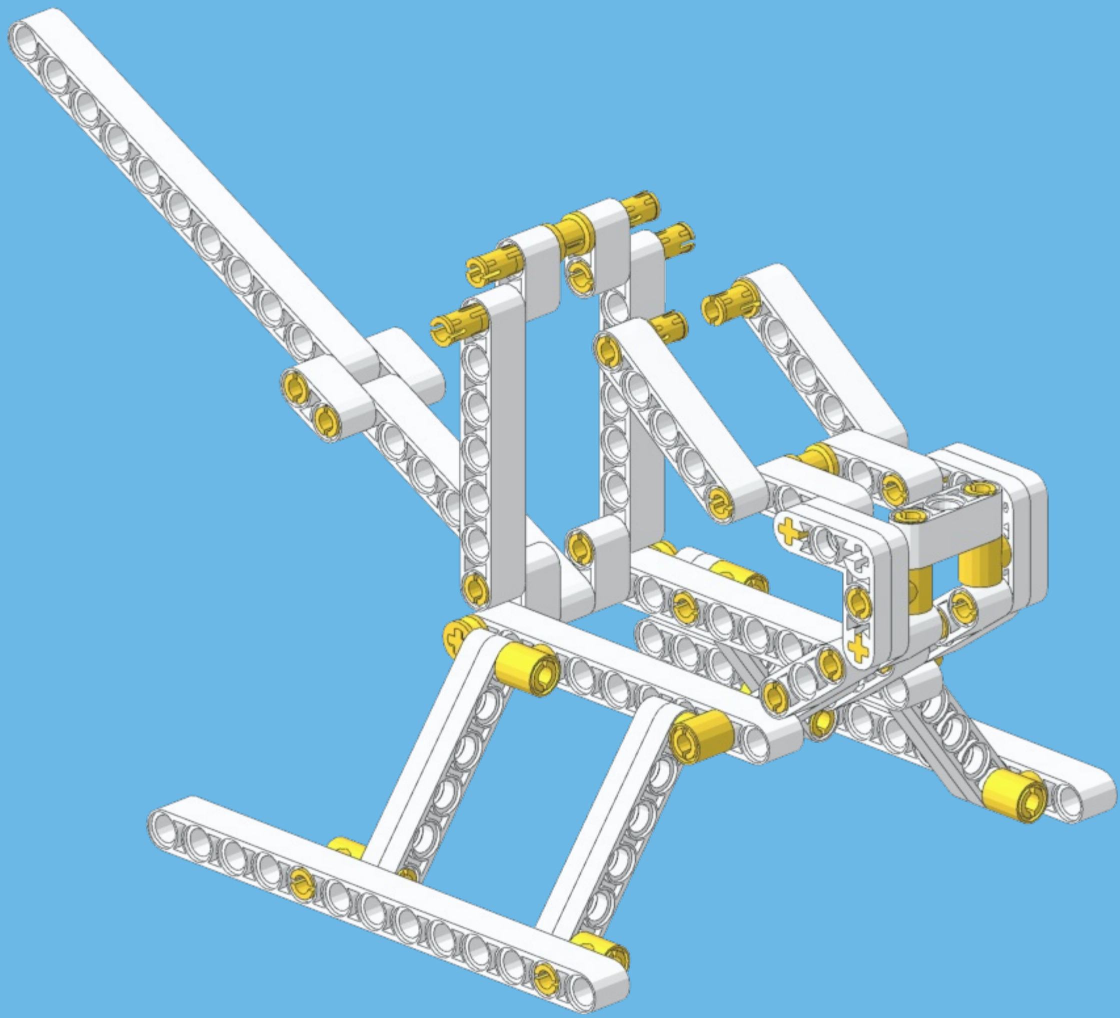
15





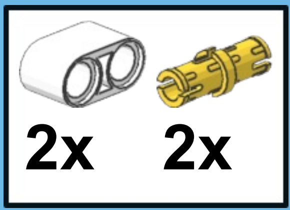
16

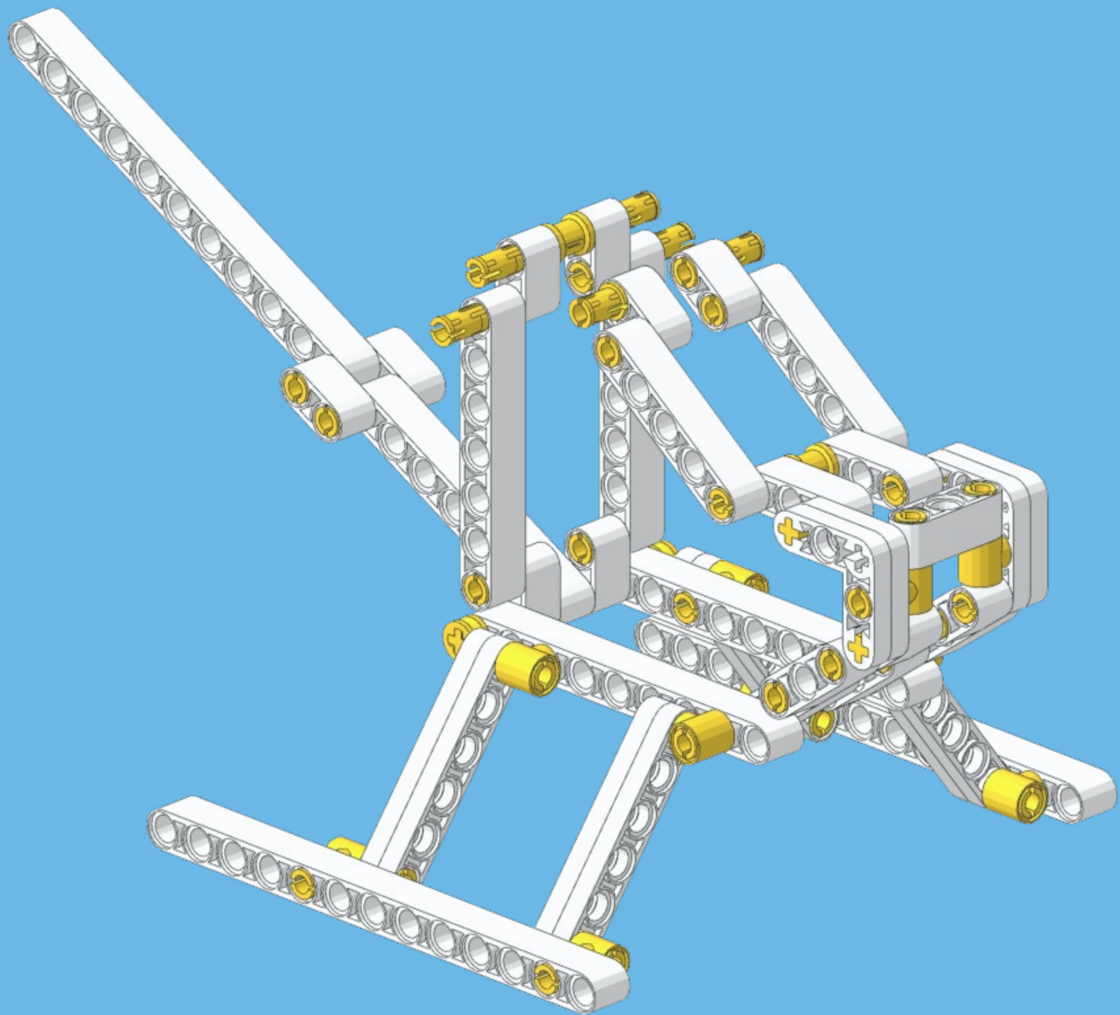




16

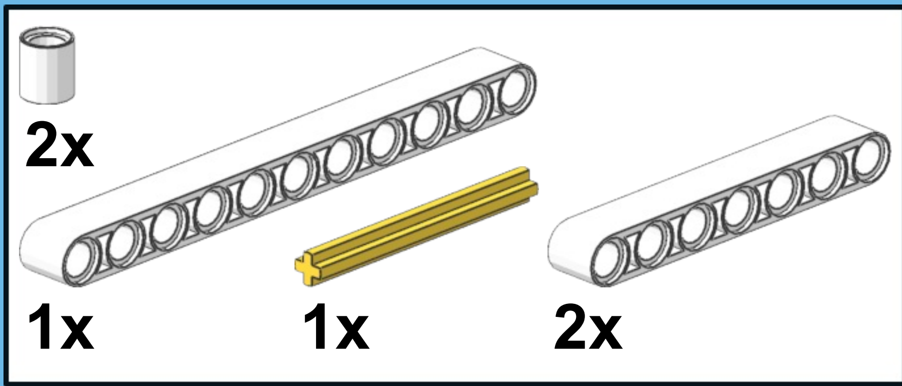
17

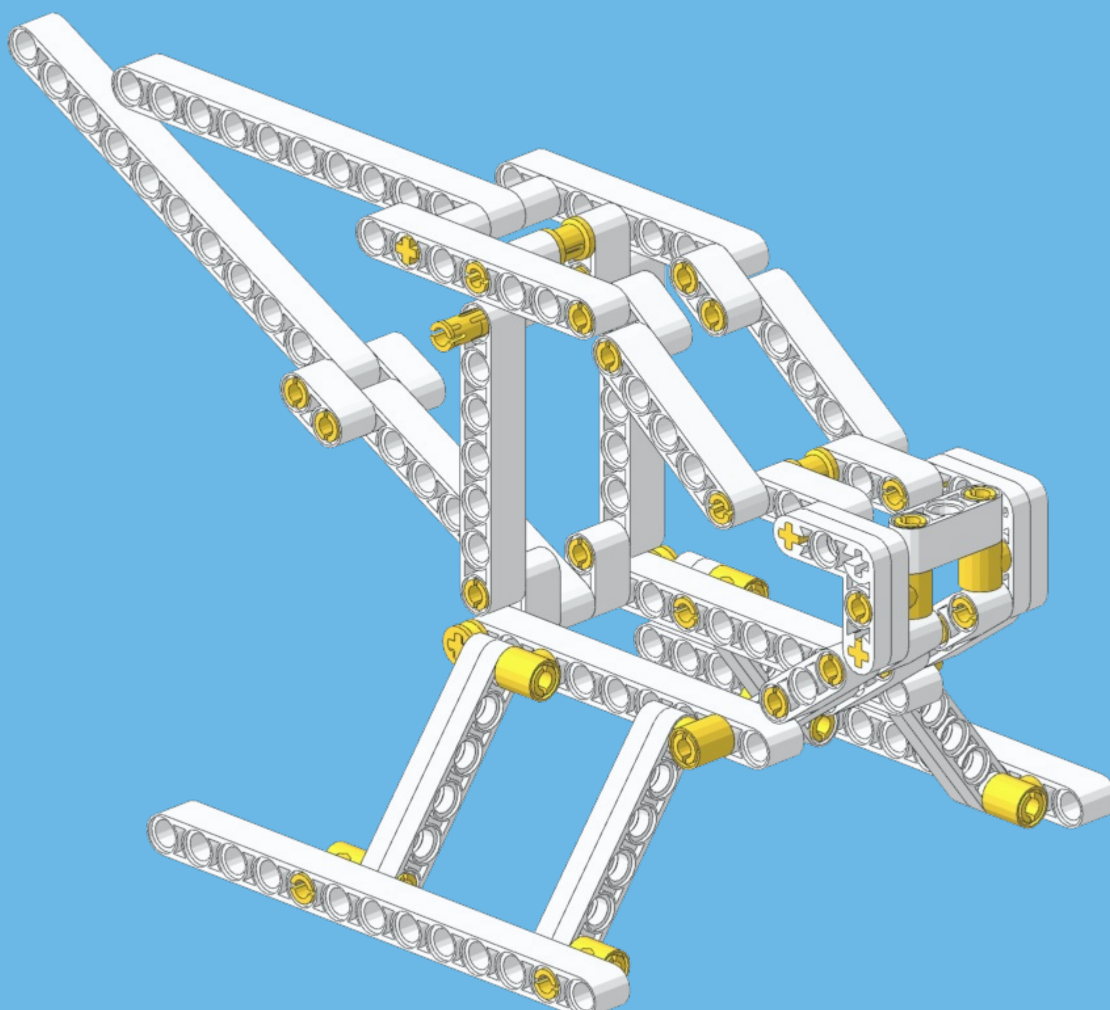




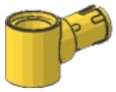
17

18

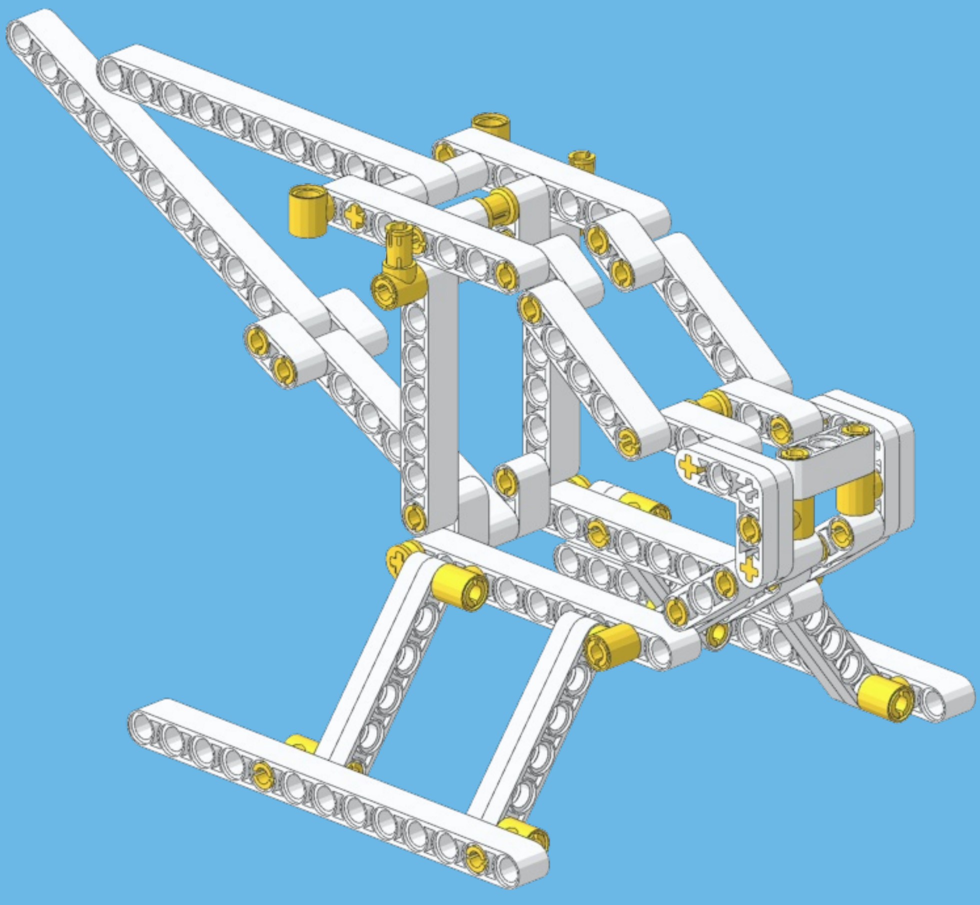




19

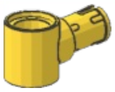


4x



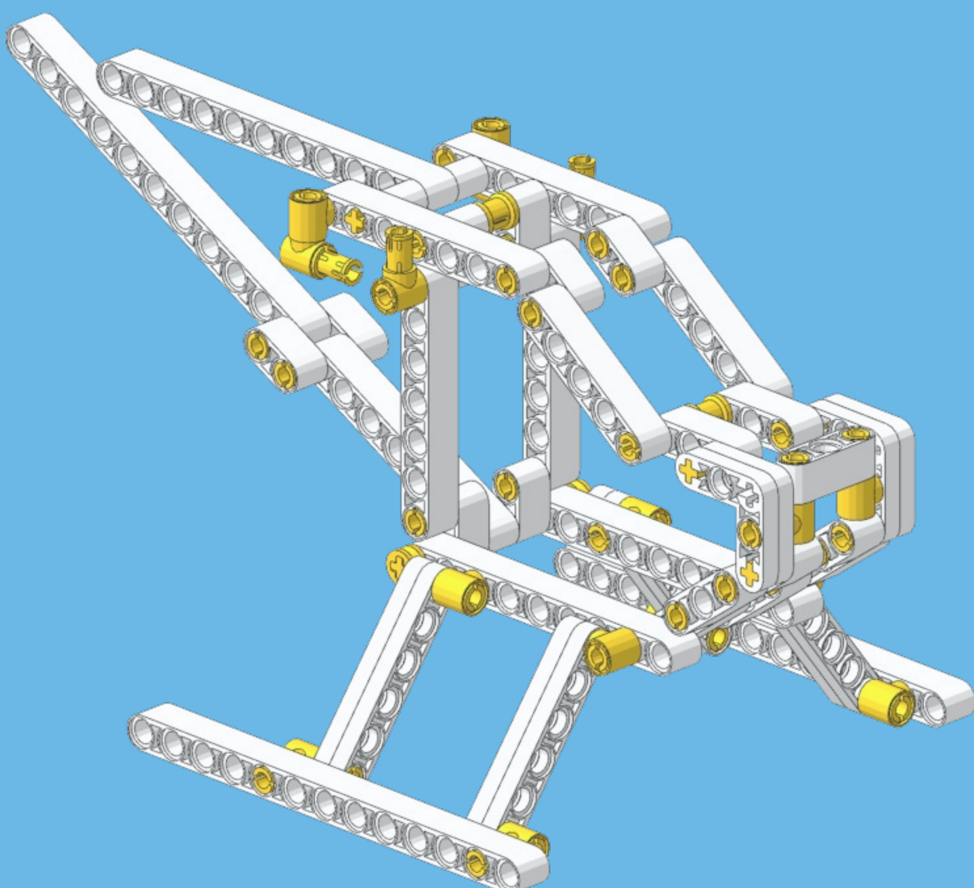
19

20



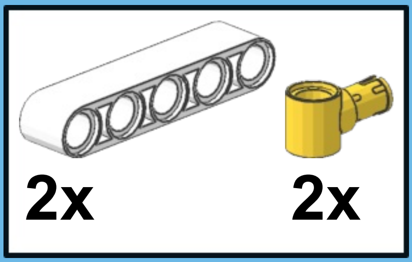
2x

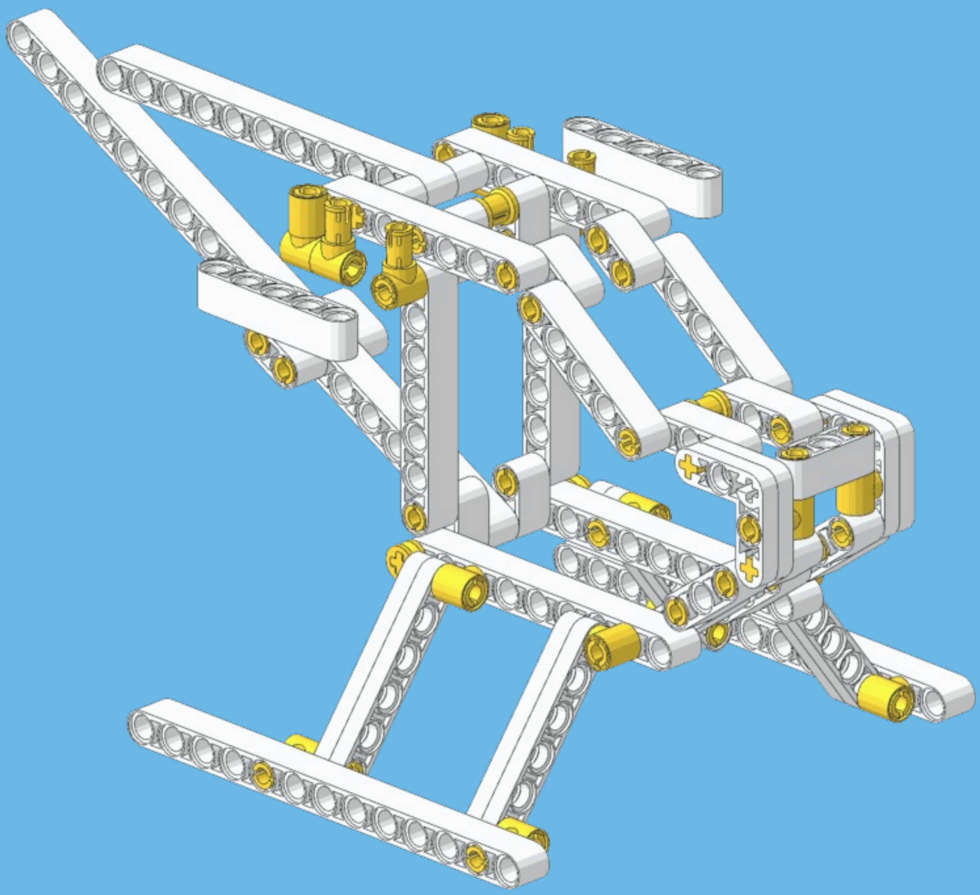
2x



20

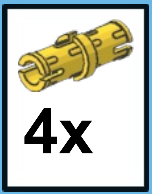
21

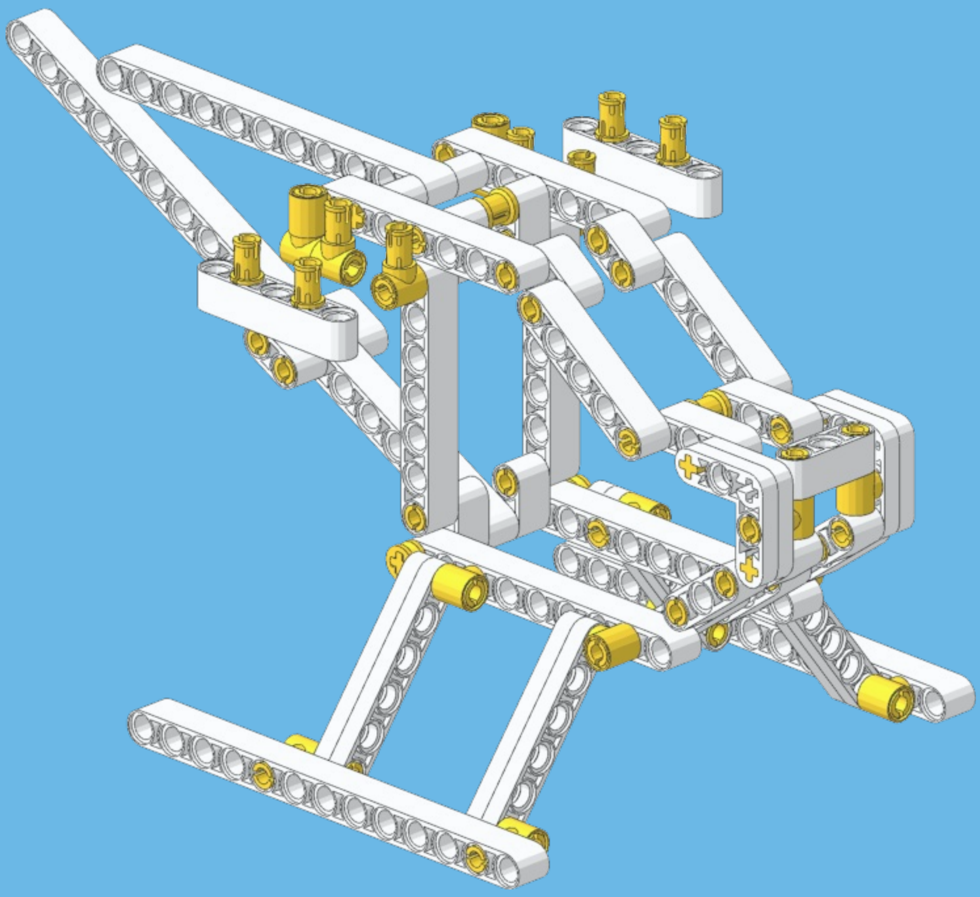




