1. Test Utility::coplanar_det()
12 faces:

A	0, 1, 0	1, 1, 1	0, 1, 1	**B**	1, 1, 0	1, 0, 1	1, 1, 1
C	0, 1, 1	1, 0, 1	0, 0, 1	**D**	0, 0, 0	0, 1, 1	0, 0, 1
E	0, 0, 0	1, 0, 0	0, 1, 0	**F**	1, 0, 0	0, 0, 1	1, 0, 1
G	0, 1, 0	1, 0, 1	0, 1, 1	**H**	1, 1, 0	1, 0, 1	0, 0, 1
I	1, 1, 0	0, 0, 1	0, 1, 1	**J**	0, 0, 0	1, 0, 1	0, 1, 1
K	1, 1, 0	1, 0, 1	0, 1, 1	**L**	0, 0, 0	0, 1, 1	1, 1, 0

7 points:

P1	1.5, 0.5, 0.5	**P2**	−0.5, 0.5, 0.5	**P3**	0.5, 0.5, −0.5	**P4**	0.5, 0.5, 1.5
P5	0.5, 1.5, 0.5	**P6**	0.5, −0.5, 0.5	**P7**	0.5, 0.5, 0.5		

Expected results:

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**
A	-	-	-	-	+	-	-
B	+	-	-	-	-	-	-
C	-	-	-	+	-	-	-
D	-	+	-	-	-	-	-
E	-	-	+	-	-	-	-
F	-	-	-	-	-	+	-
G	+	-	0	0	+	-	0
H	0	0	-	+	+	-	0
I	+	-	-	+	0	0	0
J	+	-	+	-	+	-	+
K	+	-	-	+	+	-	-
L	+	-	-	+	-	+	+

2. Test Utility::before/behind/coplanar()
H against A, D, B, E

	A	**B**	**D**	**E**
H	Before	Before	Span	Behind

3. Test Utility::intersection()
K,
E1 (0, 0, 0)-(1, 1, 1); E2(0, 0, 1)-(1, 1, 1)

	E1 (0, 0, 0)-(1, 1, 1)	**E2 (0, 0, 1)-(1, 1, 1)**
K	(2/3, 2/3, 2/3)	(0.5, 0.5, 1)

4. Test Utility::split_polygon()
split I with G
F: (0, 1, 1) – (1, 1, 0) – (0.5, 0.5, 0.5)
B: (0, 1, 1) - (0.5, 0.5, 0.5) – (0, 0, 1)

5. Test BSPTree::add_polygon()
4 points:

A	(–0.5, 1, 0)
B	(0.5, 0, 0)
C	(–0.5, –1, 0)
D	(0, 0, 1)

4 faces:

ABD	(–0.5, 1, 0)	(0.5, 0, 0)	(0, 0, 1)	**BCD**	(0.5, 0, 0)	(–0.5, –1, 0)	(0, 0, 1)
CAD	(–0.5, –1, 0)	(–0.5, 1, 0)	(0, 0, 1)	**BAC**	(0.5, 0, 0)	(–0.5, 1, 0)	(–0.5, –1, 0)

6. Test BSPTree::build()
{ABD, BCD, CAD, BAC}

7. Test BSPTree::traverse()
2 points
P1: (1.5, 0, 0.5); P2(0, 0, –1)
P1-> ABD/CBD, CBD/ABD, BAC/CAD, CAD/BAC
P2-> BAC, ABD/CBD/CAD, CBD/CAD/ABD, CAD/ABD/CBDs