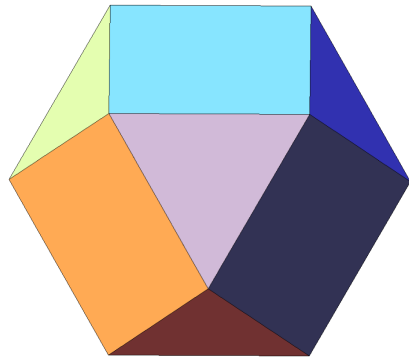


MATH CHALLENGE: POLYHEDRA RACE

Duration: 30 minutes. **NO CALCULATORS, PHONES, OR COMPUTING DEVICES!**

Overview

The picture below shows a cuboctahedron:



ESSENTIAL DEFINITION Every corner looks the same: as you go around it, you see a triangle, a square, a triangle, and a square. By recording the number of sides of each of these shapes that meet at any corner, we can call it a (3,4,3,4) polyhedron. Or we can say, “The recipe for a cuboctahedron is (3,4,3,4).” If your polyhedron does not meet this definition, then we’ll say the recipe is “impossible”, even if you can construct it.

Read all of this page carefully first! If you still have questions about the **DEFINITION** or the **INSTRUCTIONS**, please ask.

Instructions

1. Make a model of a cuboctahedron.
2. How many triangles and how many squares are needed to make a cuboctahedron? Use your result to begin filling out the table on the next page.
3. In building the shapes, **fill in the chart on the next page**. You can get points for constructing and for counting polygon components. All values must be correct for you to get the points for each item (construction, polygonal parts, and VEF values).
Warning: some of those recipes are impossible!
4. Which recipes are impossible? Place an X in the IMPOSSIBLE column if the recipe is impossible. Leave the rest of that recipe’s boxes empty.
5. The final row of the table has an empty spot for you to enter **your own recipe** that meets the definition!

Recipe	Points for construction	Points for numbers alone	Triangles	Squares	Pentagons	IMPOSSIBLE (X)	Points for computing Faces/Vertices/edges	Faces ¹	Edges	Vertices	ROW SCORE
(3,4,3,4)	5	5					3				
(4,4,4)	2	2					2				
(3,4,5)	3	3					2				
(3,3,3)	2	2					2				
(3,3,5)	2	2					2				
(5,5,5)	5	5					3				
(3,3,3,3)	4	4					2				
(3,3,3,5)	3	3					2				
(3,5,3,5)	10	10					6				
(4,5,4,5)	3	3					2				
(3,3,3,3,3)	8	8					10				
(3,4,3,5)	2	2					2				
	7	7					7				
Possible points	56	56					45				

Additional space for explanations:

¹ A polyhedron consists of polygonal *faces*, the polygons' sides are known as *edges*, and the corners known