21

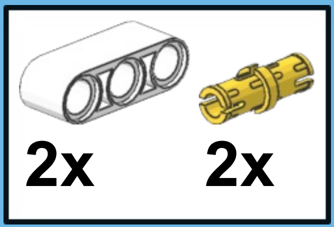
22

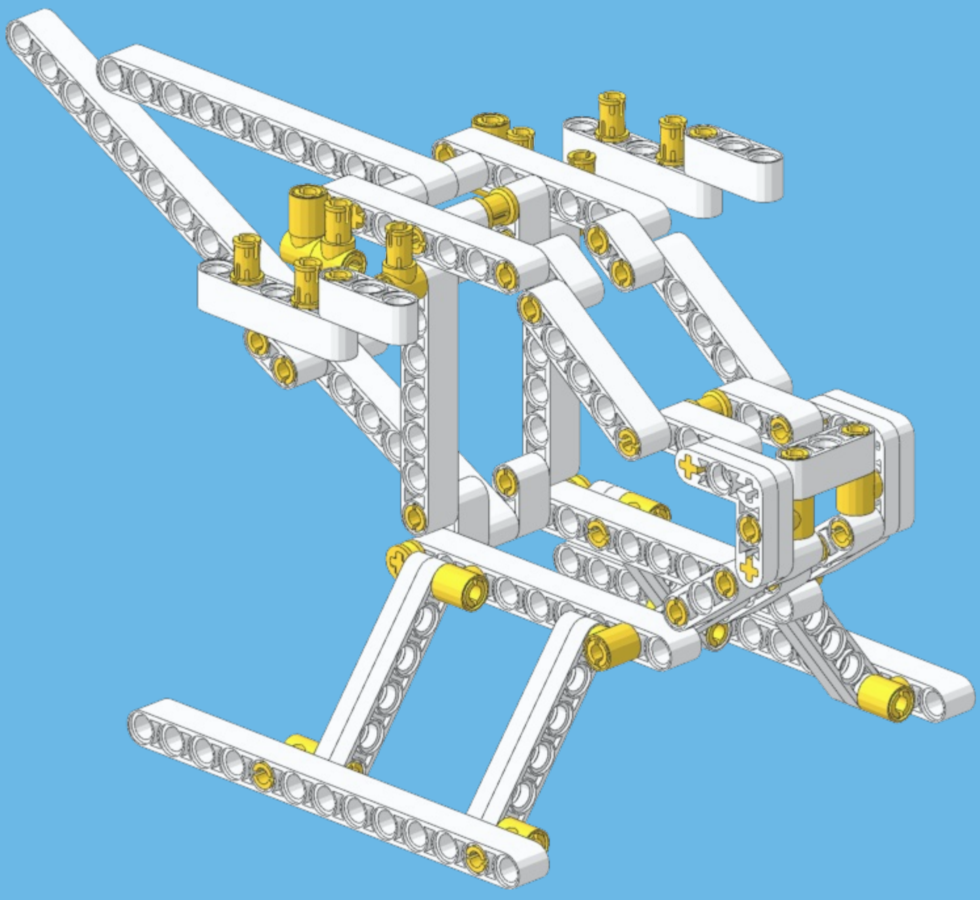




22

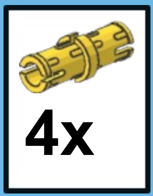
23

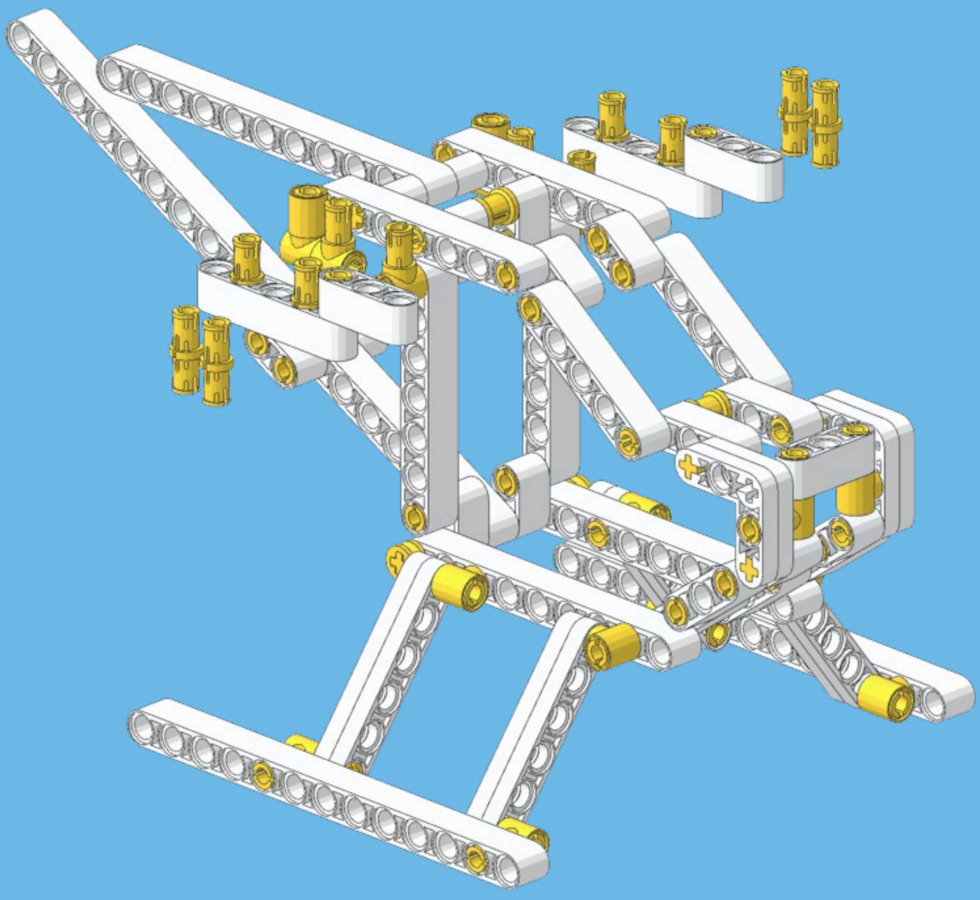




23

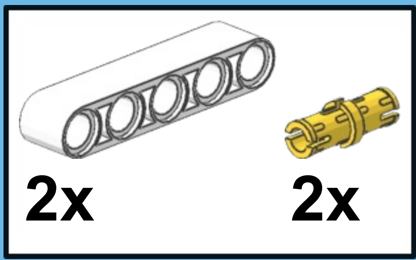
24

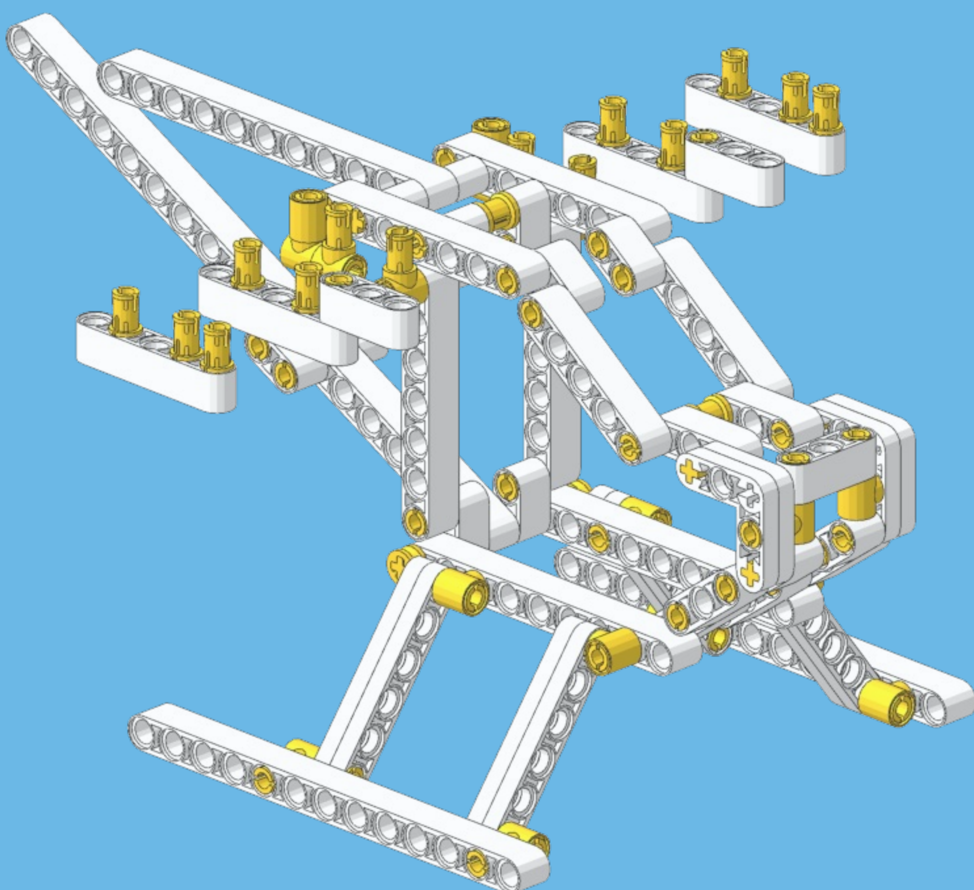




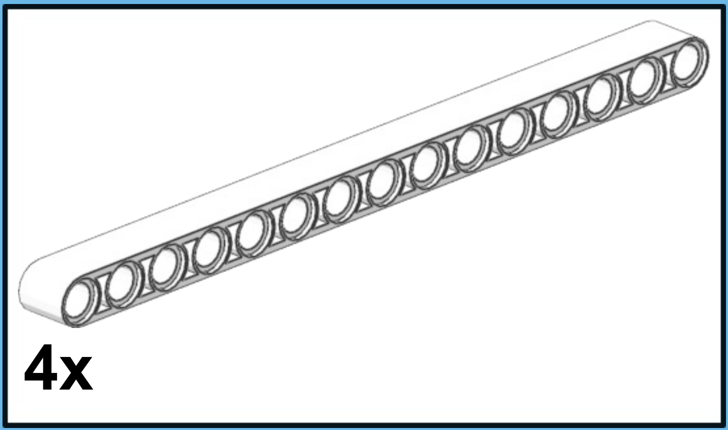
24

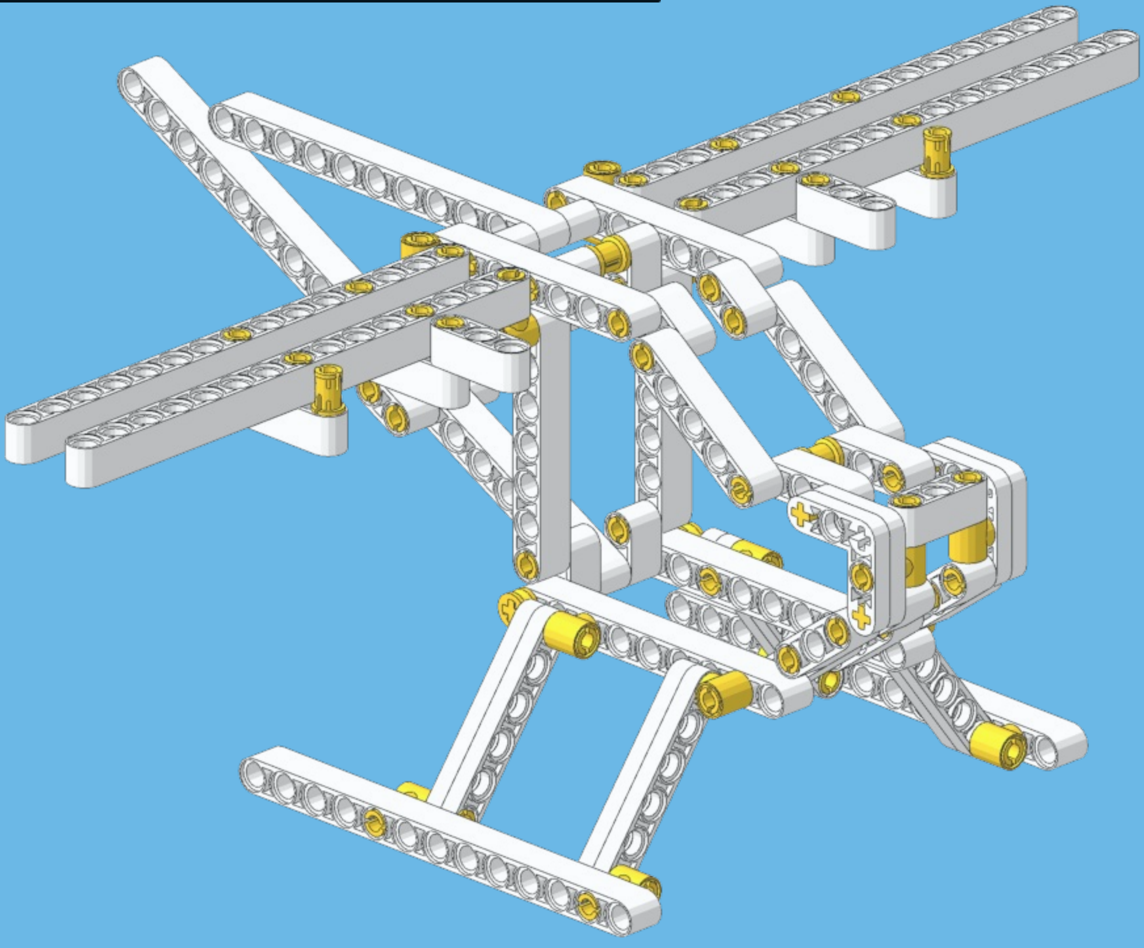
25





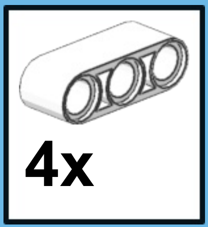
26

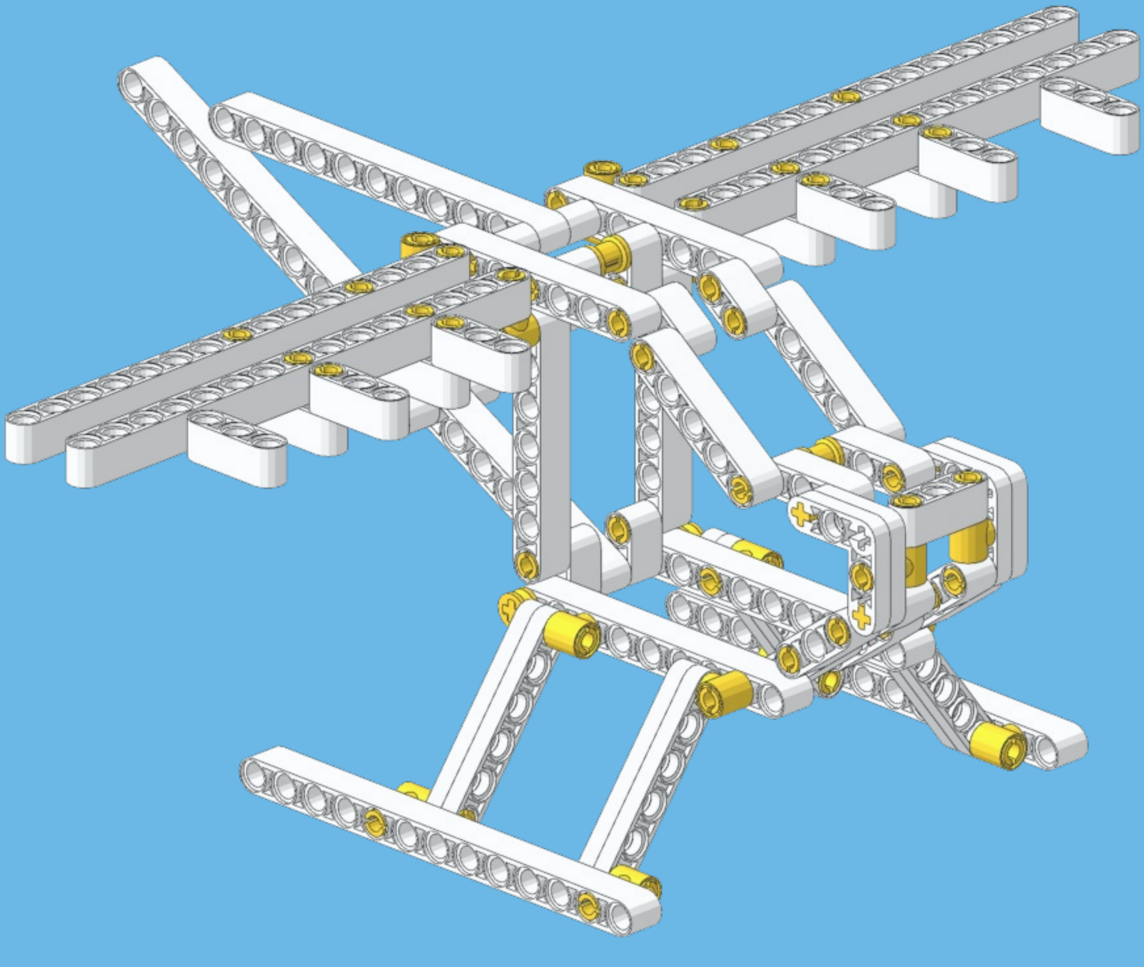




26

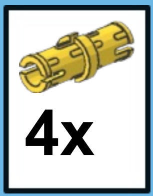
27

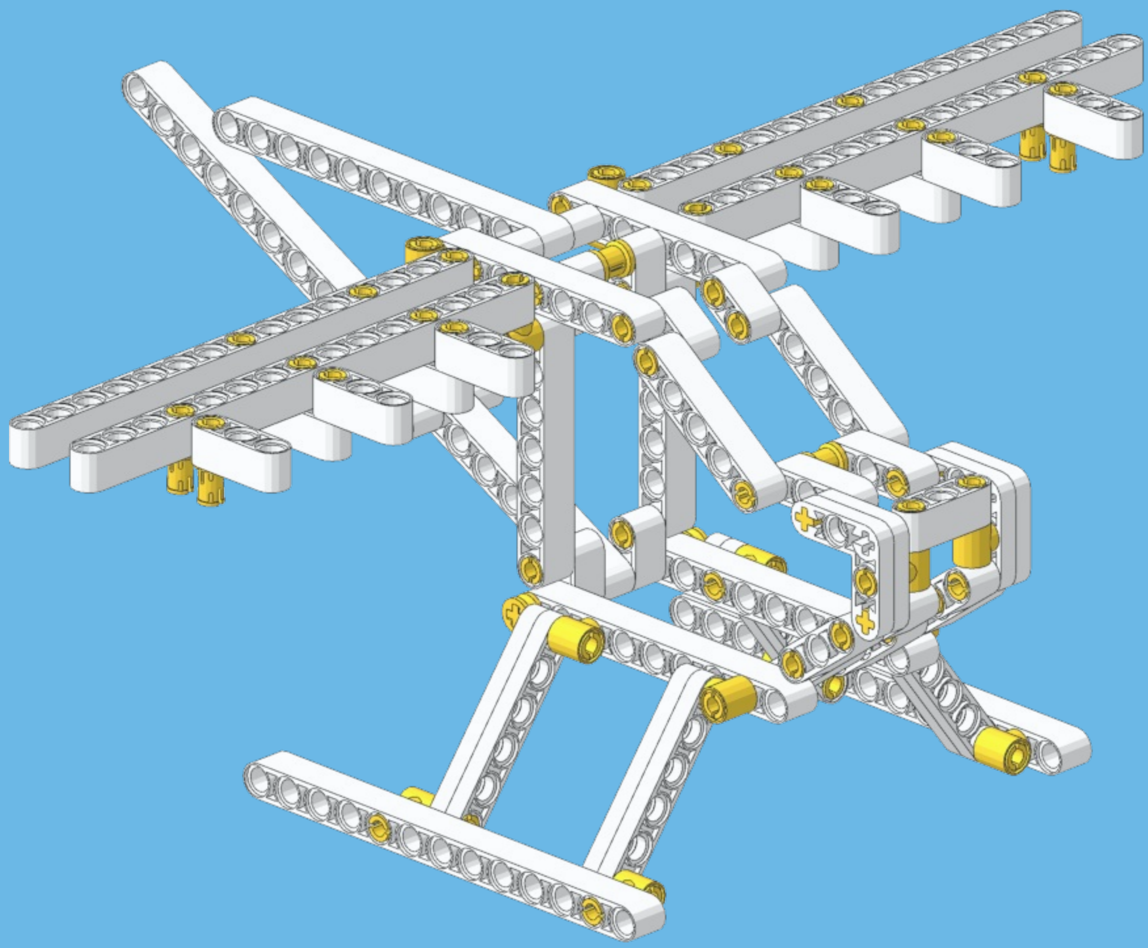




27

