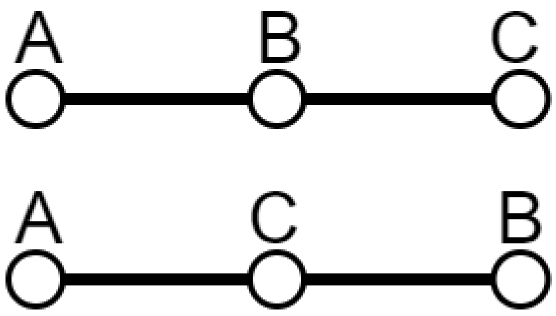
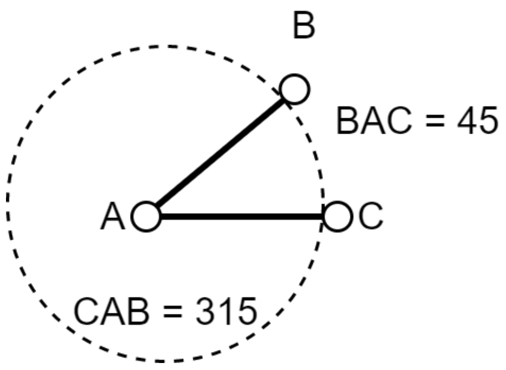
List of inputs:

line([a,b,c,…])

This creates a line with points a,b,c,... on it, making them all collinear points in the process. Be careful to ensure that you enter the points in the right order. Lines ACB and ABC are different after all.





angle(a,b,c,y)

This assigns the value y (in degrees) to the angle ∠ABC. Note, if you enter y as π, the code will assume the angle is $\pi°$

#Make sure to input your angles in the clockwise direction. A computer doesn't read lines and angles in the same way as a person, and so it considers the angle ∠ABC and ∠CBA as different#

congruent(a,b,c,d,e,f)

This tells the code that the triangles ΔABC and ΔDEF are congruent. Again, remember to be specific with your inputs. After all, just because ΔABC is congruent with ΔDEF doesn't mean it's also congruent with ΔDFE.

similar(a,b,c,d,e,f)

This tells the code that the triangles ΔABC and ΔDEF are similar. Remember to be specific with your inputs. After all, just because ΔABC is similar to ΔDEF doesn't mean it's also similar to ΔDFE.

eq_line(a,b,c,d):

This tells the code that line AB is equal in length to line CD, simple as.

length(a,b,y):

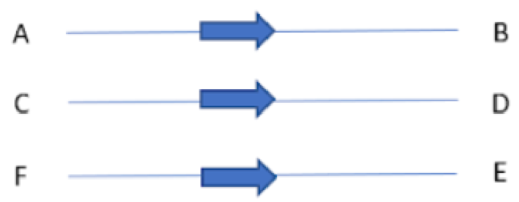
This tells the code that the length of the line AB is y.

perp(a,b,c,d):

This tells the code that line AB is perpendicular to line CD. Just as with angles, remember to be specific in your input. Because the computer reads angles in the **clockwise**(see angle) direction, saying AB is perpendicular to CD is not the same as saying AB is perpendicular to DC.

par(a,b,c,d):

This tells the code that line AB is parallel to line CD. Once again, be careful with your inputs. If line AB is parallel to line CD, it is not parallel with line DC as well. In general, you need to make sure the directions of AB and CD are the same.



AB is parallel to CD and FE but not DC or EF.

#Note, the input is case sensitive, do not use capital letters in the input#

Output:

To run the program, just open the file named solver, it will solve the given problem automatically.

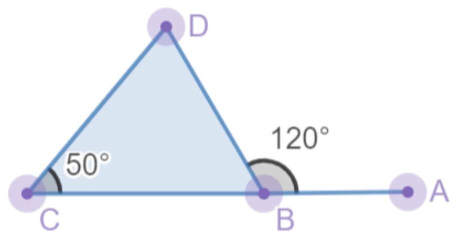
To get outputs, you unfortunately have to deal with the prolog interface (Sorry about that).

The output commands are as follows.

1. out_line(a,b). : returns true if a line AB exists.
2. out_angle(a,b,c). : returns the value of the angle ∠ABC in degrees.
3. out_congruent(a,b,c,d,e,f). : returns true if the triangles ΔABC and ΔDEF are congruent.
4. out_similar(a,b,c,d,e,f). : returns true if the triangles ΔABC and ΔDEF are similar.
5. out_eq_line(a,b,c,d). : return true if the lines AB and CD are equal in length.
6. out_length(a,b). : returns the length of the line AB.
7. out_par(a,b,c,d). : returns true if the lines AB and CD are parallel.
8. out_perp(a,b,c,d). : returns true if the lines AB and CD are perpendicular.

#Be careful when entering the output code. Prolog is case sensitive, so you cannot use capital letters in calling the output. Also make sure to end the output functions in a period to get the output.#

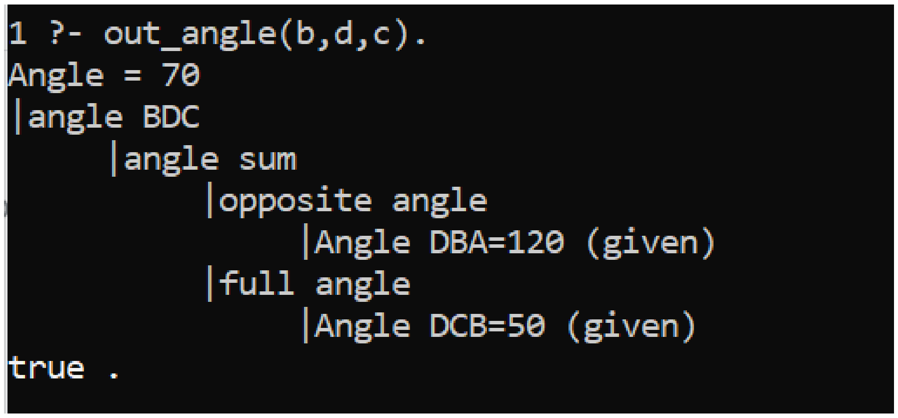
Example:



Find ∠BDC

Input:
line([a,b,c])
line([c,d])
line([b,d])
angle(d,b,a,120)
angle(d,c,b,50)

Output:
out_angle(b,d,c)



```
1 ?- out_angle(b,d,c).
Angle = 70
|angle BDC
     |angle sum
          |opposite angle
               |Angle DBA=120 (given)
          |full angle
               |Angle DCB=50 (given)
true .
```