enum Face {U, D, L, R. F, B);

enum RotationDirection {NegDouble = -1, CCW = -1, Zero = 0, CW = 1, Double = 2);

void rotate(Face F, RotationDirection dir)

{}

RotationDirection sideFacesToRotation(Face from, Face to) {

if (from == to)

return RotationDirection.Zero;

int x1; int y1; int x2; int y2;

switch (from) {

case Face.L:

x1 = -1; y1 = 0;

case Face.R:

x1 = 1; y1 = 0;

case Face.F:

x1 = 0; y1 = -1;

case Face.B:

x1 = 0; y1 = 1; }

switch (to) {

case Face.L:

x2 = -1; y2 = 0;

case Face.R:

x2 = 1; y2 = 0;

case Face.F:

x2 = 0; y2 = -1;

case Face.B:

x2 = 0; y2 = 1; }

int dir = (x1 == 0) ? x2/y1 : -y2/x1;

if (dir == 0)

return RotationDirection.Double;

else

return (RotationDirection)dir;

}

* Cross edge:
  + top layer, oriented right (y == 1 && colorY == white)
    - correct location: do nothing
    - rotate side face with edge (CW)
      * get face and turn it
    - turn top face to bring in correct position
      * get other edge color
      * find face with that color
      * get rotation amount
        + rotate(Face.U, sideFacesToRotation(edge expected face, edge original face);
    - unrotate side face with edge (CCW)
      * get face and turn it
    - unturn top face to bring back to original position
      * do all that stuff again, or save it in vars
  + top layer, oriented wrong (y == 1 && color1 != white)
    - rotate side face with edge (CW)
      * get face and turn it
    - turn top face to bring in correct position
      * from edge expected face to face with nonwhite side of edge
    - unrotate new side face with edge (CW)
      * get new face and turn it
    - unturn top face to bring back to original position
      * do all that stuff again, or save it in vars
  + side layer (edge orientation == XY OR y == 0)
    - turn top face to bring in correct position
      * from edge expected face to face with nonwhite side of edge
    - rotate side face with edge
      * get new face and turn it in direction based on position
    - unturn top face to bring back to original position
      * do all that stuff again, or save it in vars
  + bottom layer, oriented upside down (y == -1 && colorY = white)
    - turn bottom face to be in correct position
      * rotate(Face.B, sideFacesToRotation(edge expected face, edge original face); //double negative, one from (expected->original) and one from bottom face
    - turn correct face twice
  + bottom layer, oriented otherwise (y == -1 && colorY != white)
    - turn bottom face to be in correct position
      * rotate(Face.B, sideFacesToRotation(edge expected face, edge original face); //double negative, one from (expected->original) and one from bottom face
    - rotate entire cube so correct face is F
    - F U L’ U’
* Top layer corners
  + Top layer, oriented correctly
    - Correct location, do nothing