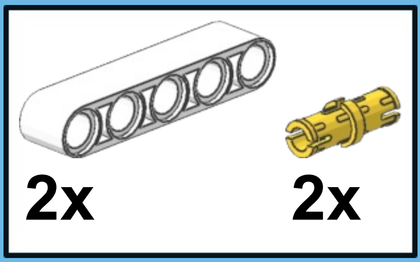
28

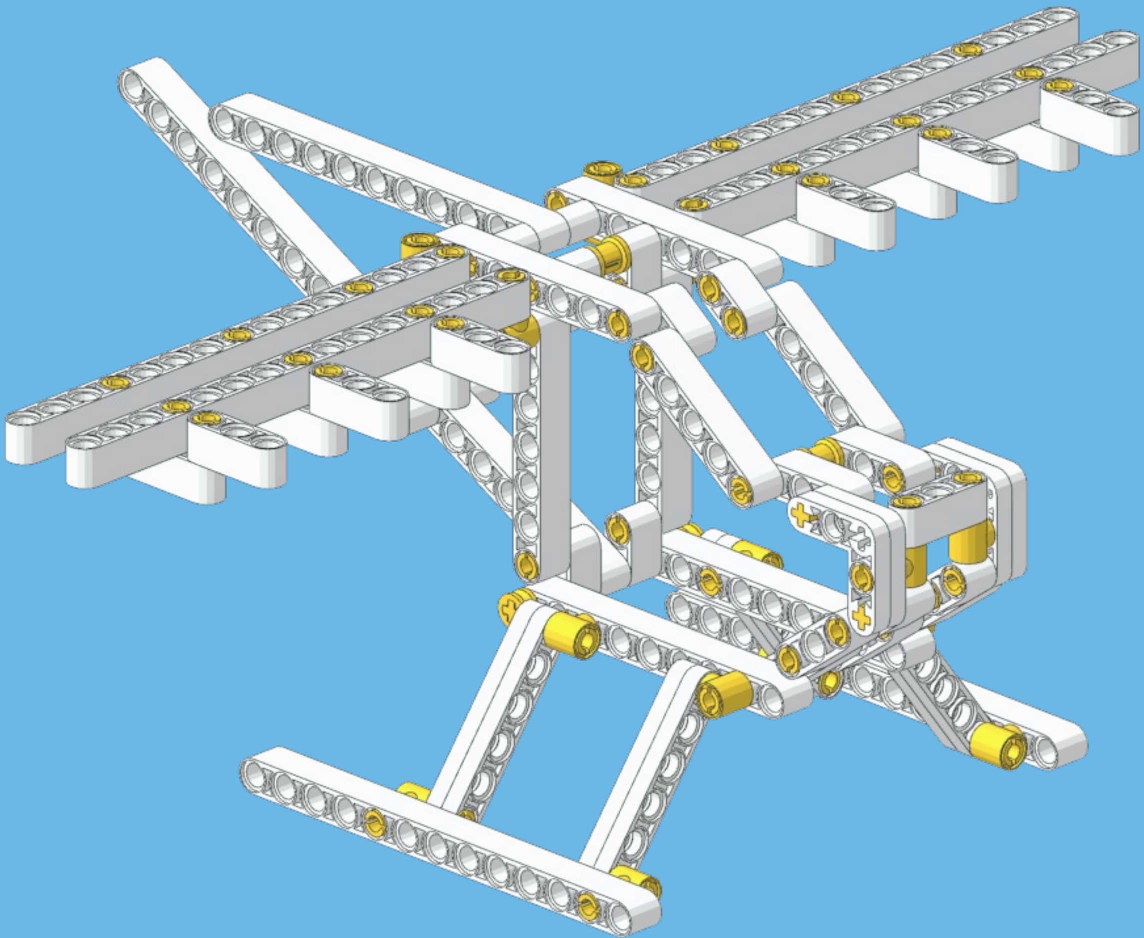




28

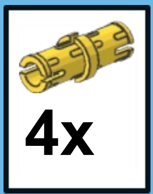
29

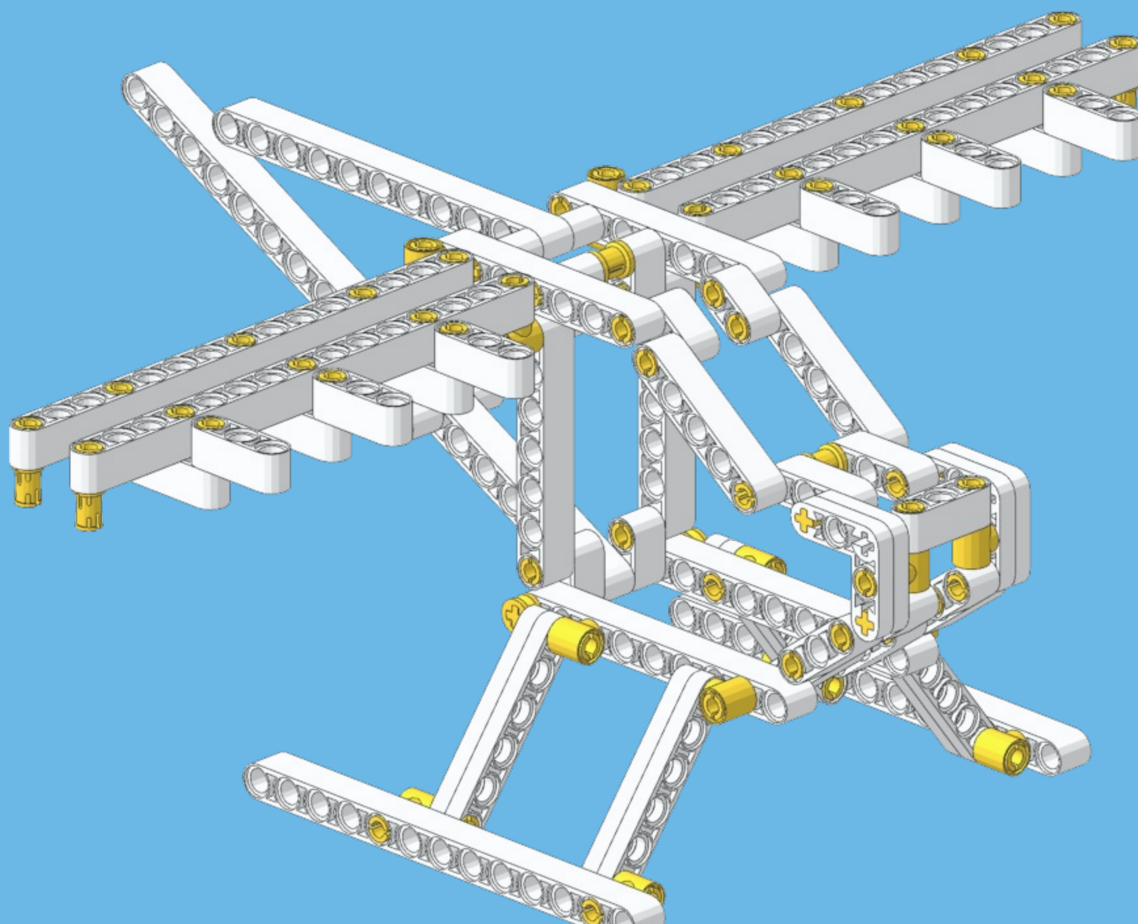




29

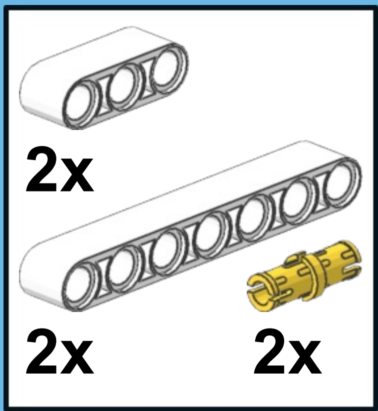
30

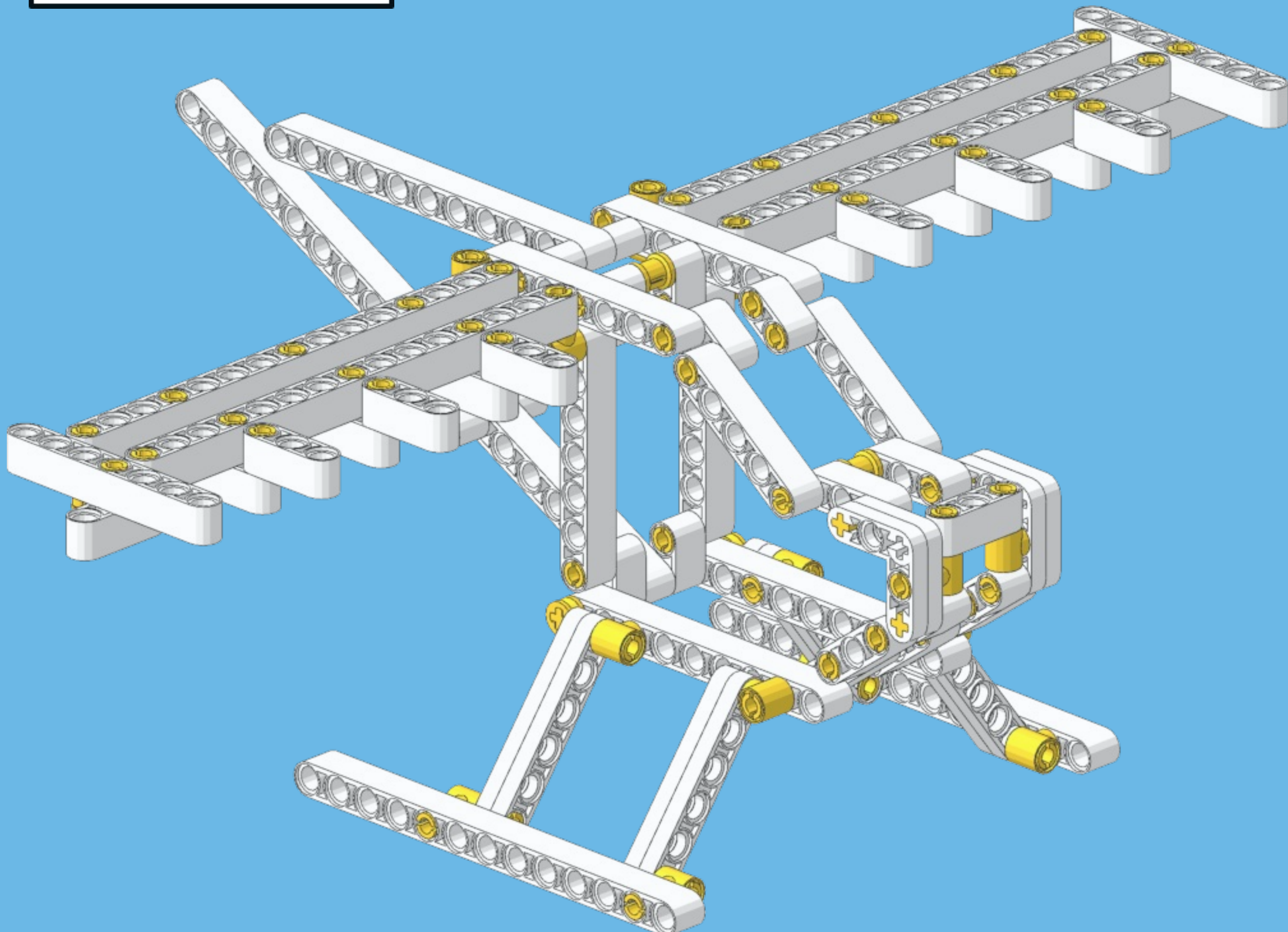




30

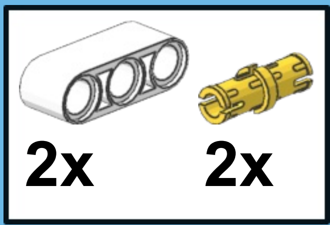
31

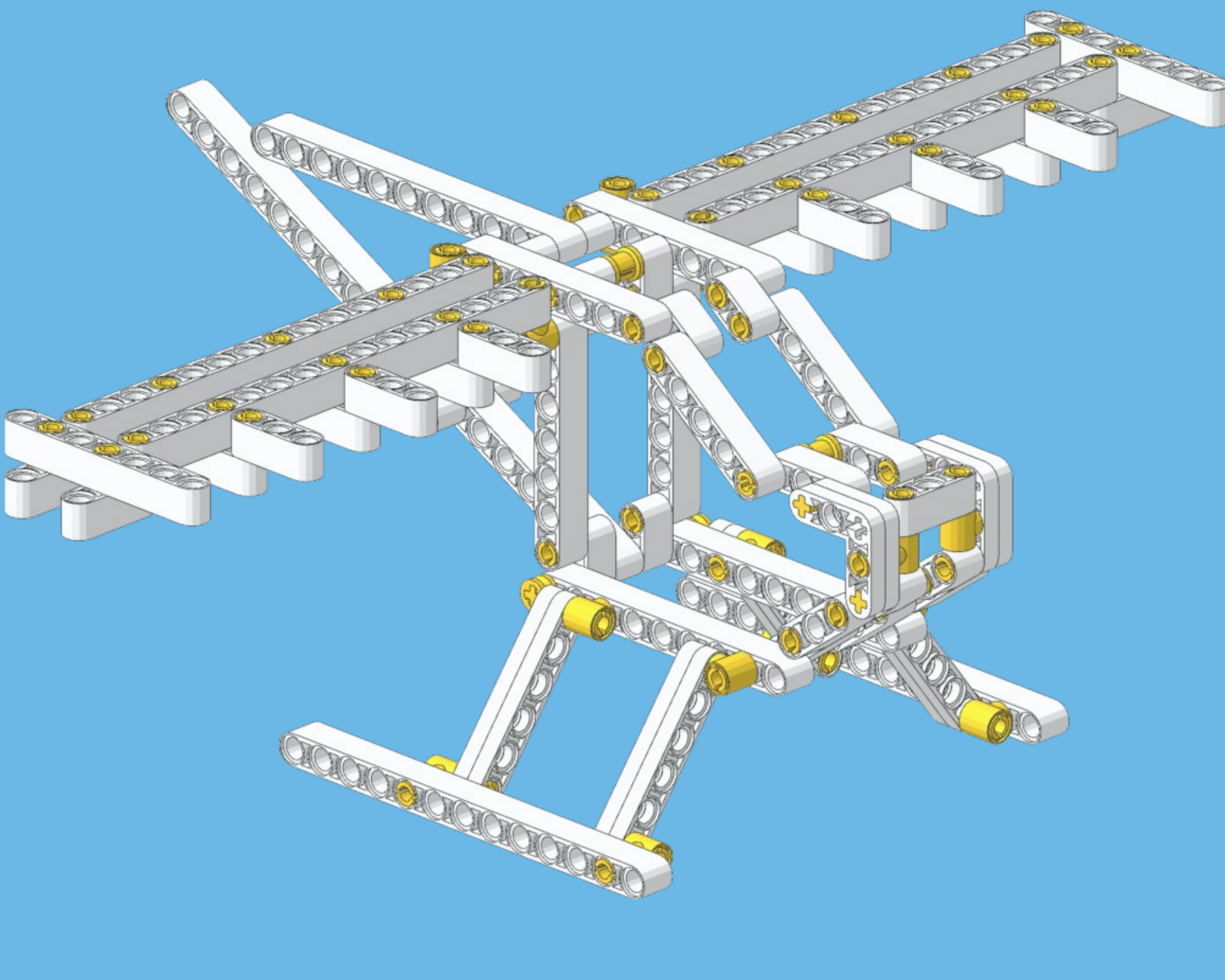




31

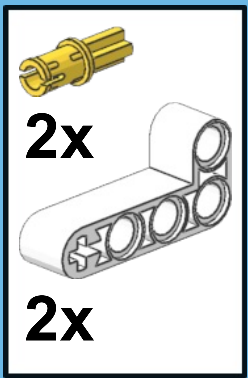
32

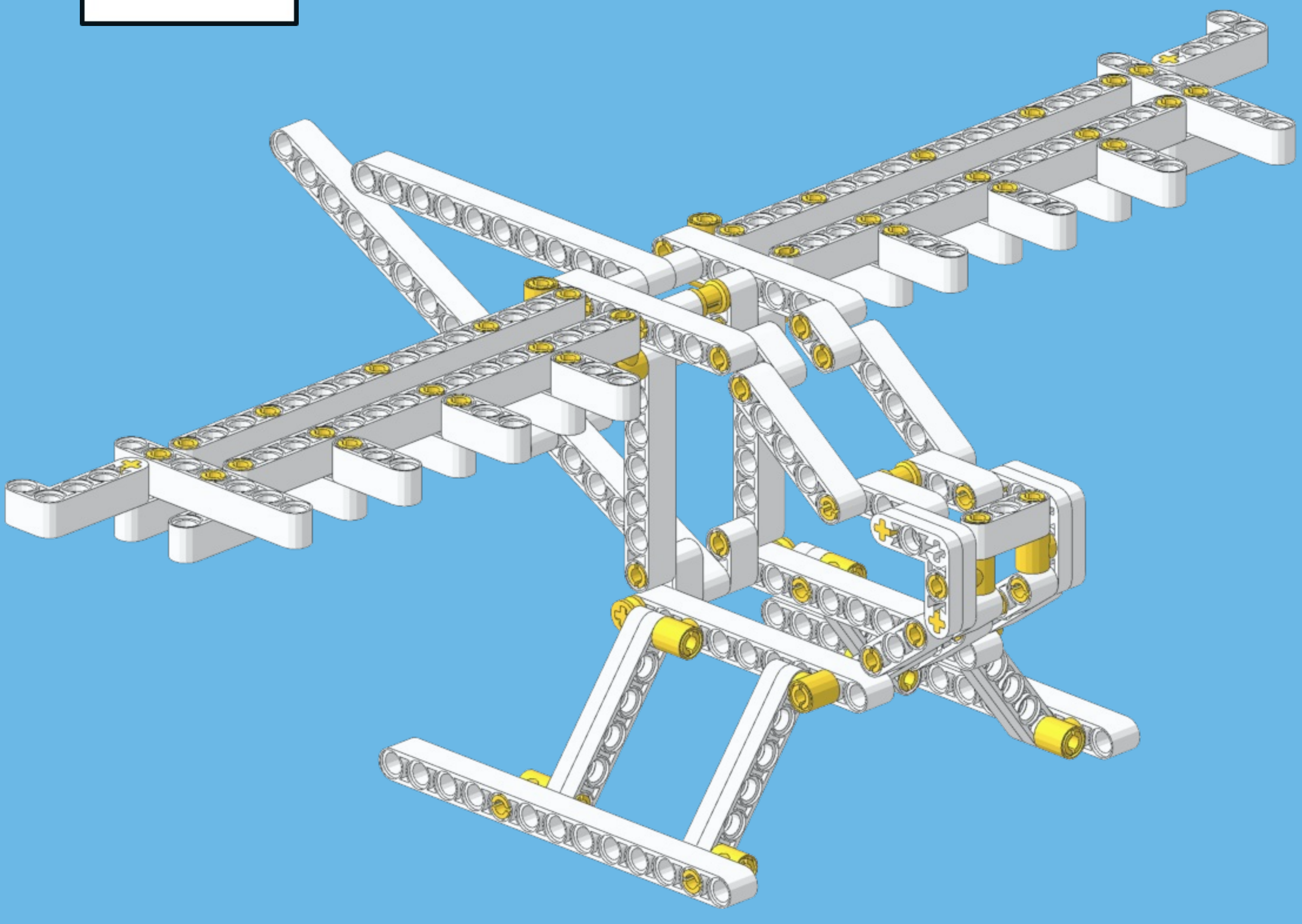




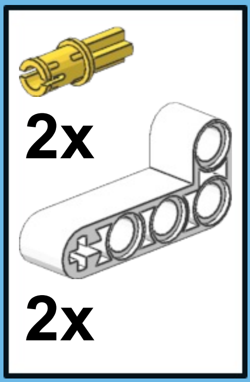
32

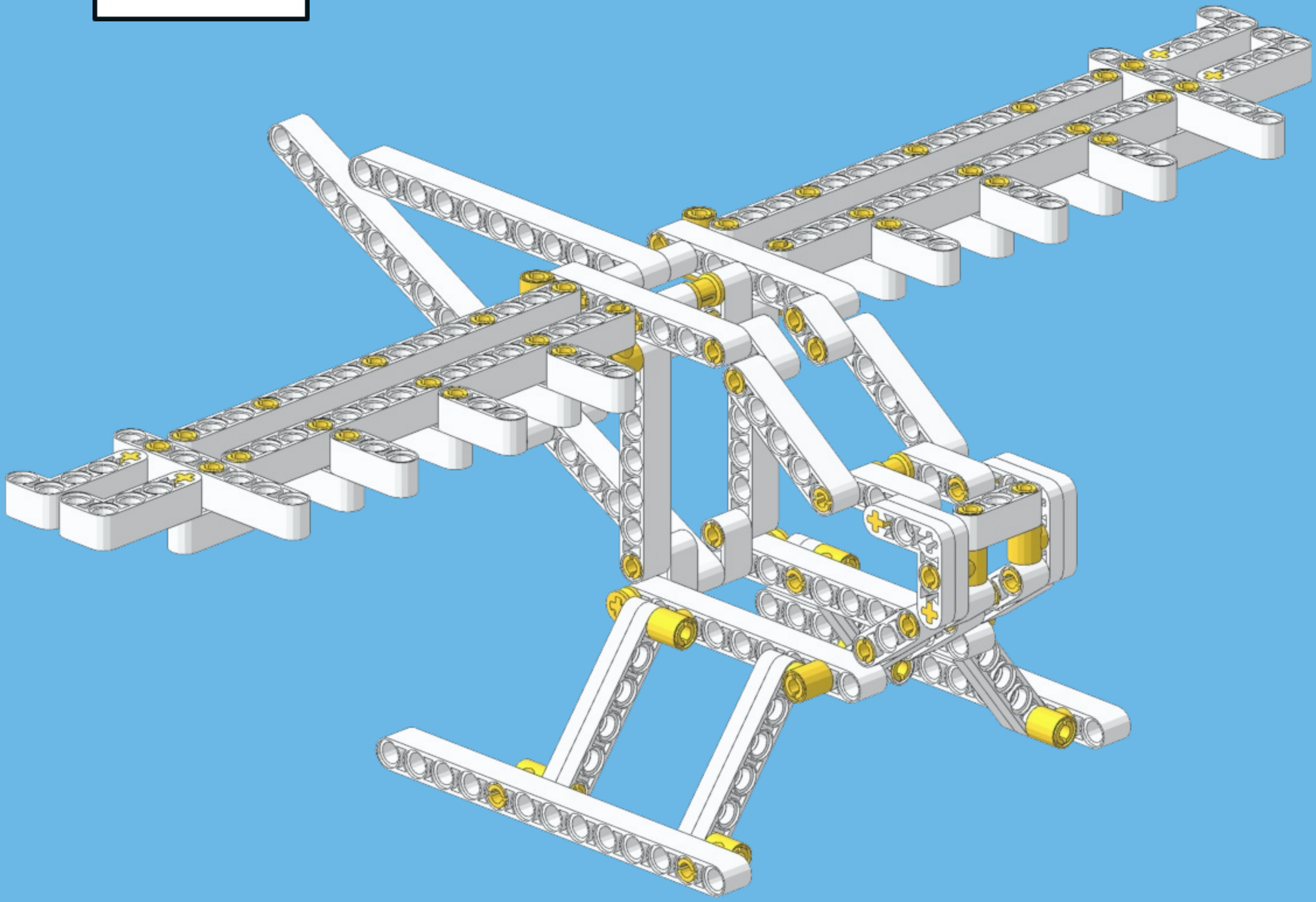
33





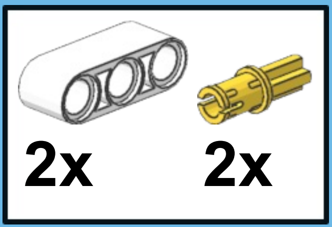
34

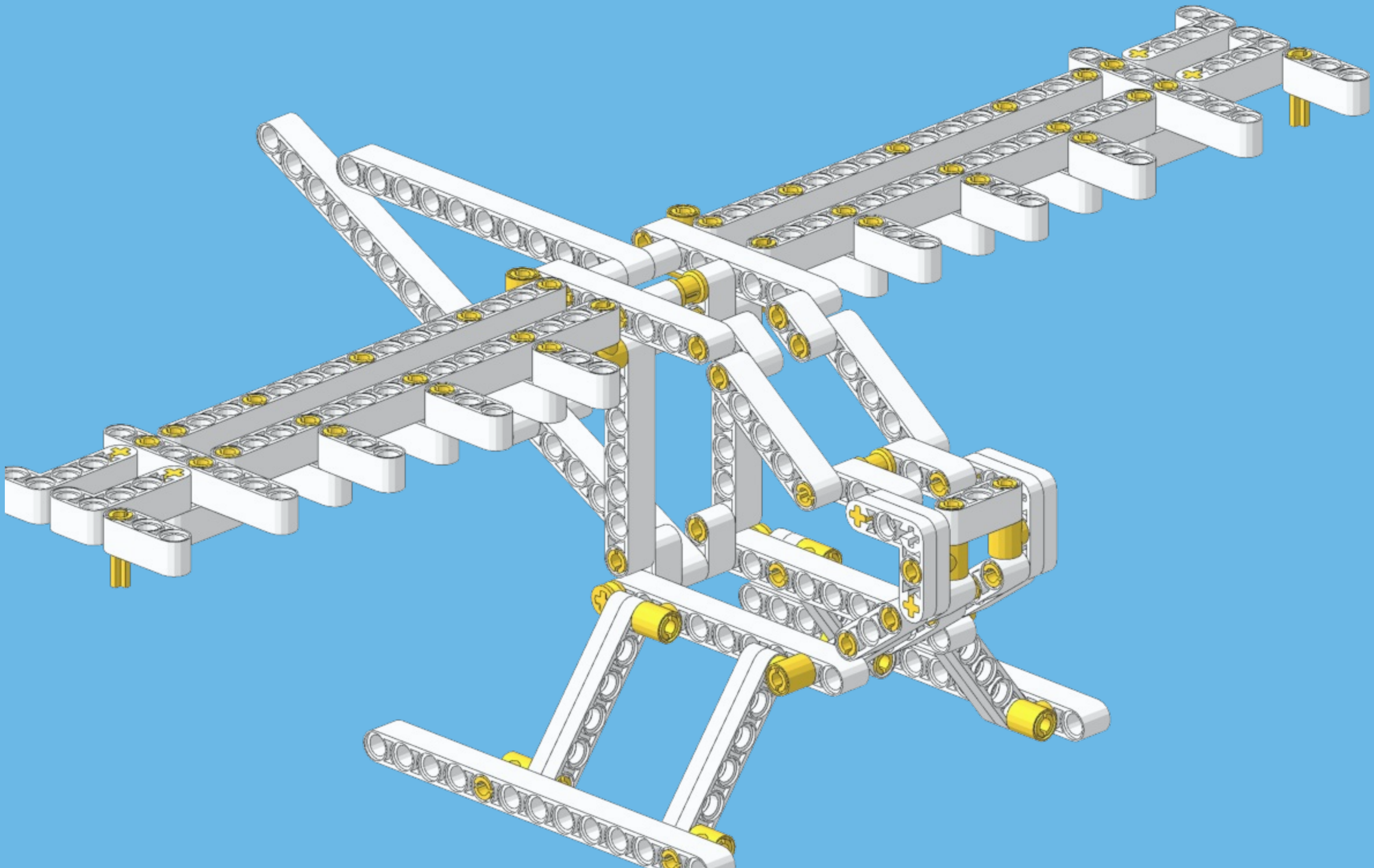




34

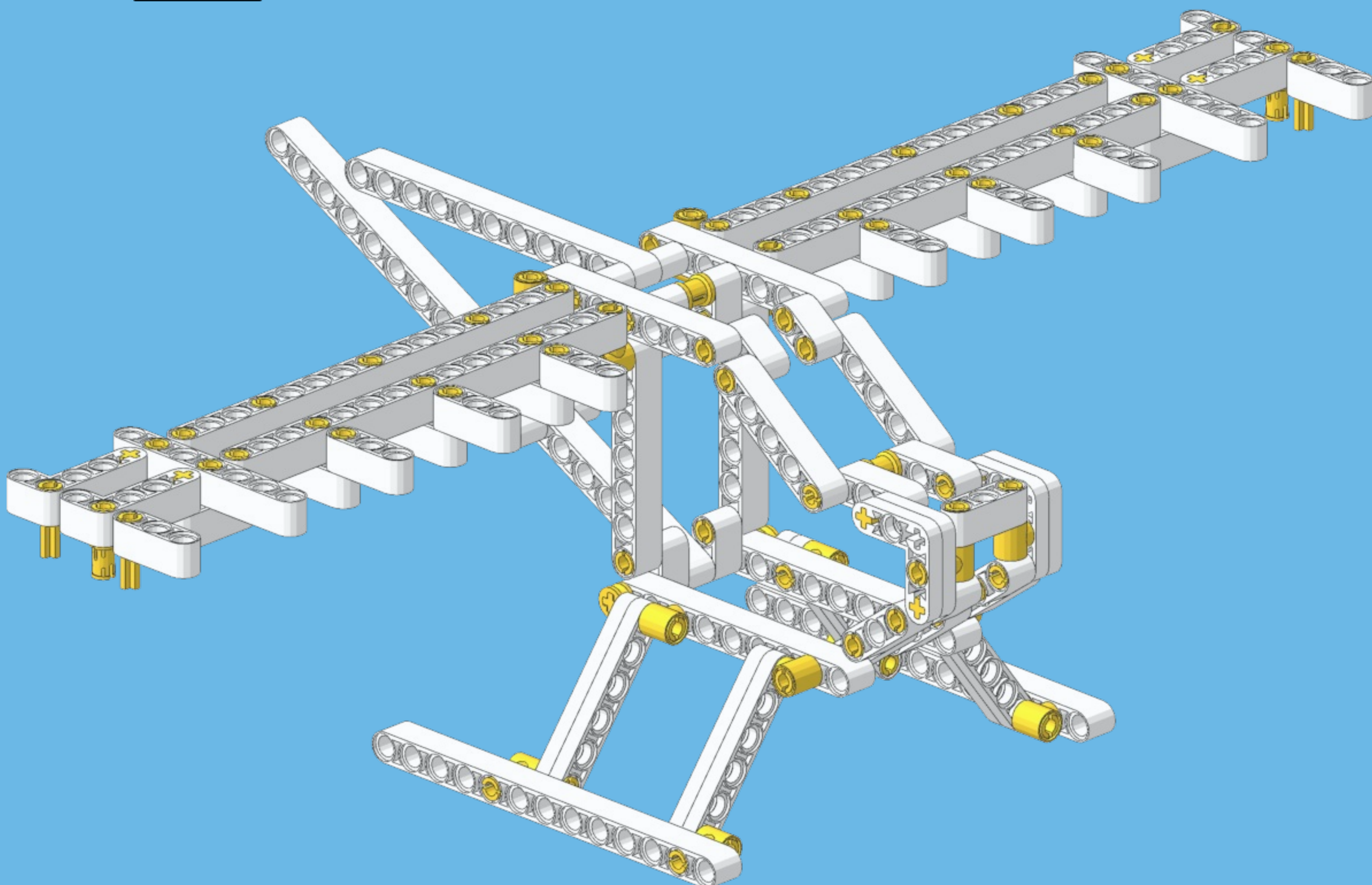
35





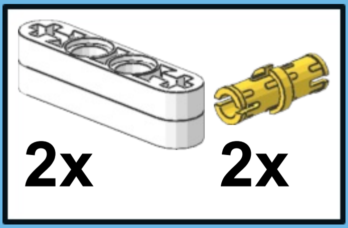
35

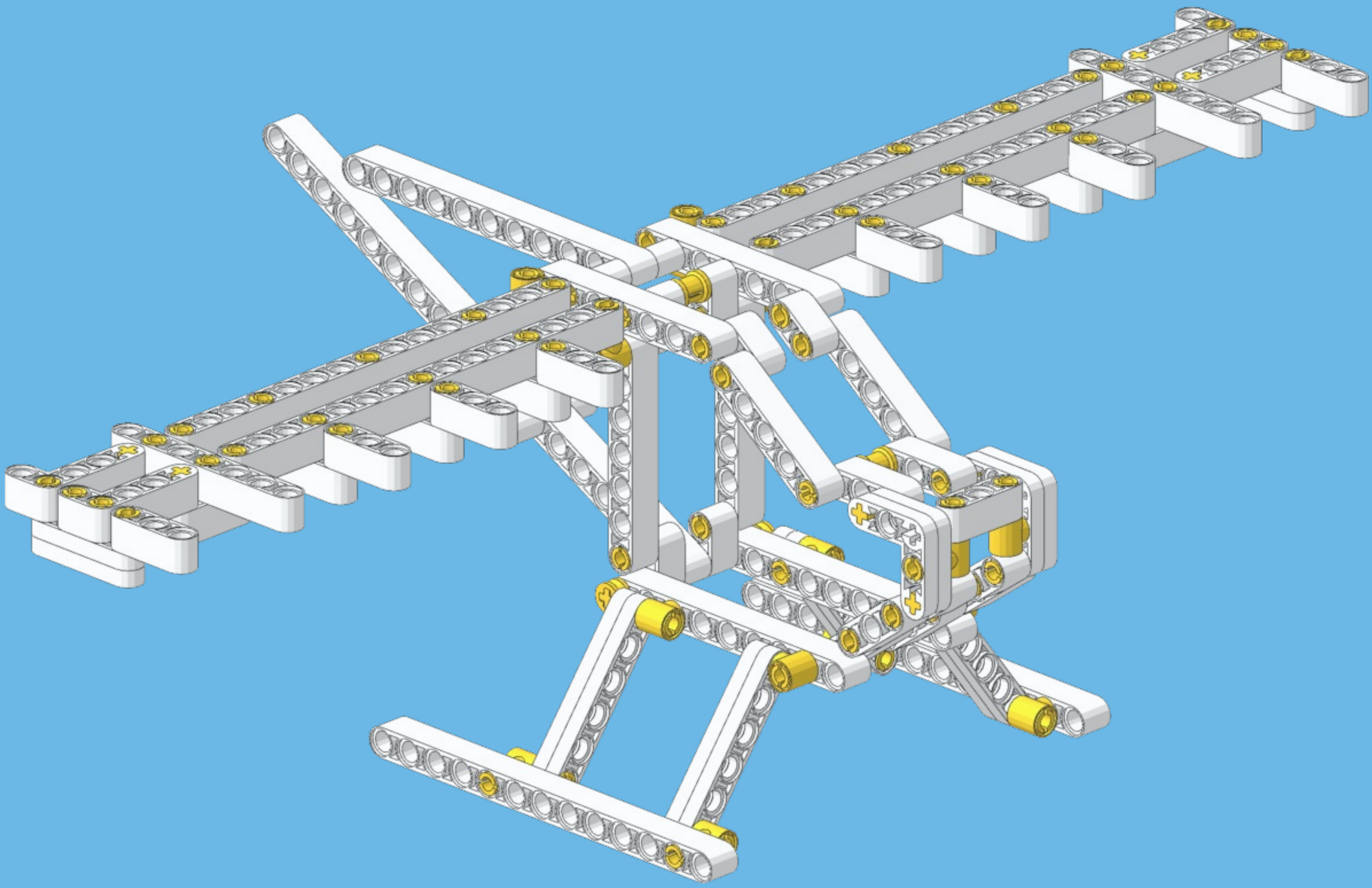
36



36

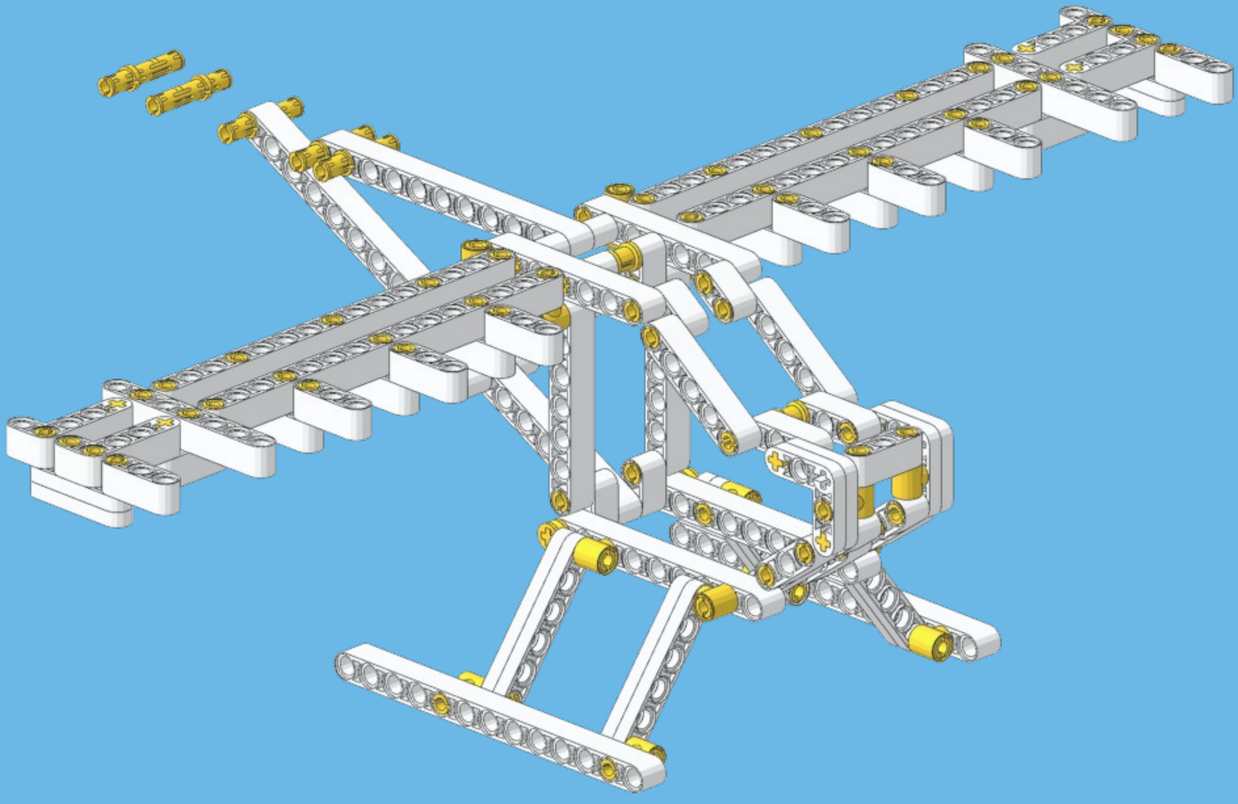
37





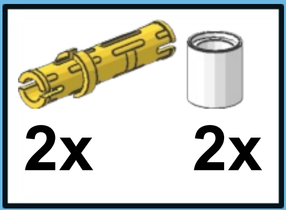
37

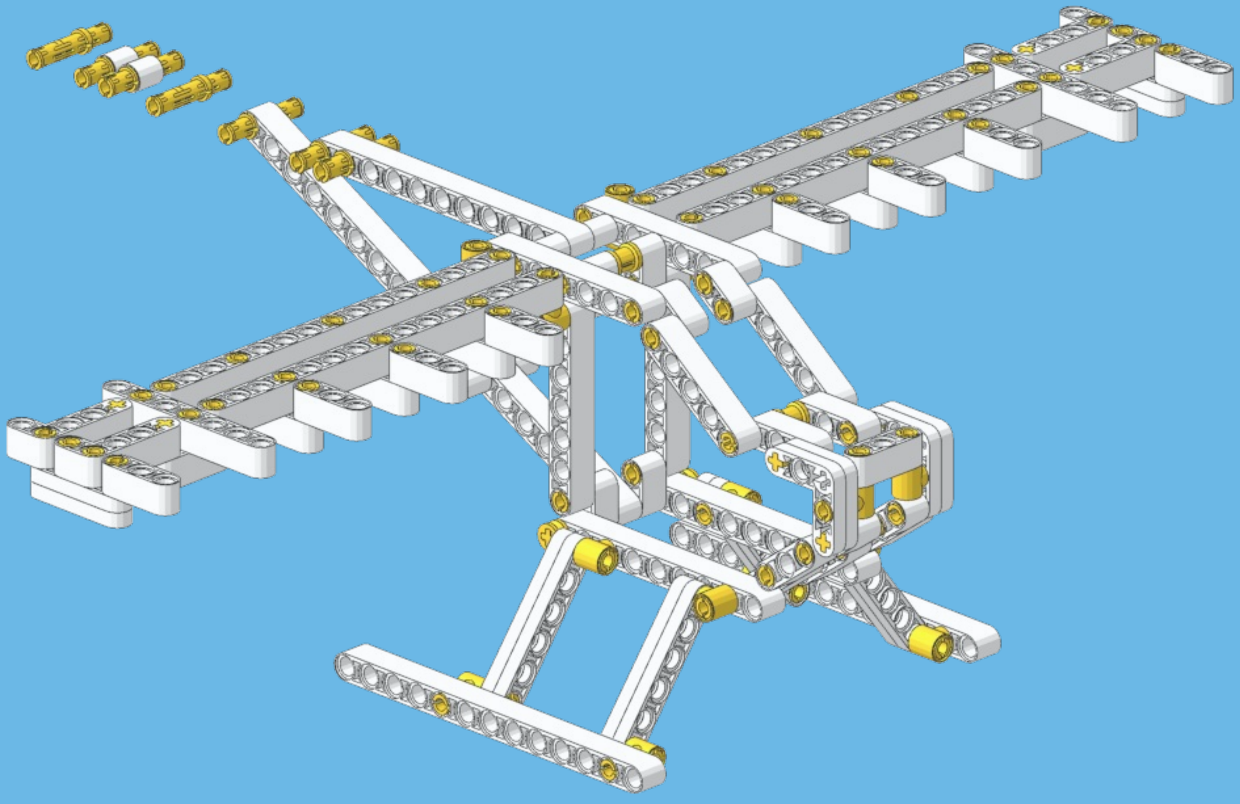
38



38

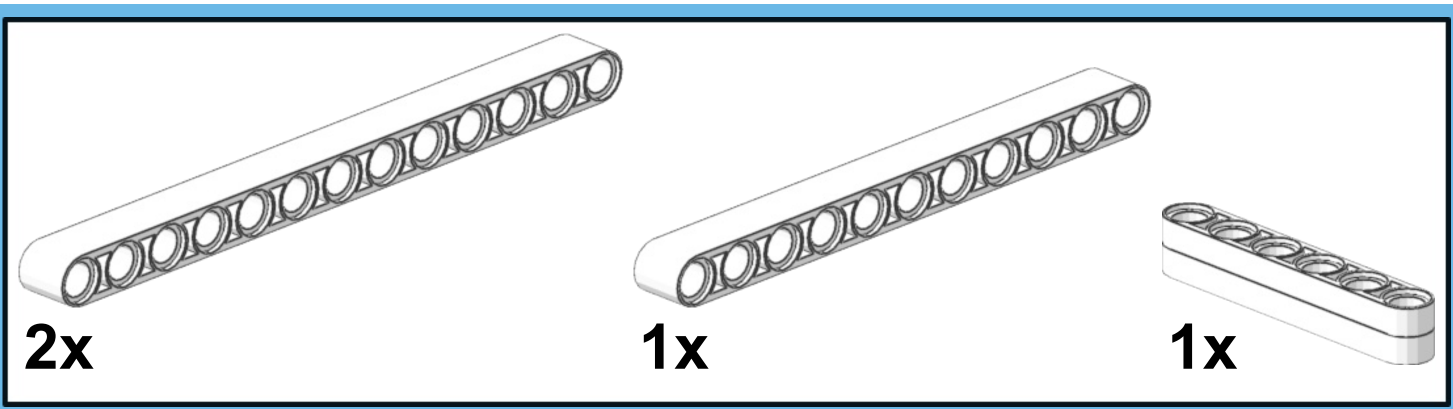
39

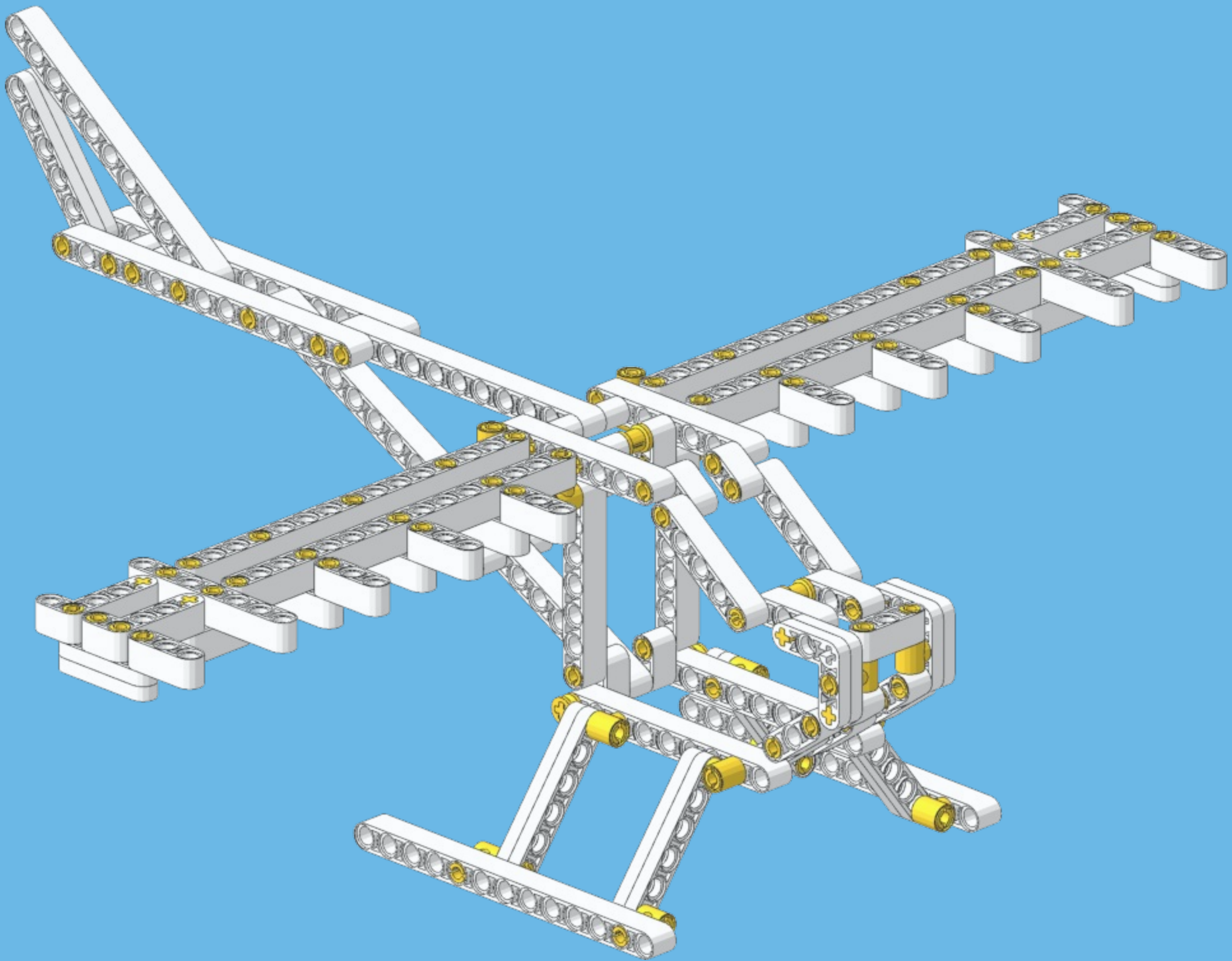




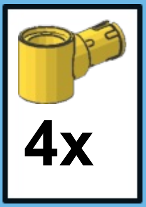
39

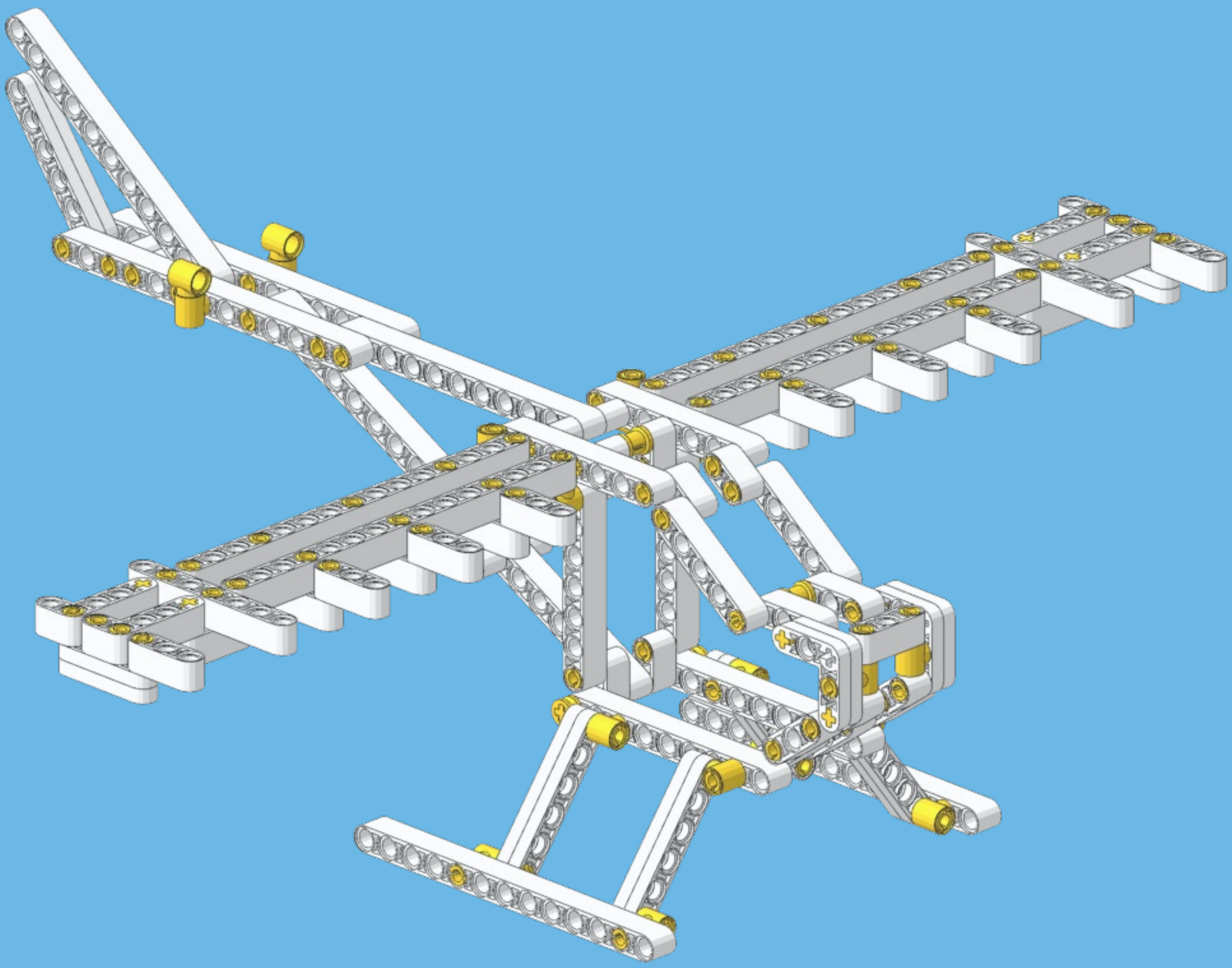
40





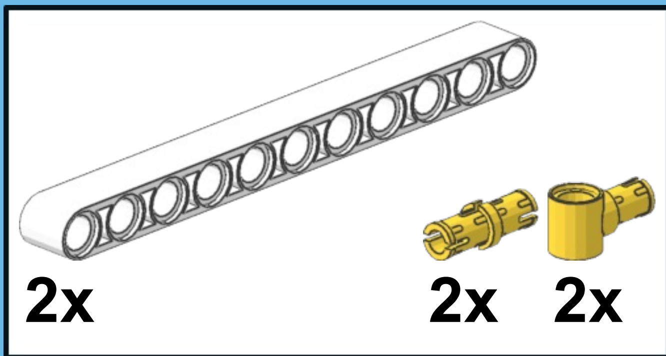
41

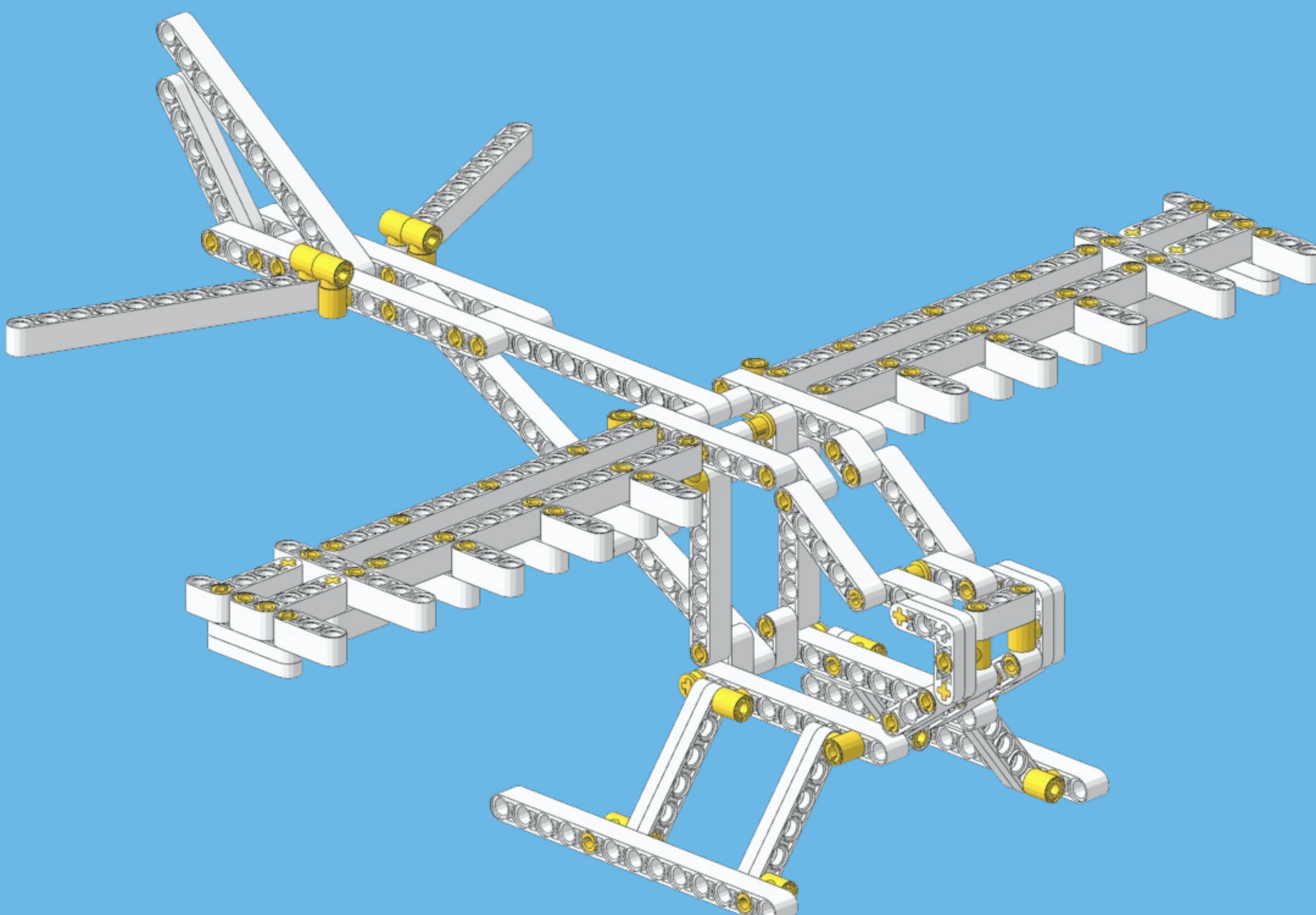




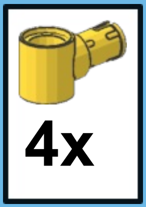
41

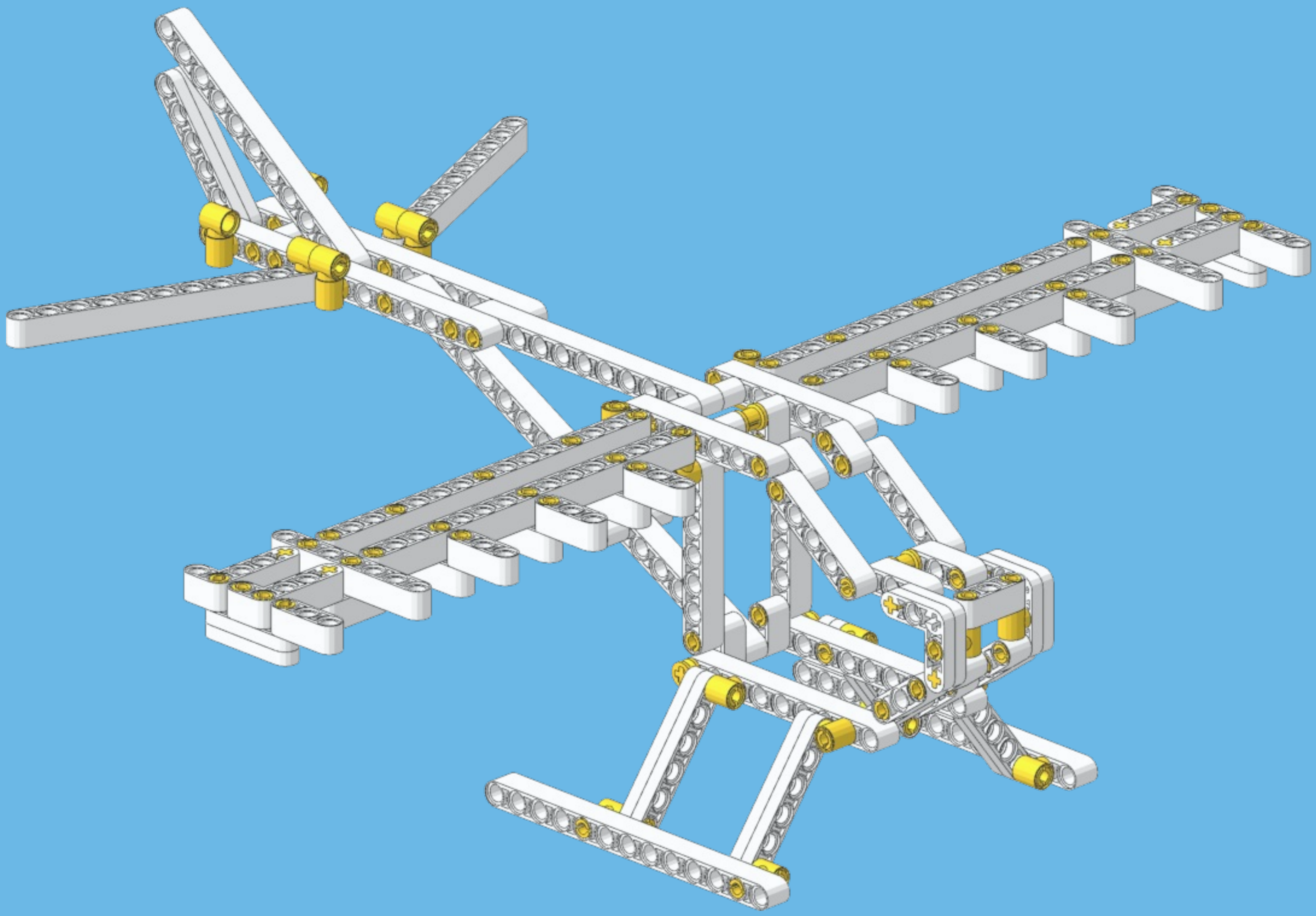
42





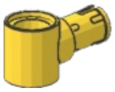
43





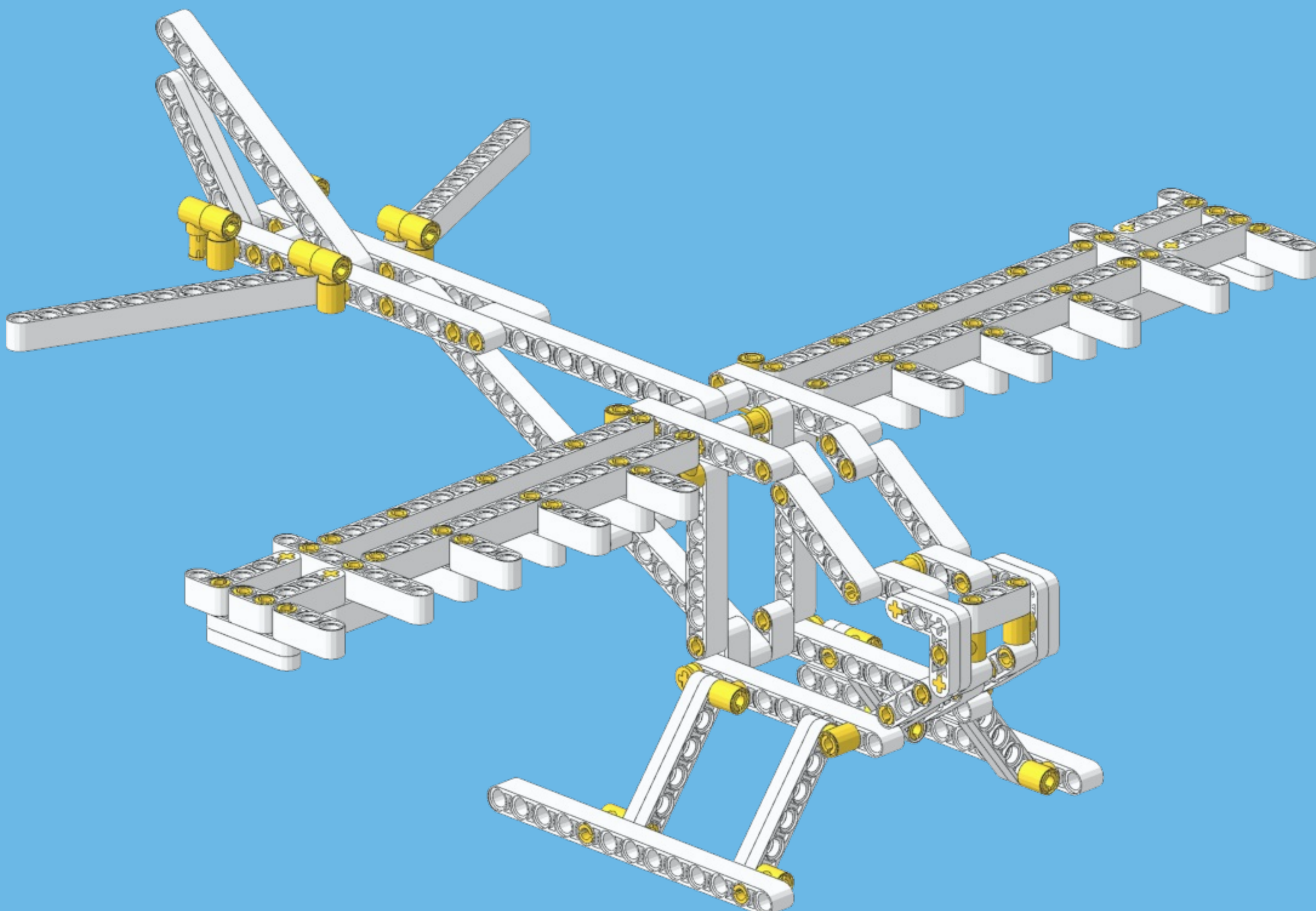
43

44

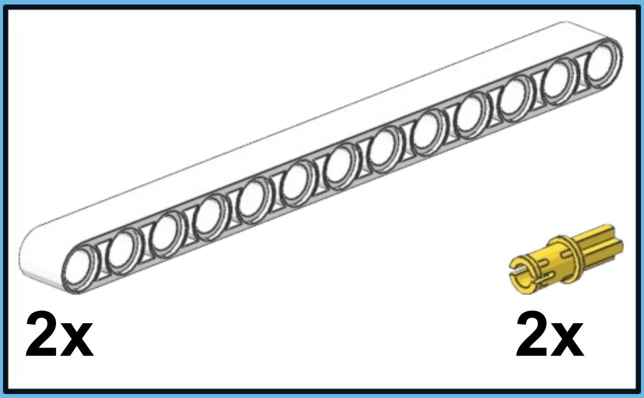


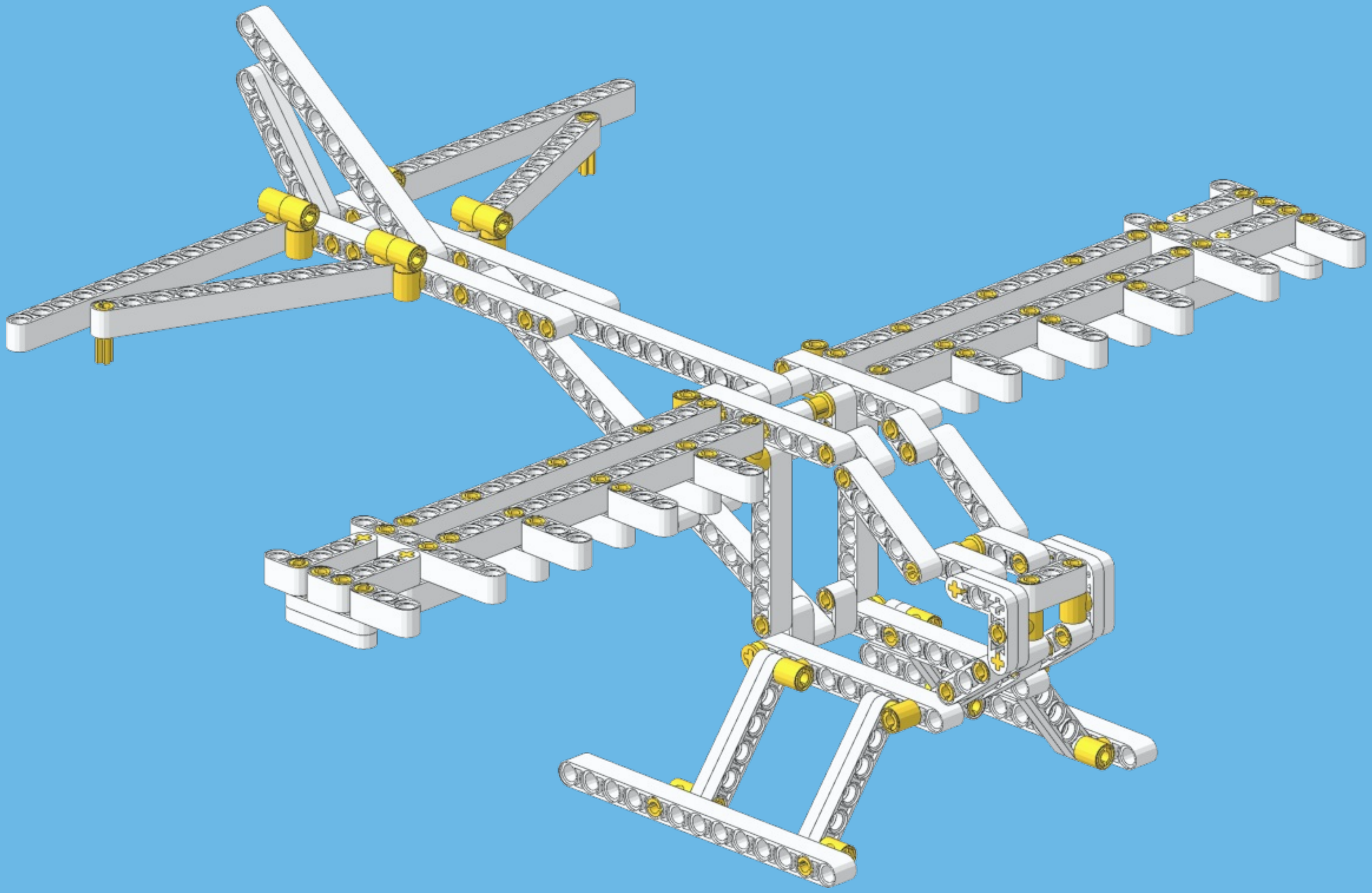
2x

2x



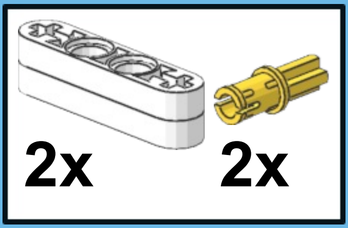
45

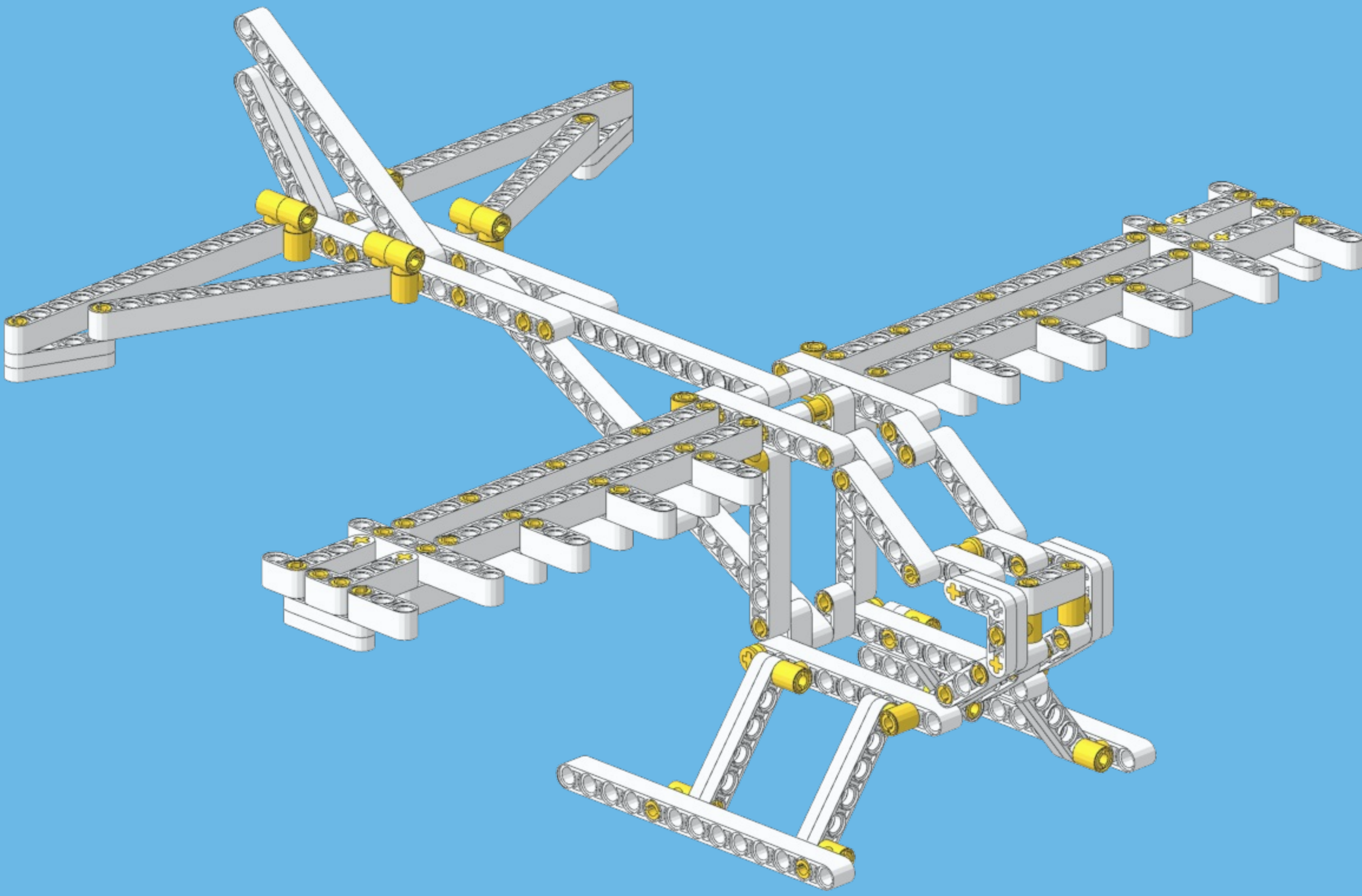




45

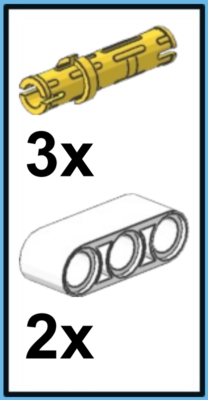
46

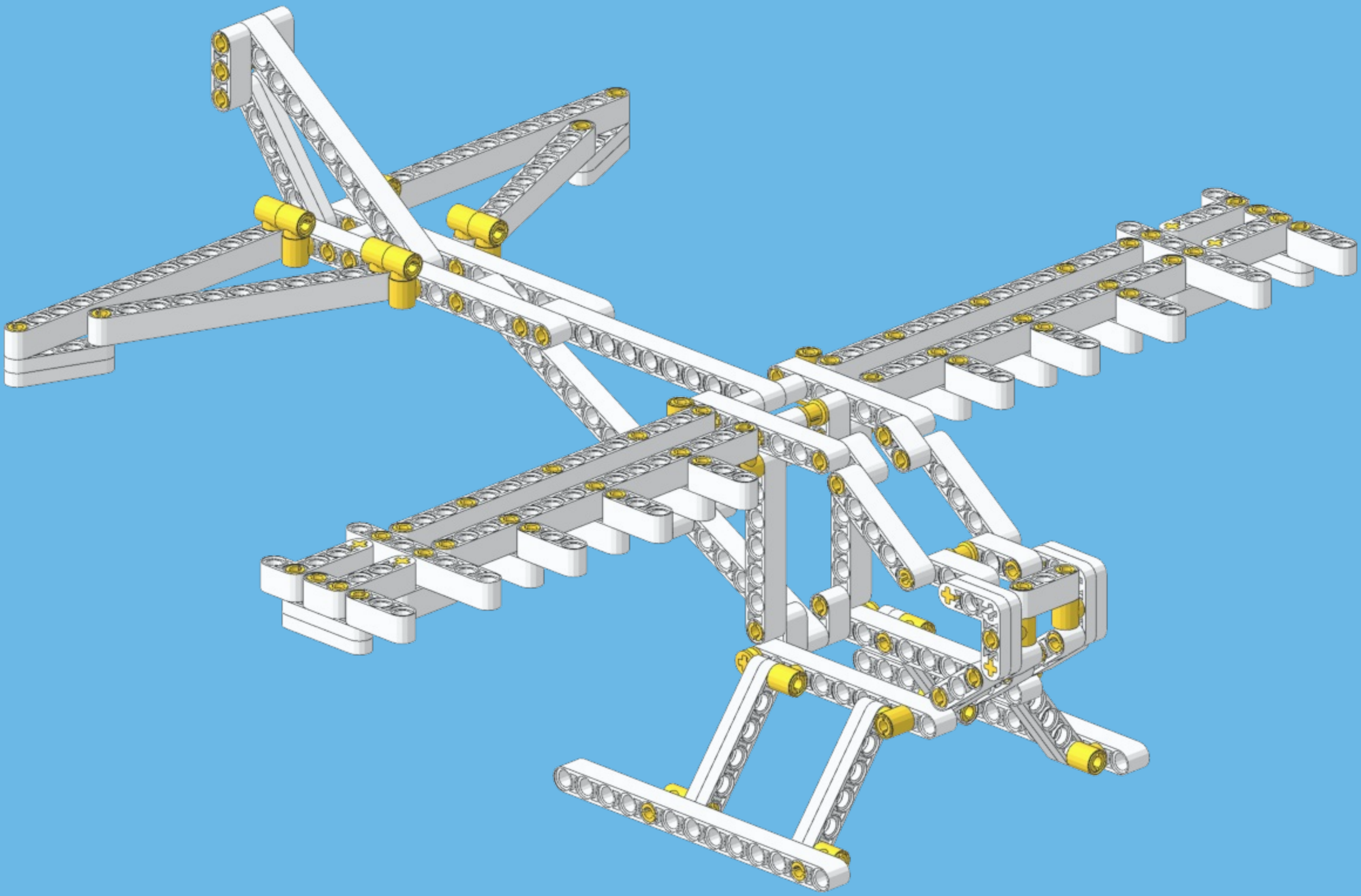




46

47



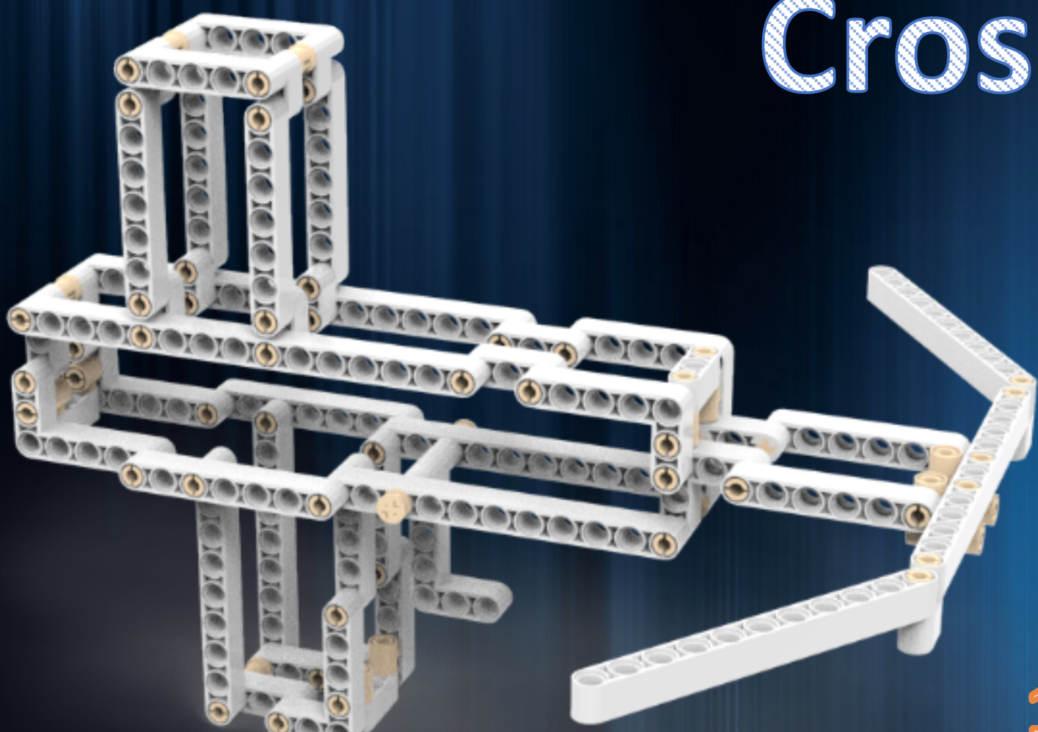


47

Computational LEGO Technic Design

Assembly Instructions:

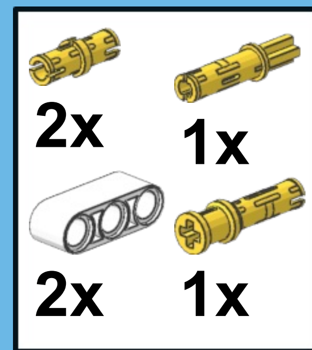
Crossbow

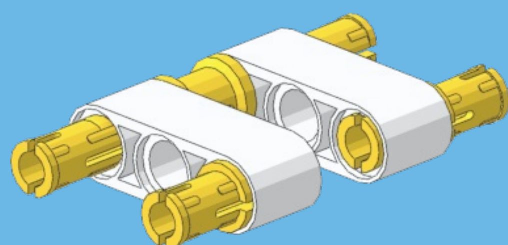


119 bricks

23cm x 25cm x 17cm

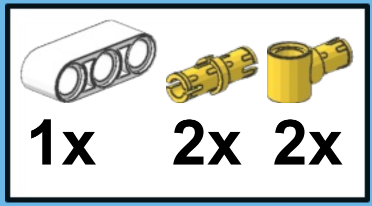
1

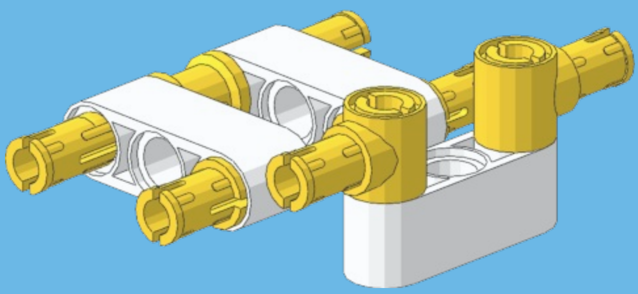




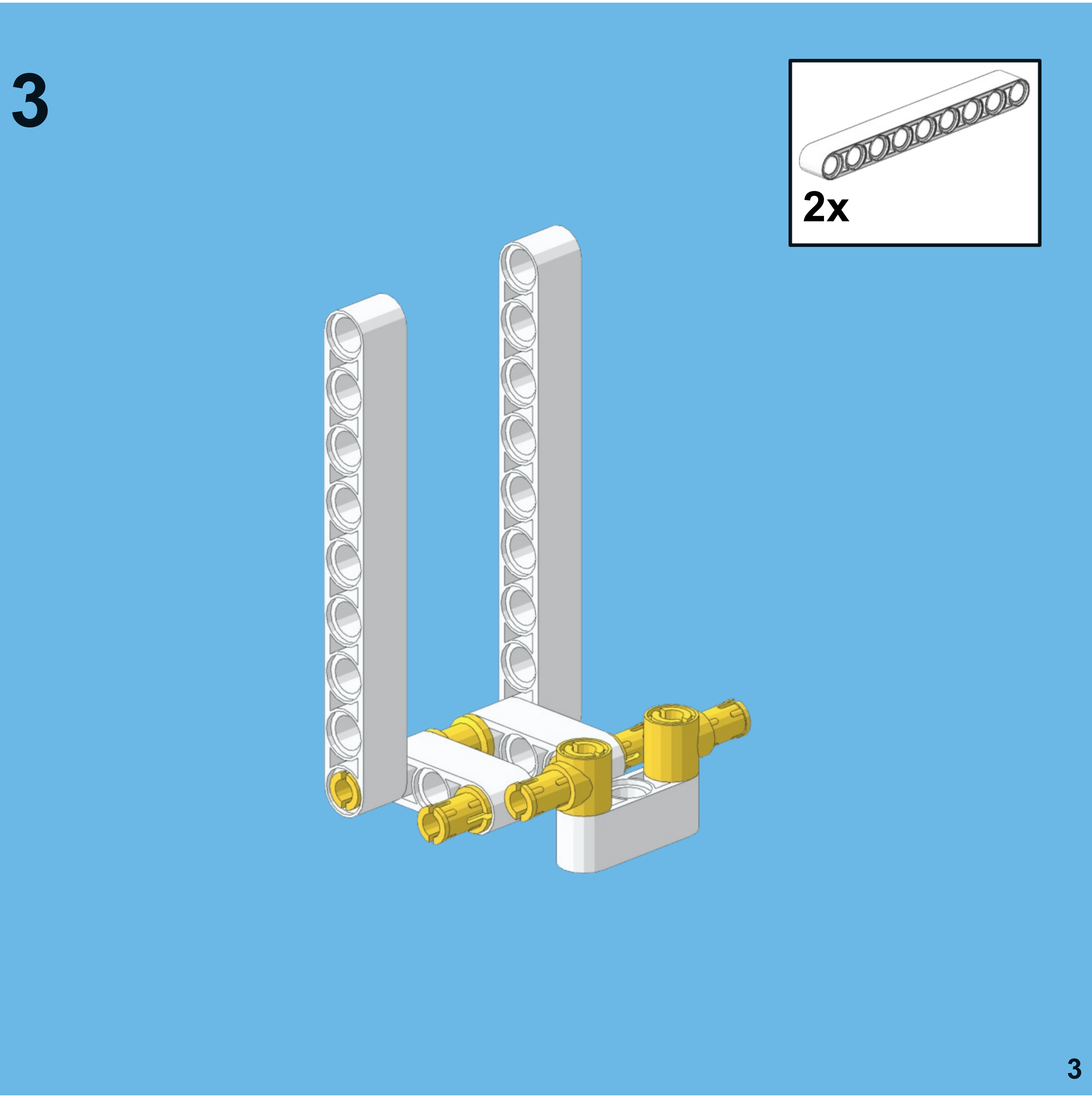
1

2

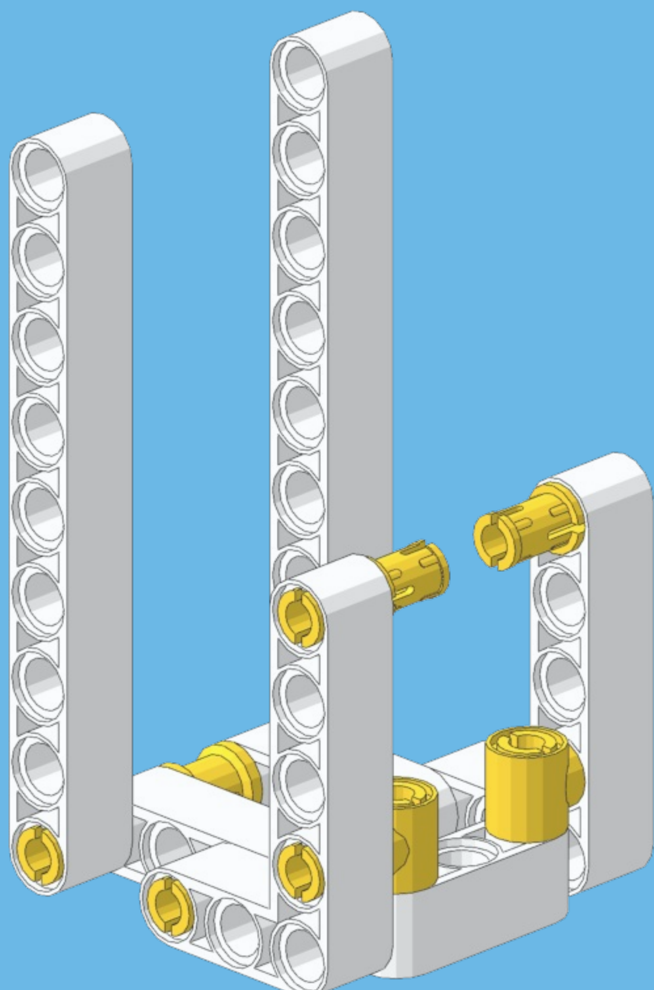


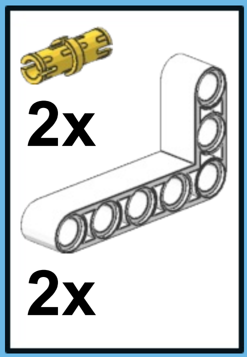


2

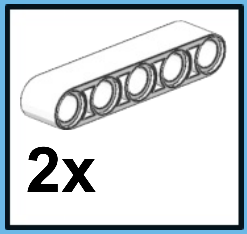

3
2x
3

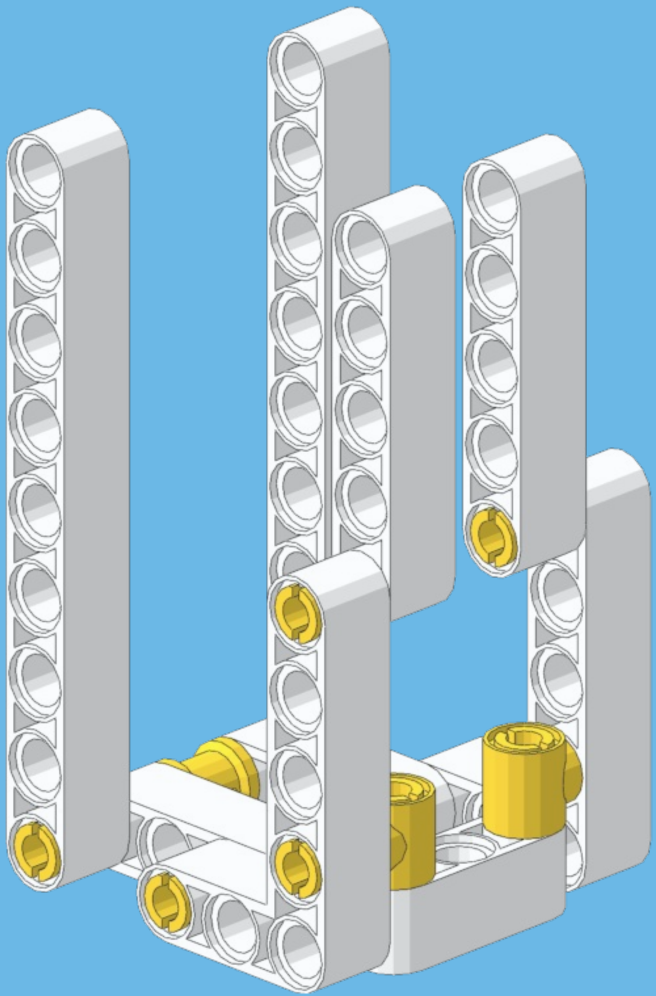
4





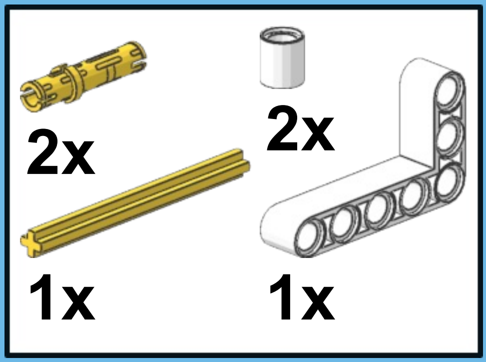
5

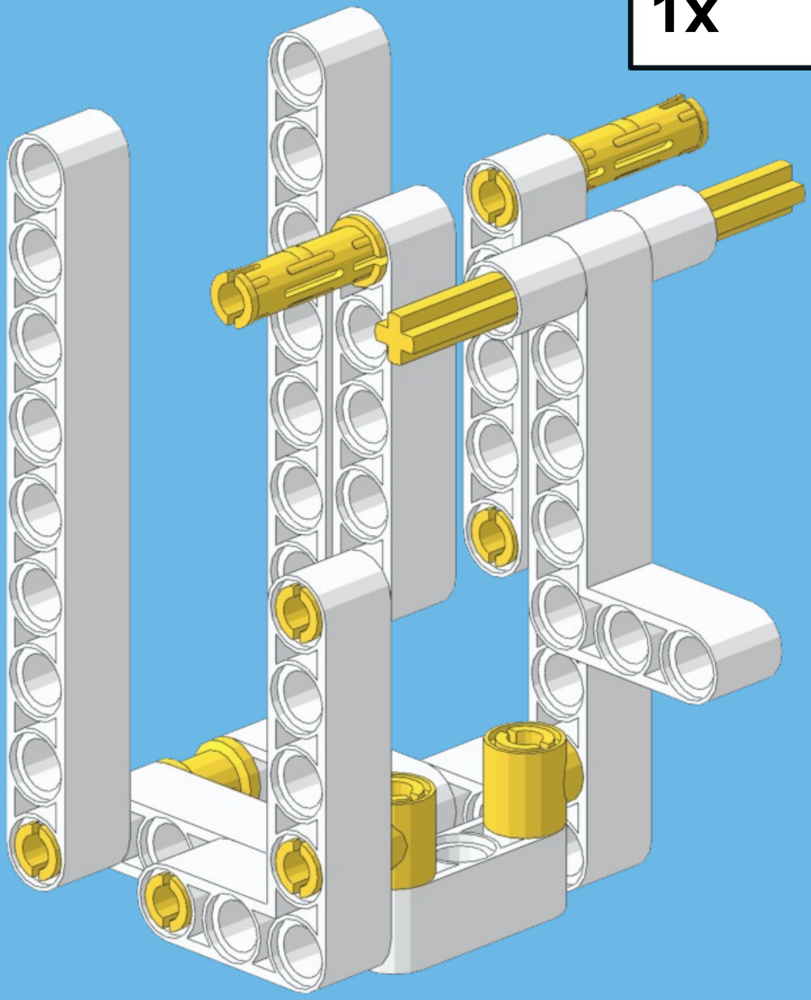




5

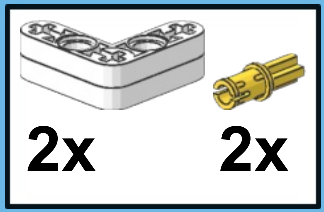
6

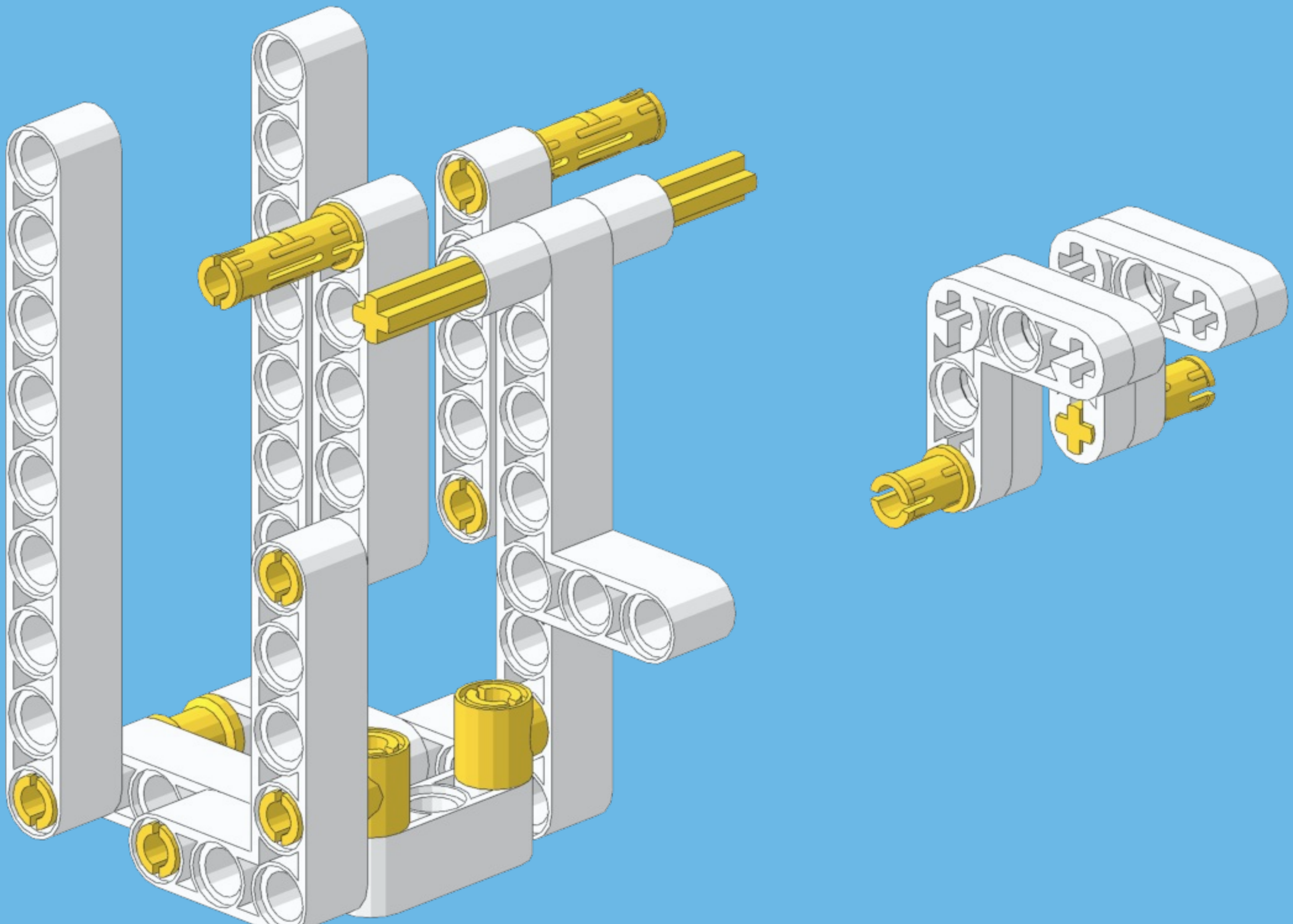




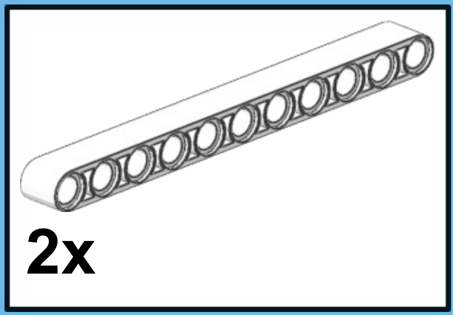
6

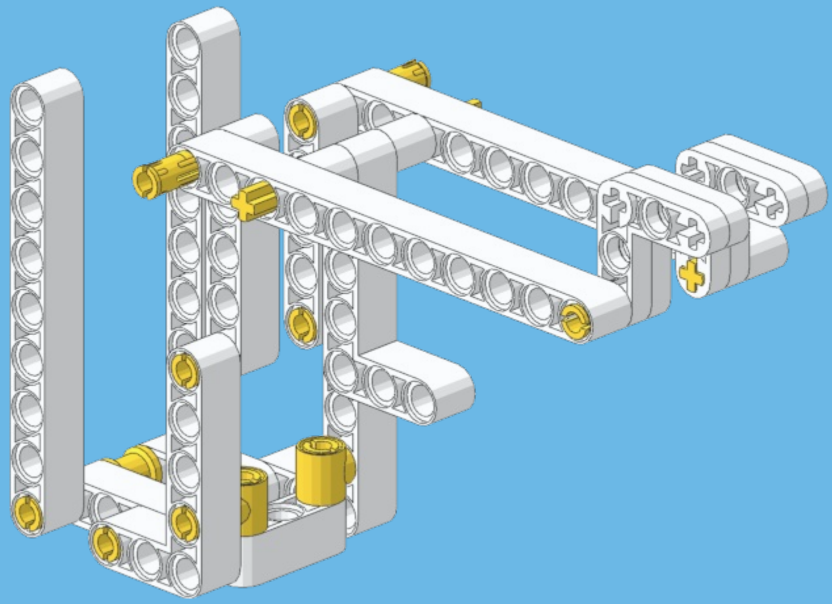
7





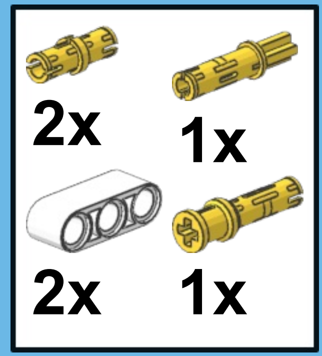
8

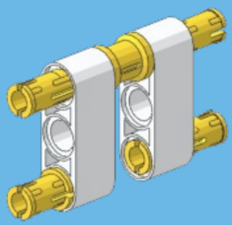


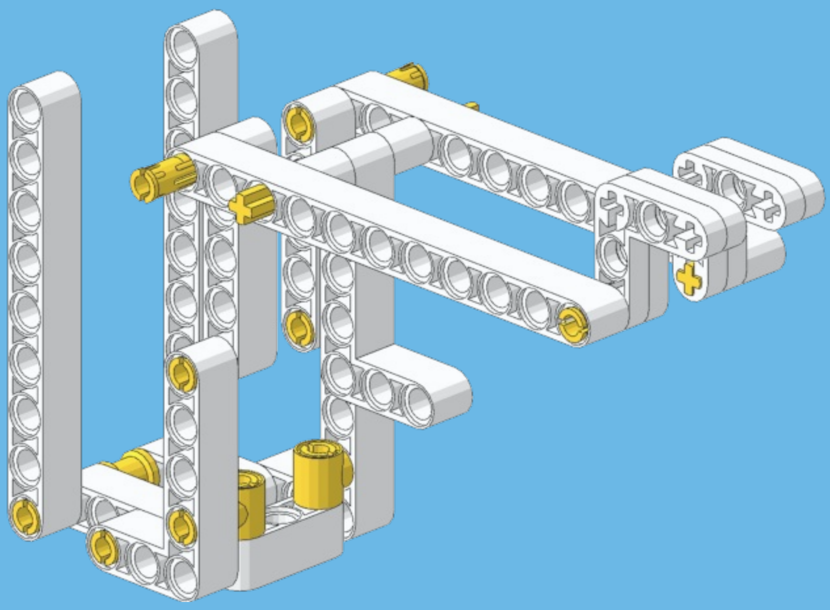


8

9

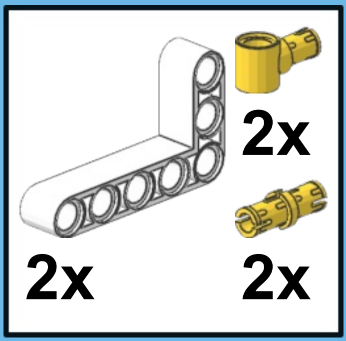


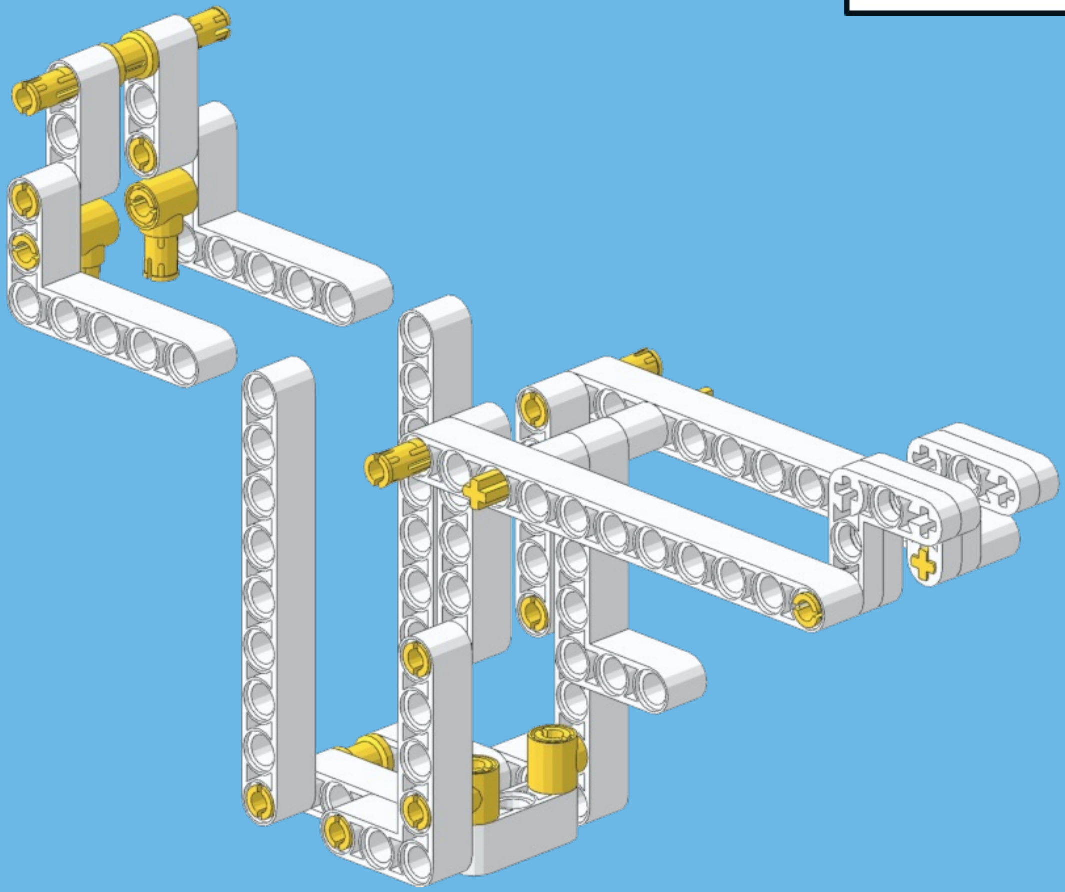




9

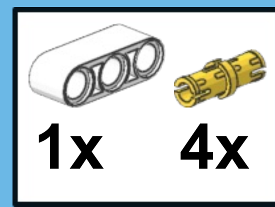
10

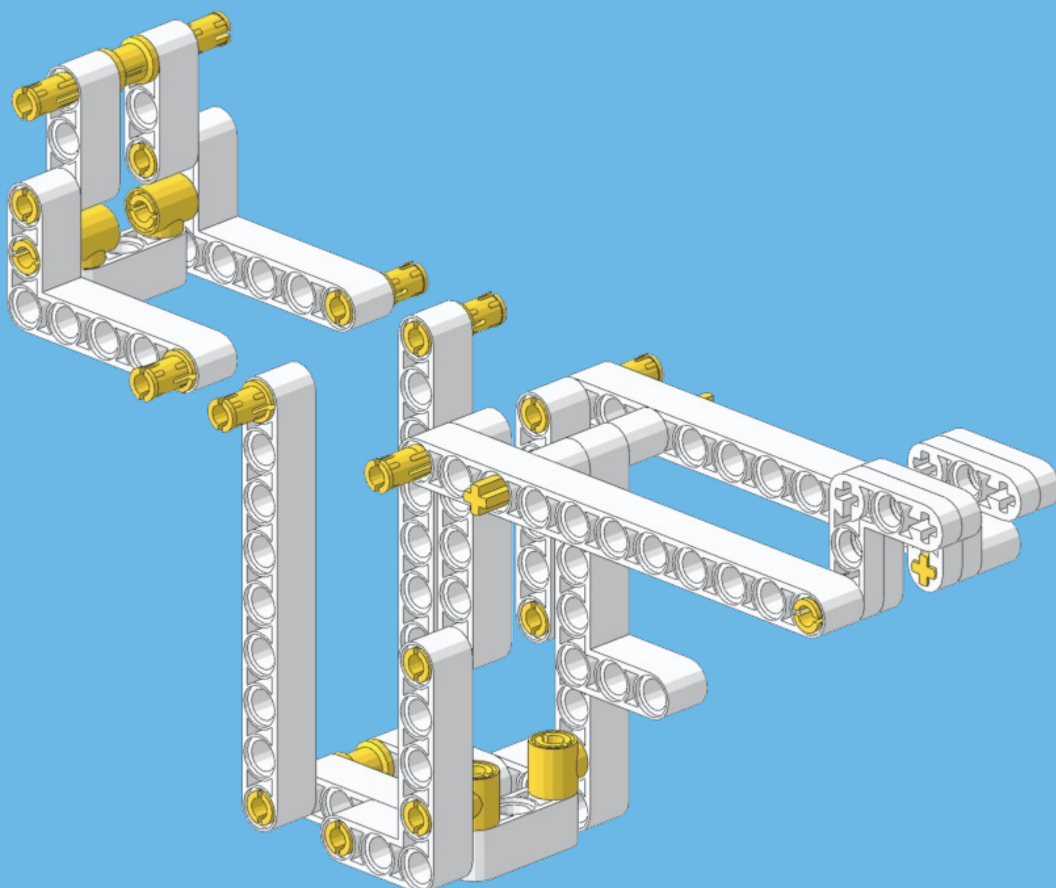




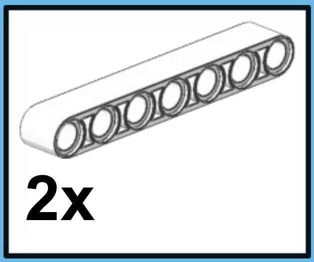
10

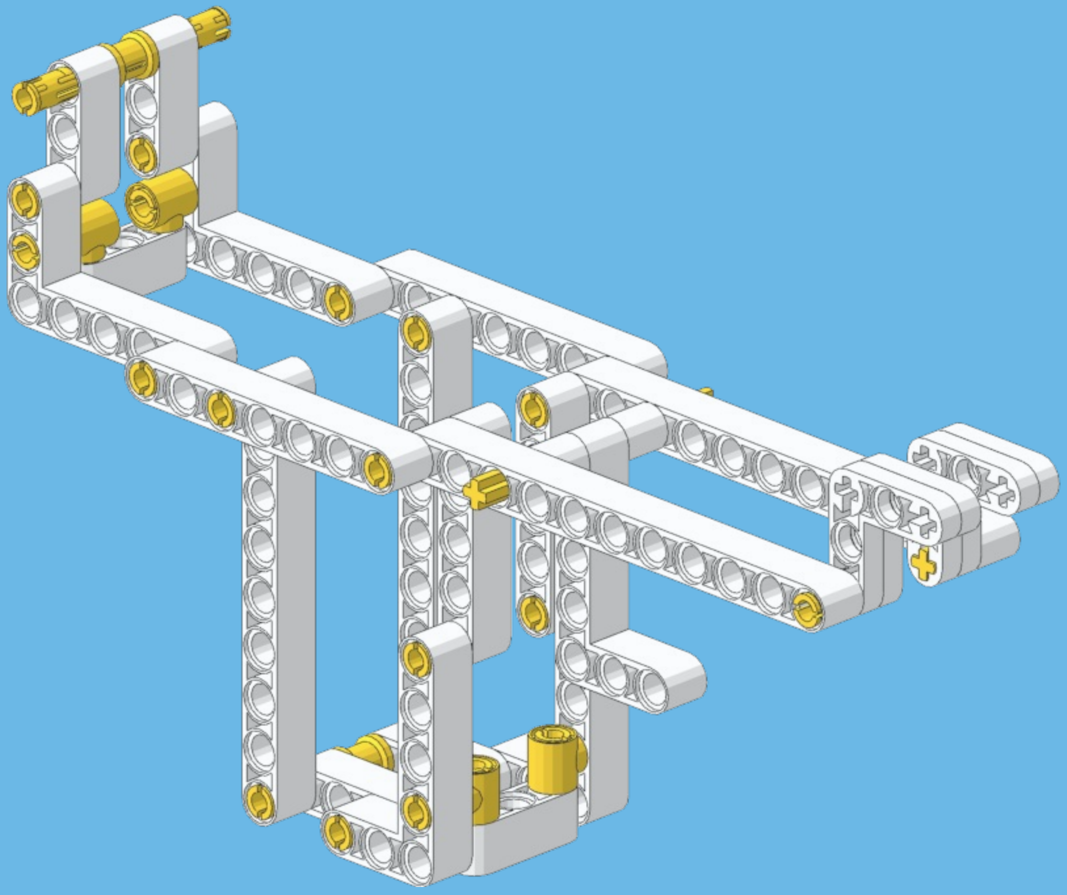
11





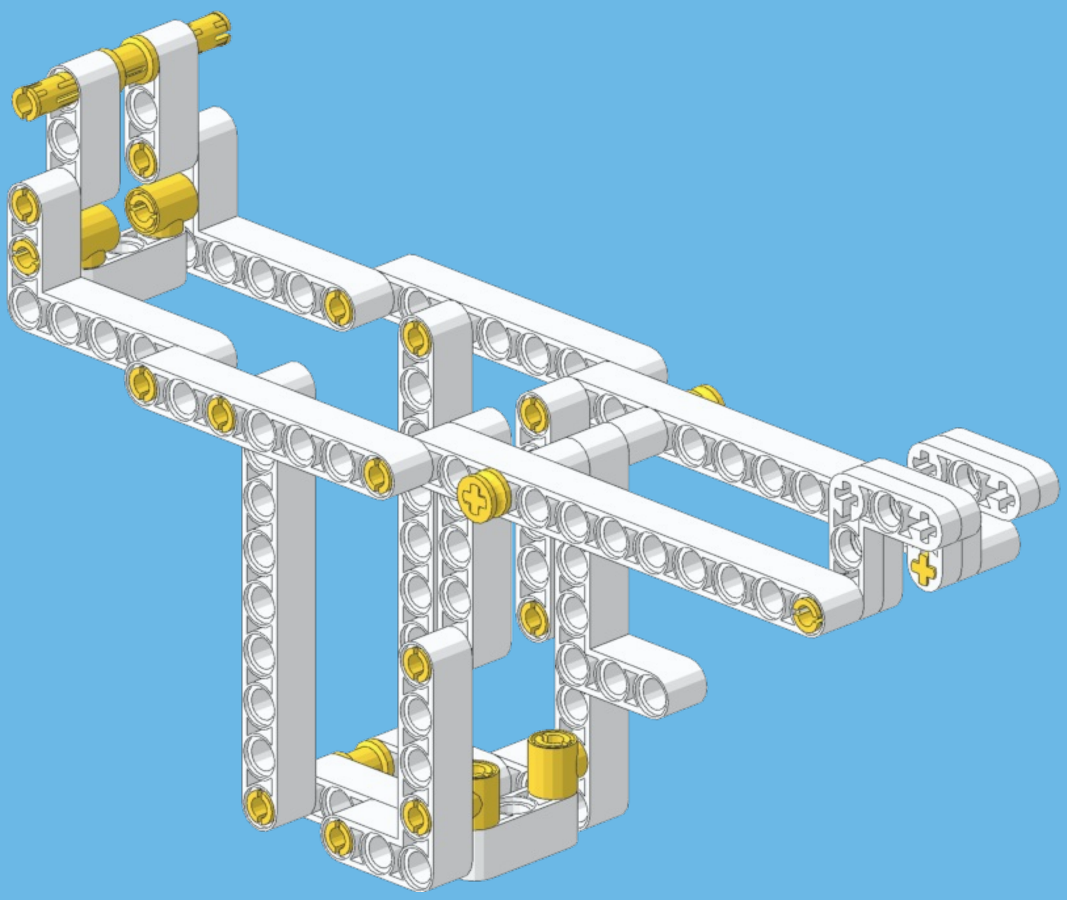
12





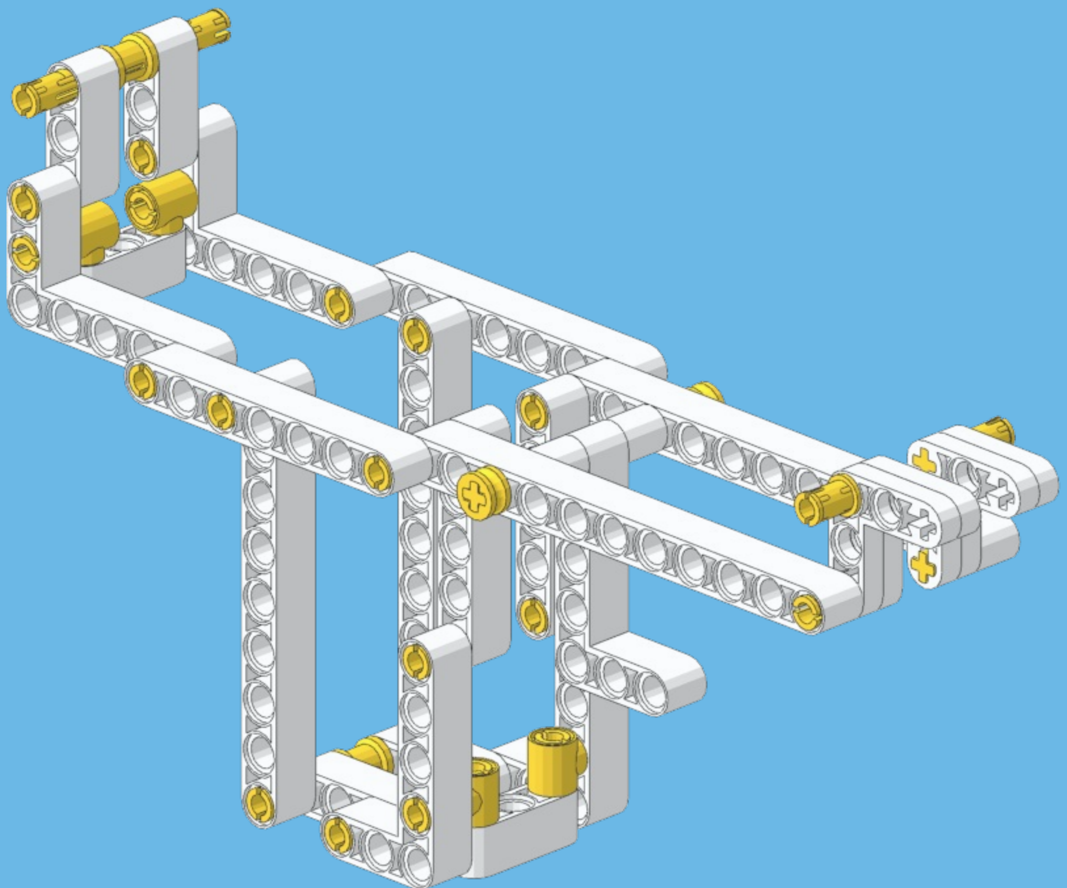
12

13



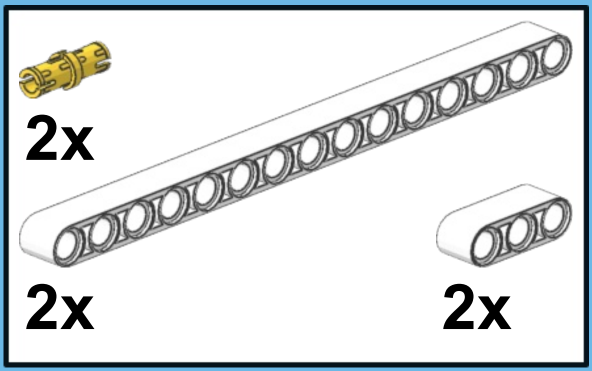
13

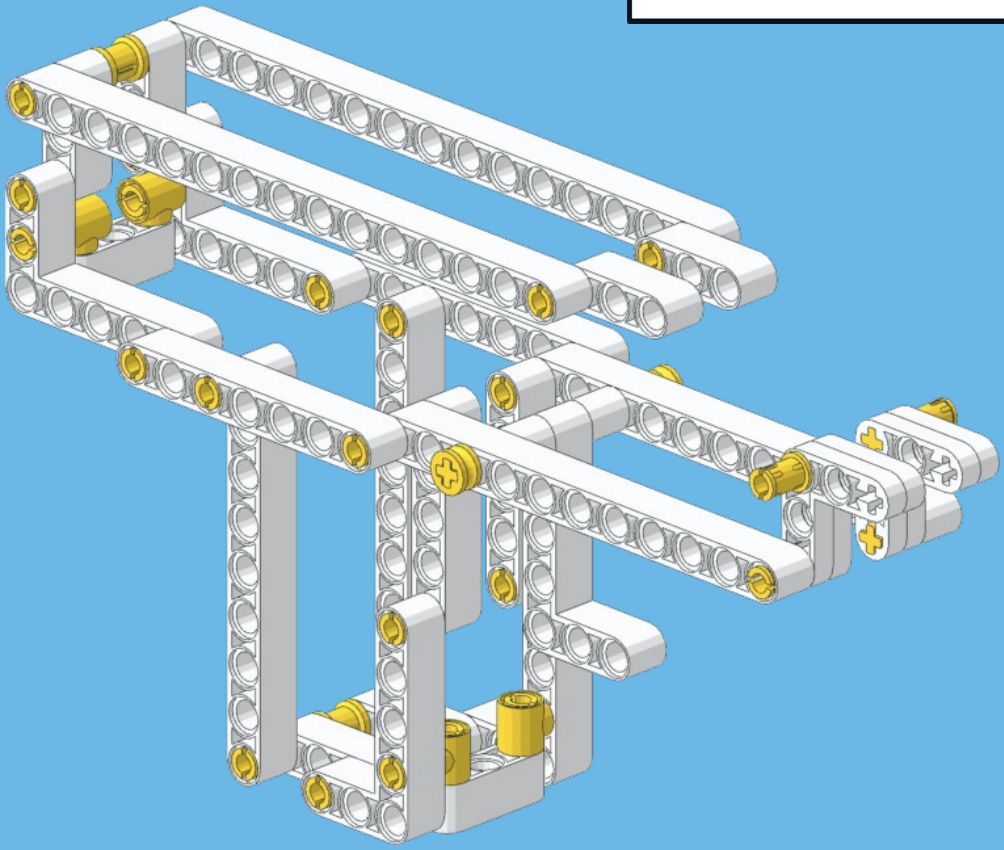
14



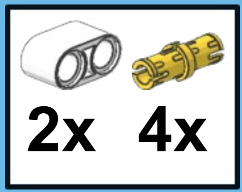
14

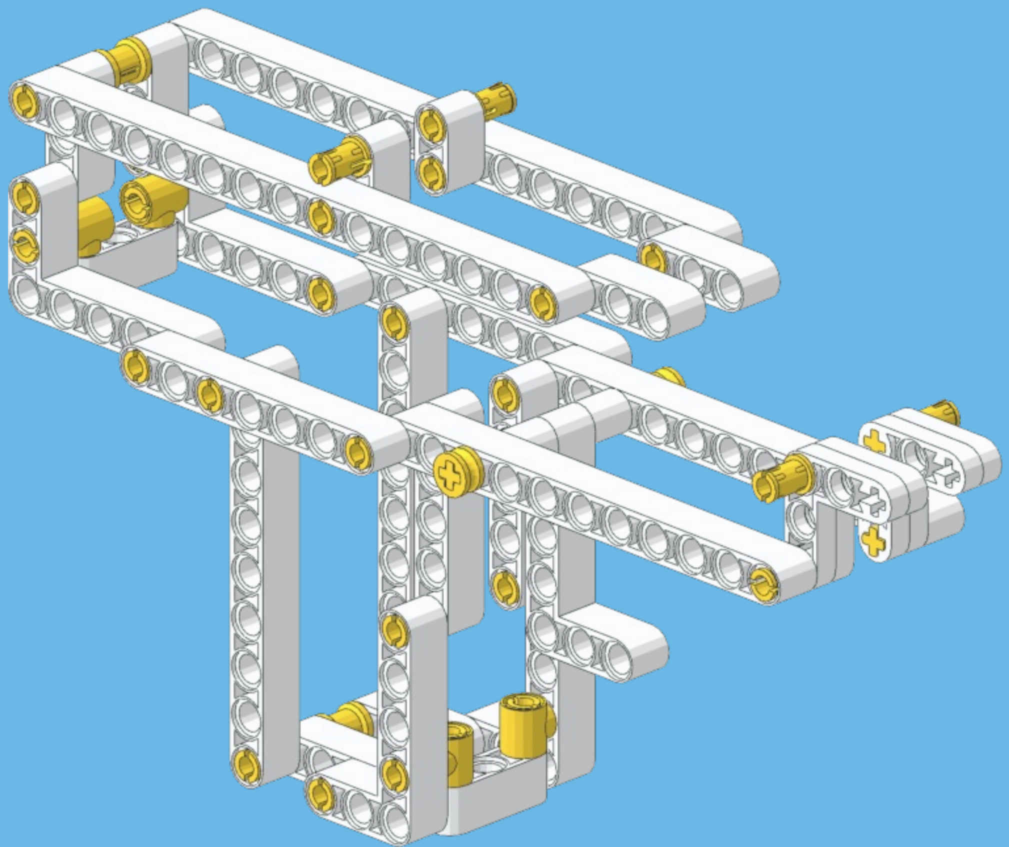
15



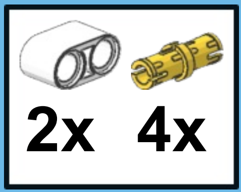


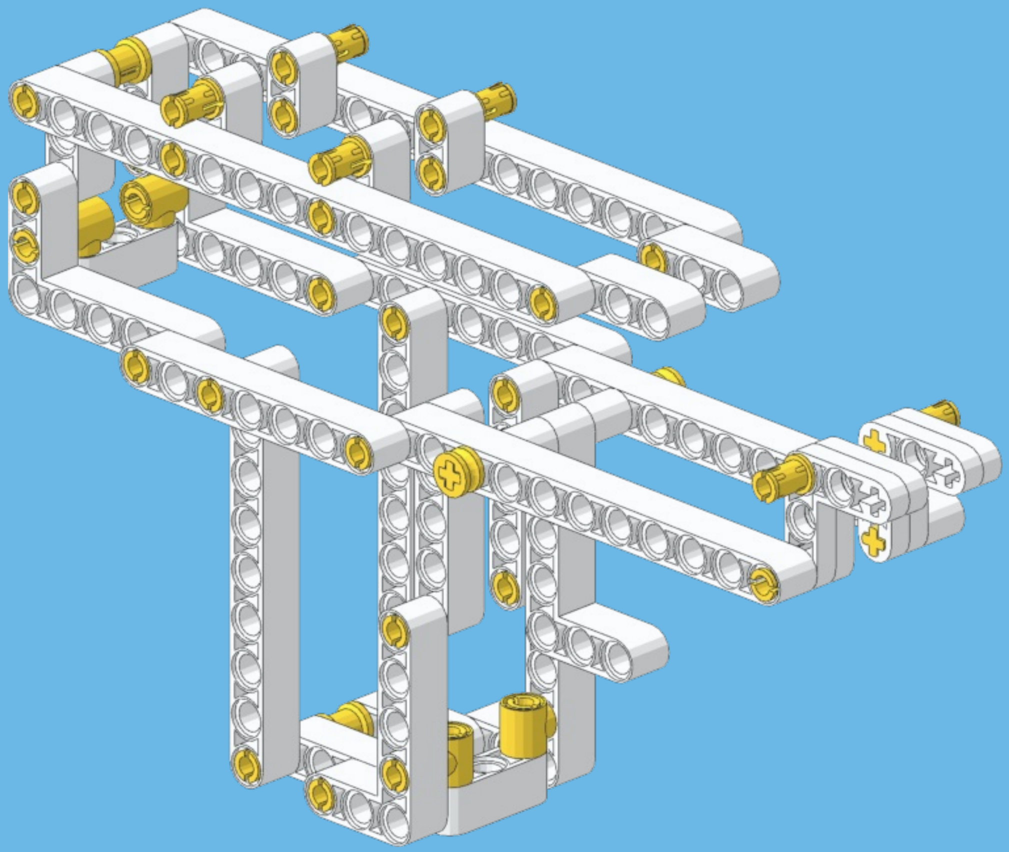
16





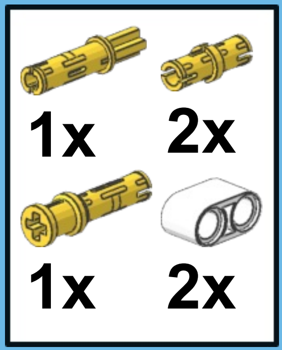
17

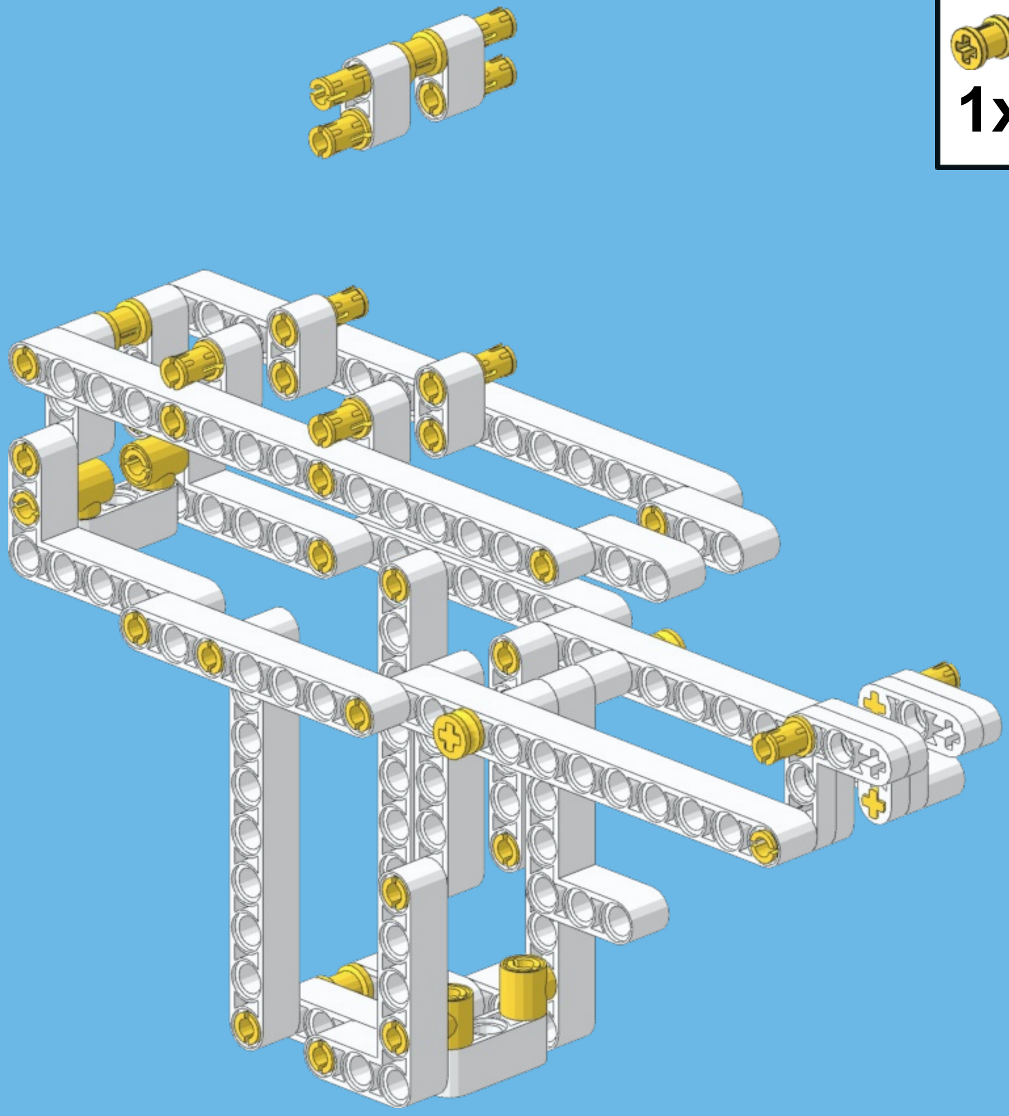




17

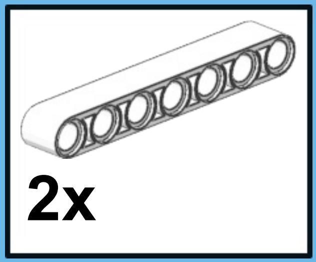
18

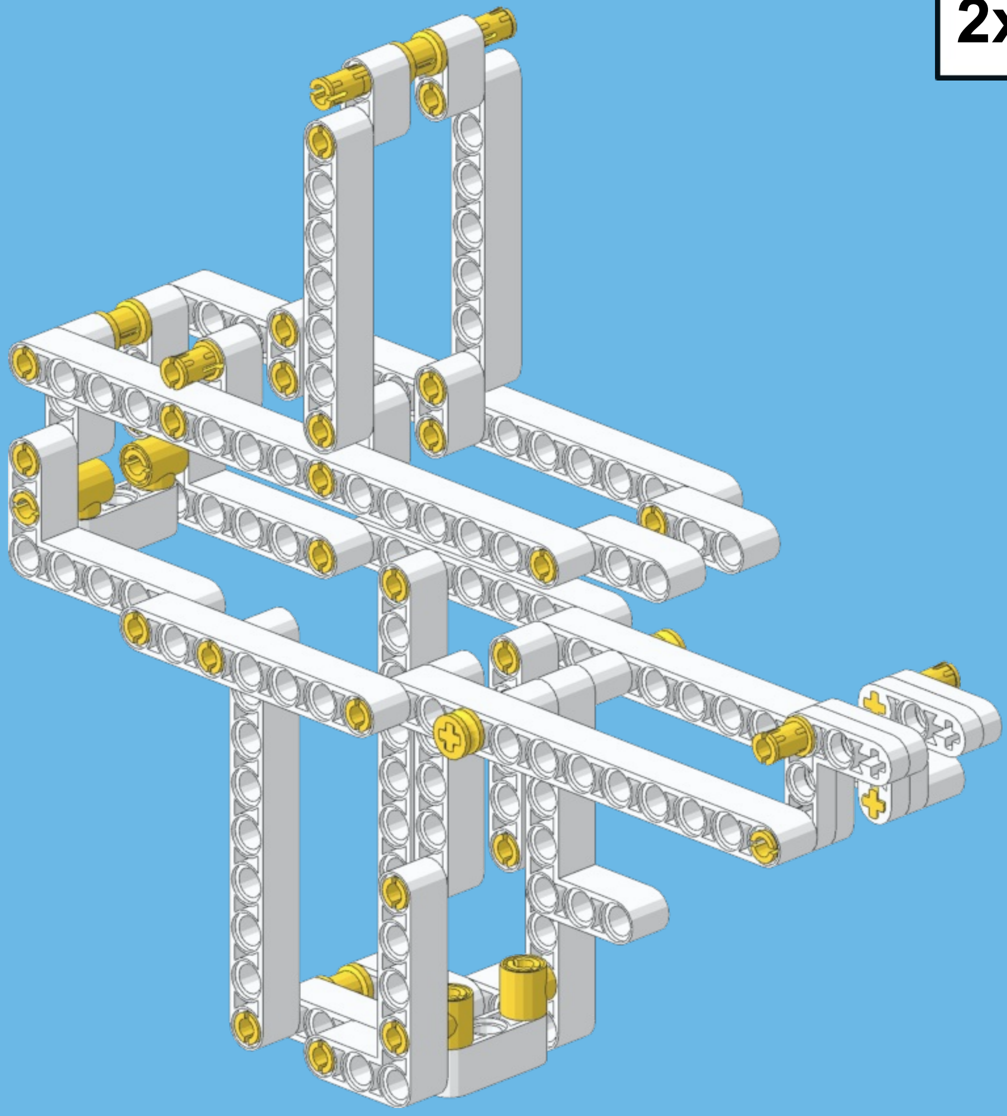




18

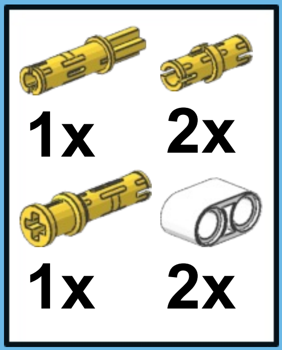
19

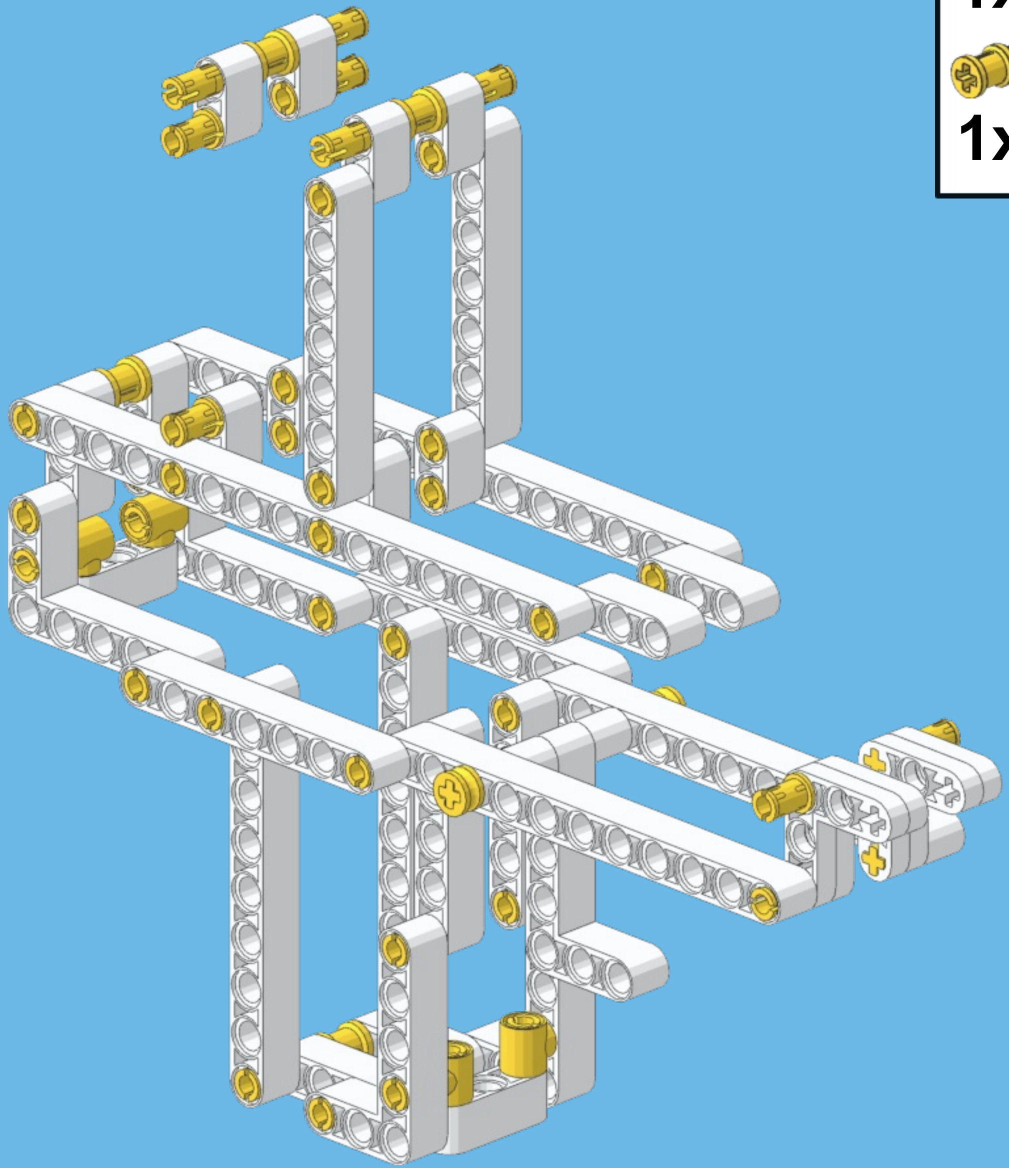




19

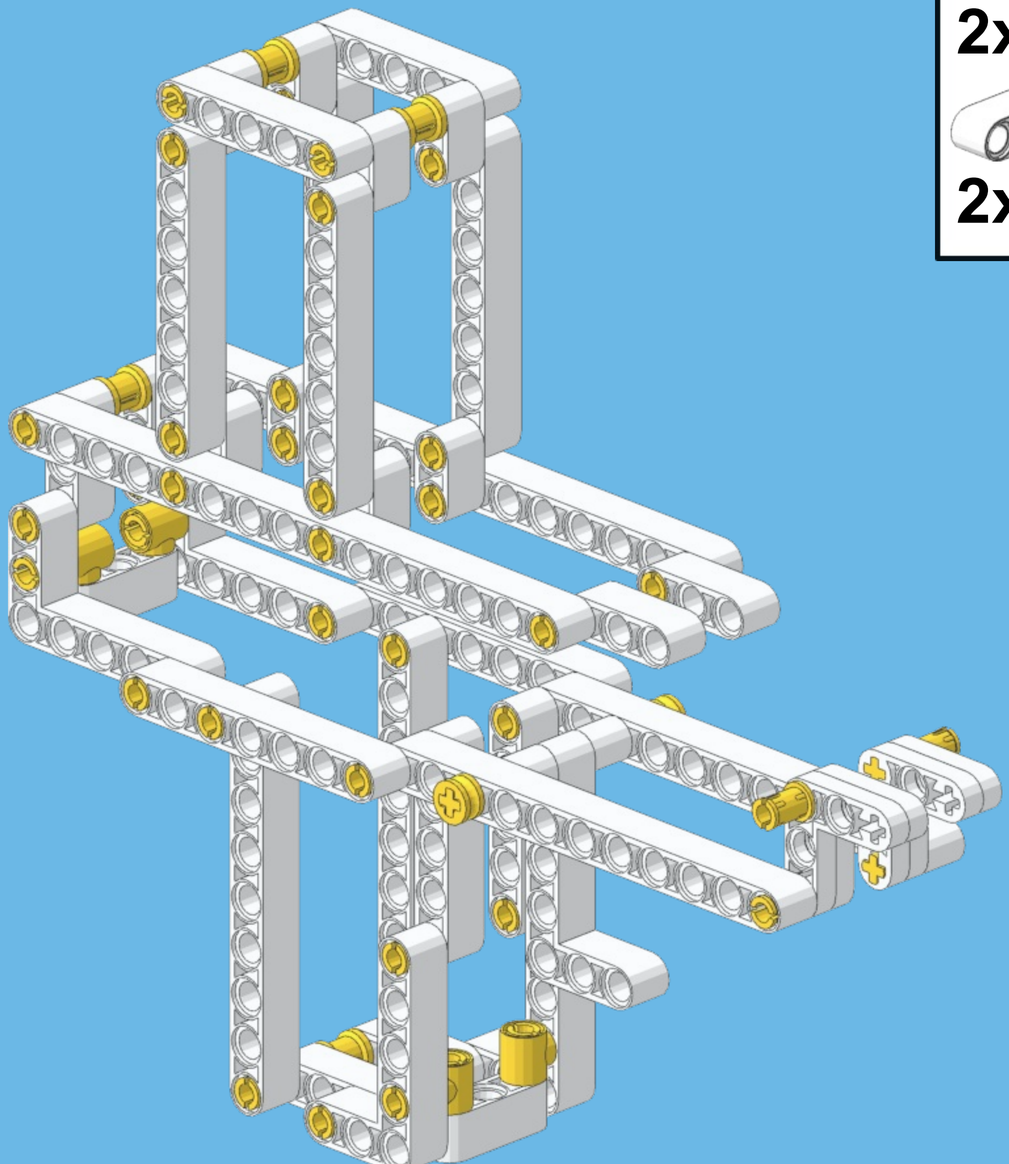
20

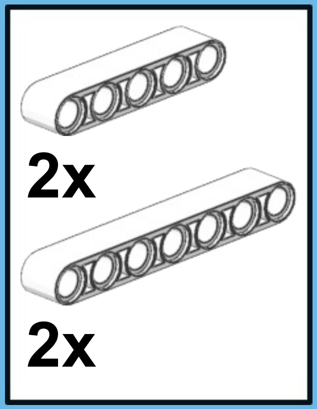




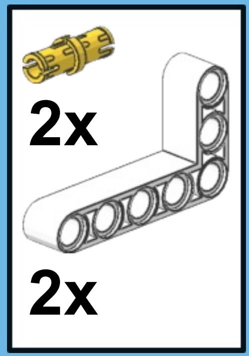
20

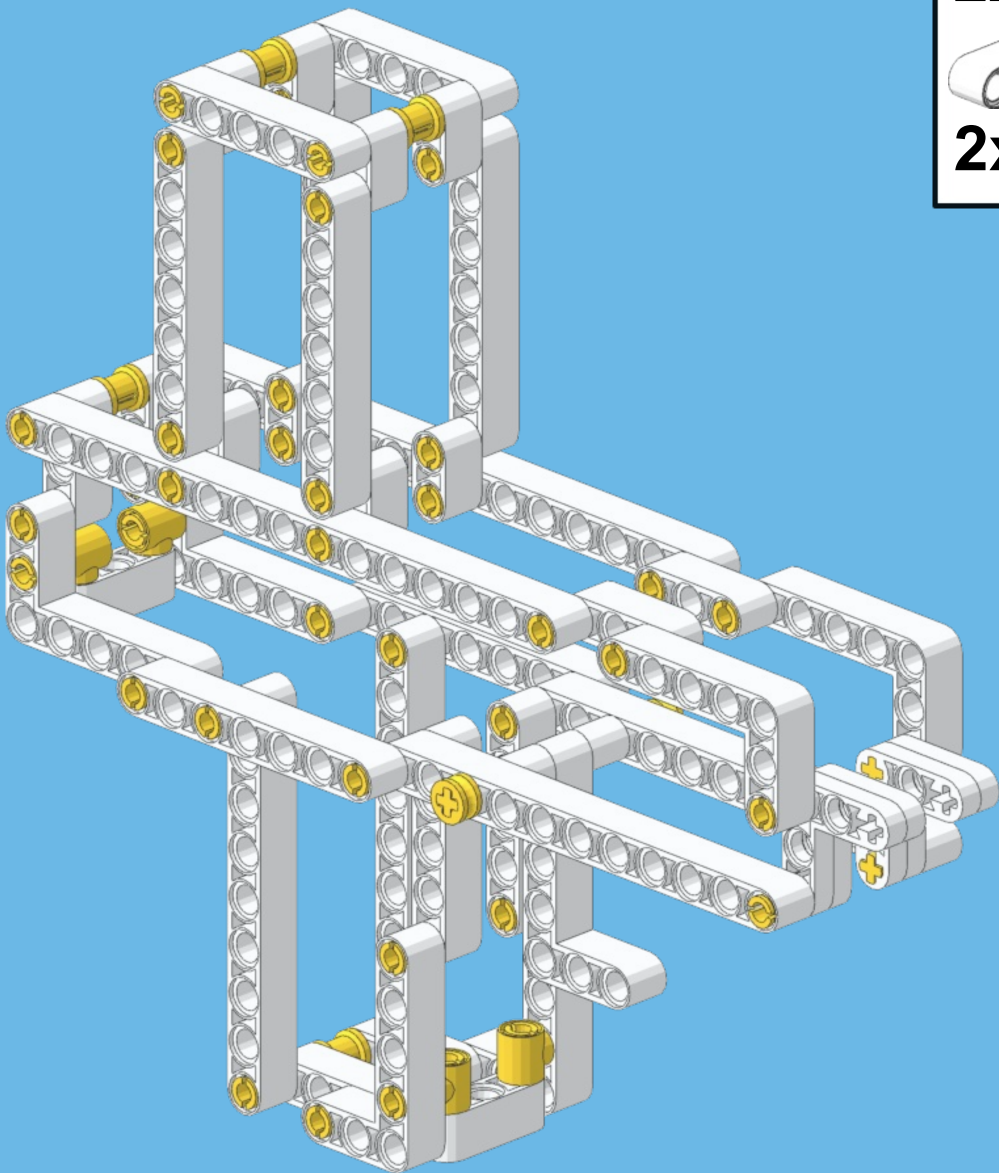
21



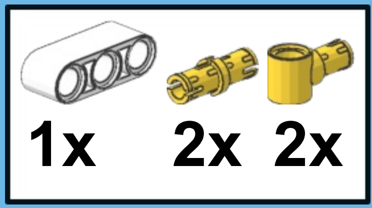


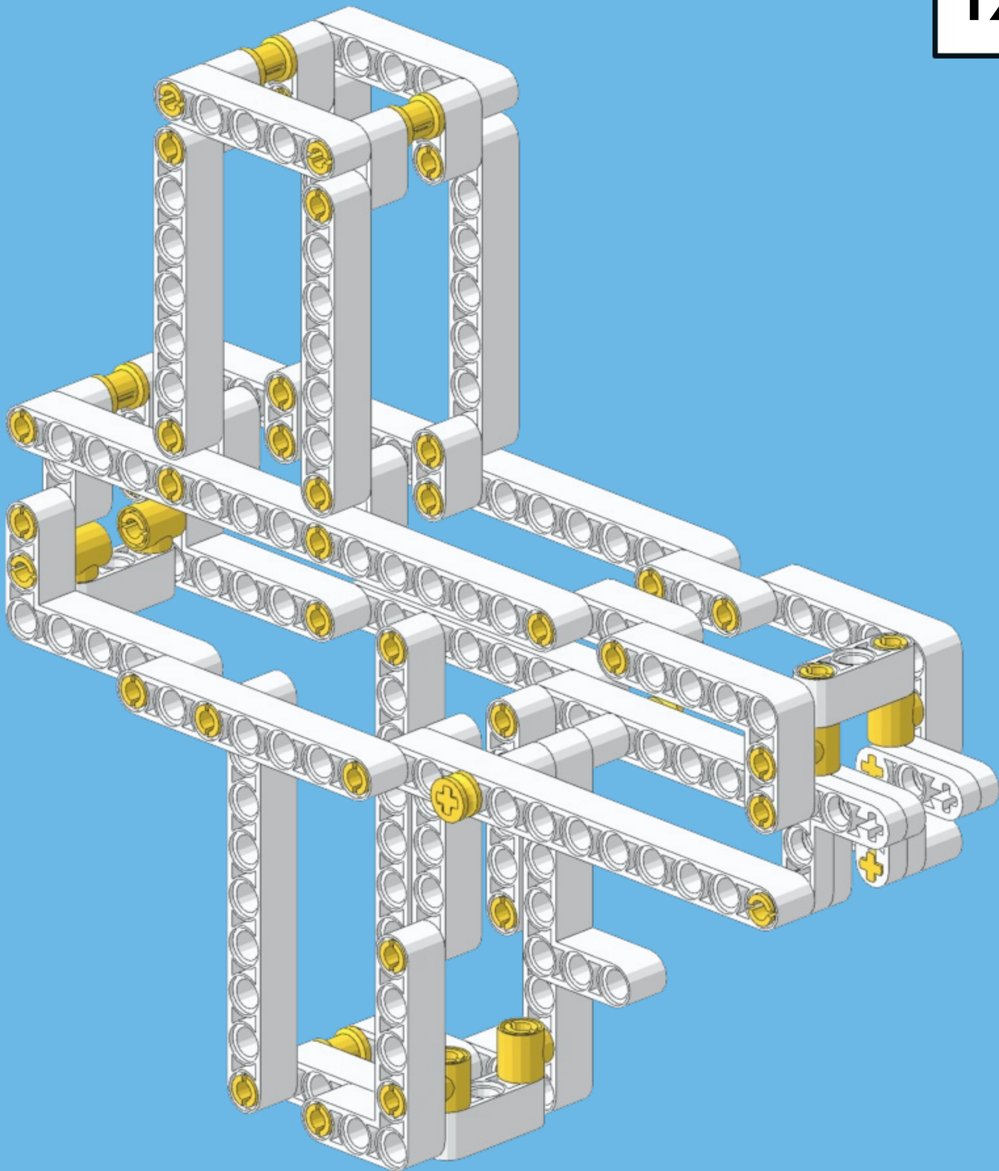
22



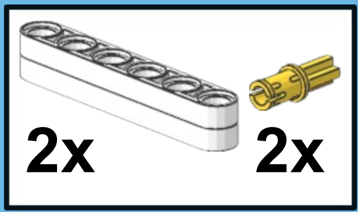


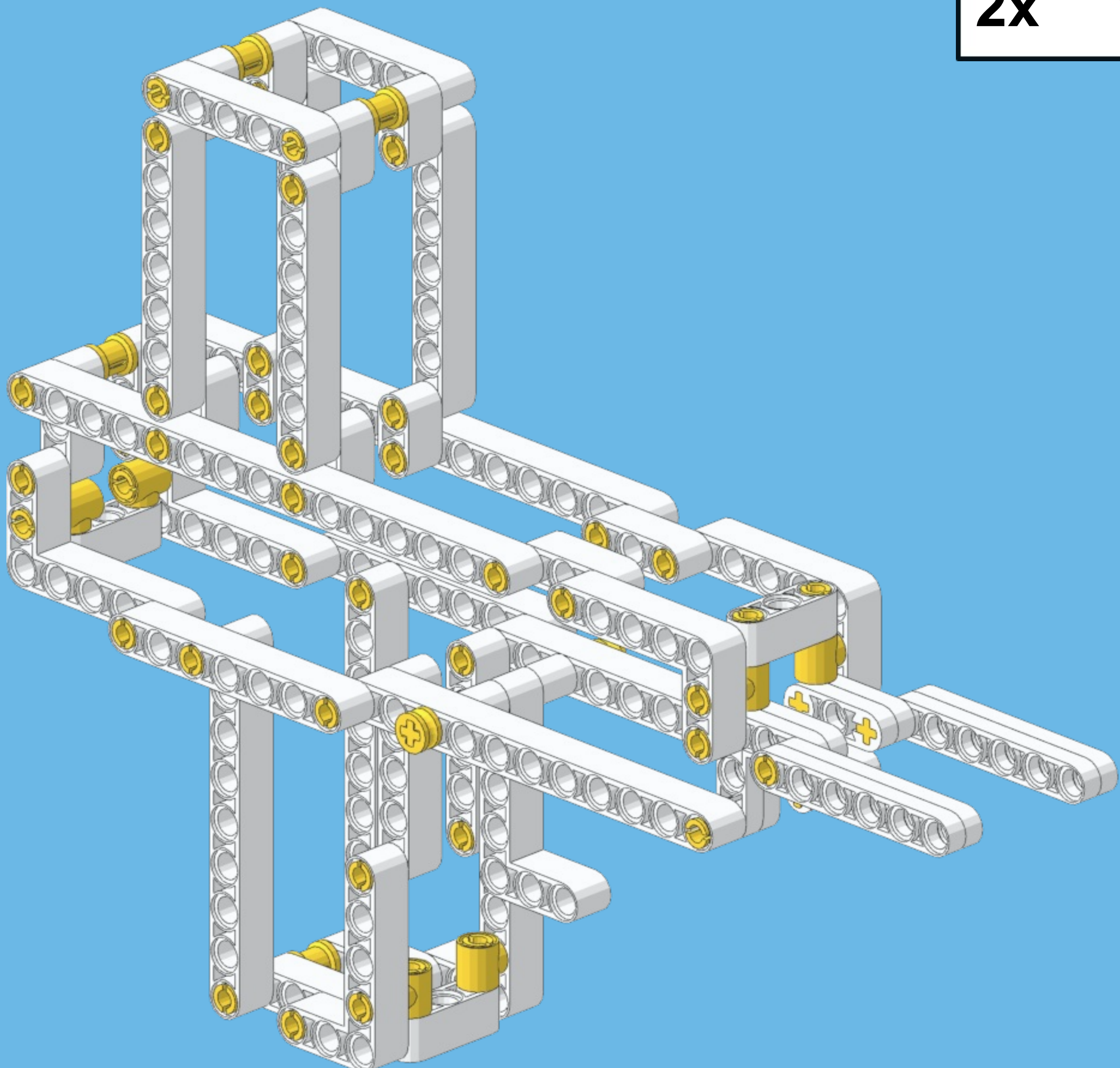
23





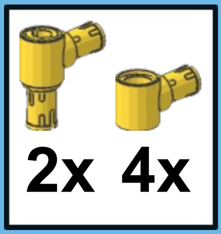
24

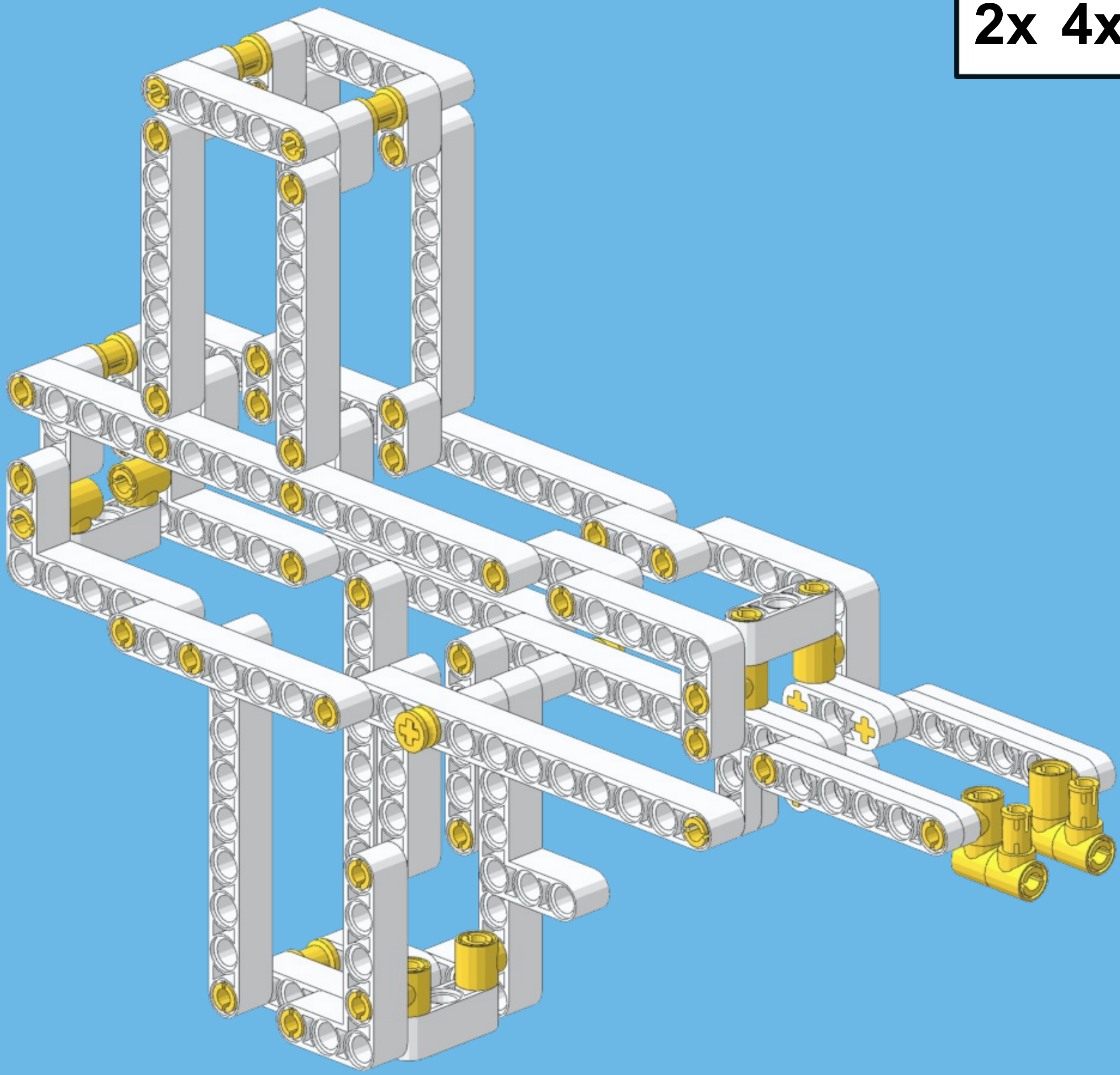




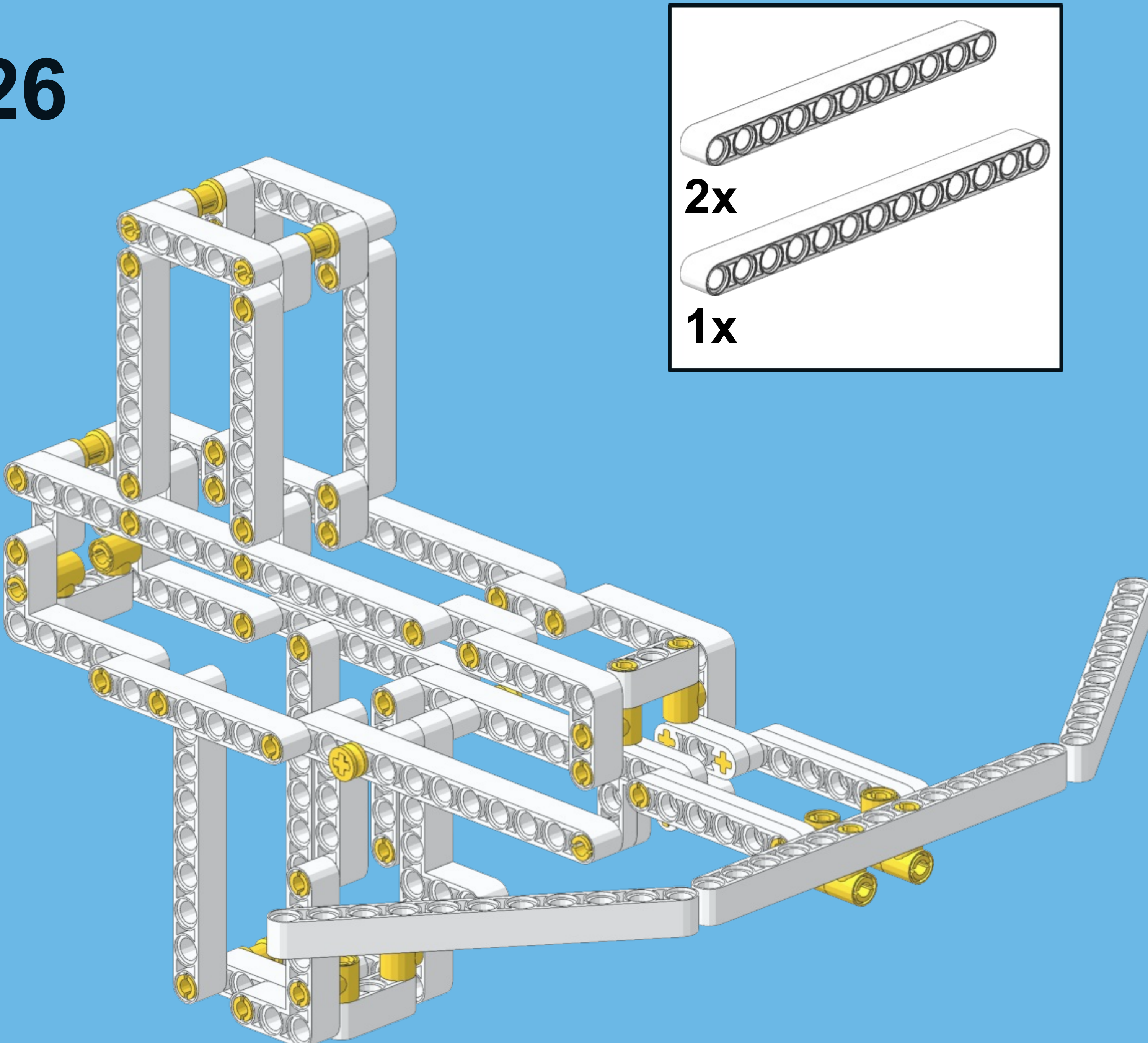
24

25



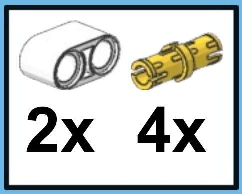


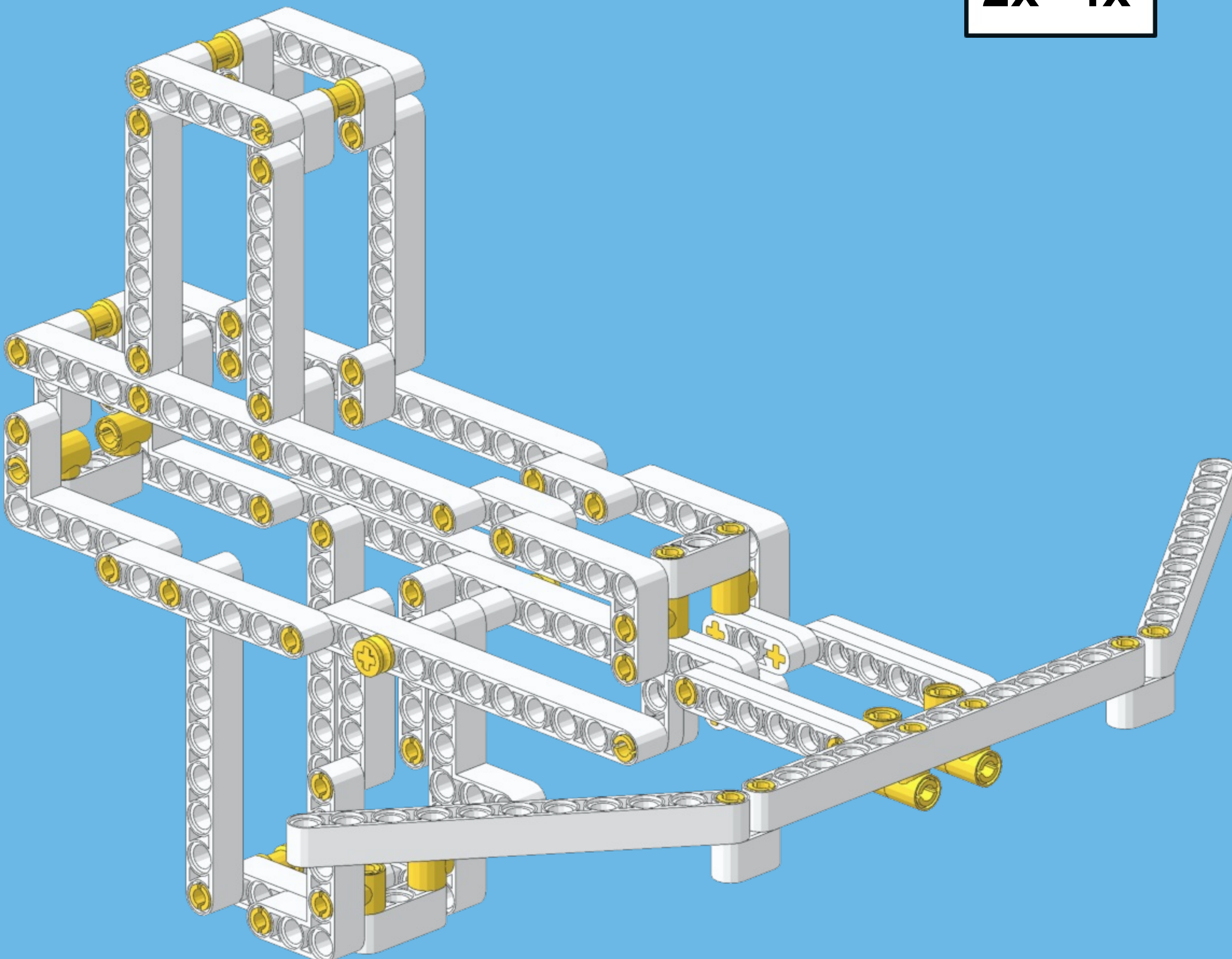
26



26

27





27

The End