```c
#pragma config(Sensor, dgtl1,  RightEncoder,   sensorQuadEncoder)
#pragma config(Sensor, dgtl3,  LeftEncoder,    sensorQuadEncoder)
#pragma config(Motor,  port1,           RightMotor,    tmotorVex393_HBridge, PIDControl, reversed, encoderPort, dgtl3)
#pragma config(Motor,  port10,          LeftMotor,     tmotorVex393_HBridge, PIDControl, encoderPort, dgtl1)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

void move(int direction, int maxSpeed, int pulses){
  //direction 1 = forward, 2 = backwards, 3 = left, 4 = right
  int errorLeft = 0, errorRight = 0;
  int priorErrorLeft = 0, priorErrorRight = 0;
  float KP = .25, bias = 25;
  int iterationTime = 1;
  int outputLeft = 0, outputRight = 0;
  for(int c=1; abs(SensorValue[LeftEncoder])<= pulses || abs(SensorValue[RightEncoder]<=pulses); c++){  //loop to run until encoders are equal or g
    //(run until one revolution completed)
    errorLeft = pulses - SensorValue[LeftEncoder];      errorRight = pulses - SensorValue[RightEncoder];
    outputLeft = KP*errorLeft + bias;     outputRight = KP*errorRight + bias;
    priorErrorLeft = errorLeft;     priorErrorRight = errorRight;
    if(outputLeft > maxSpeed) outputLeft = maxSpeed; else if (outputLeft < -(maxSpeed)) outputLeft = -(maxSpeed);
    if(outputRight > maxSpeed) outputRight = maxSpeed; else if (outputRight < -(maxSpeed)) outputRight = -(maxSpeed);
    switch(direction){
    case(1):
      motor[LeftMotor] = outputLeft;
      motor[RightMotor] = outputRight;
      break;
    case(2):
      motor[LeftMotor] = -(outputLeft);
      motor[RightMotor] = -(outputRight);
      break;
    case(3):
      motor[LeftMotor] = -(outputLeft);
      motor[RightMotor] = outputRight;
      break;
    case(4):
      motor[LeftMotor] = outputLeft;
      motor[RightMotor] = -(outputRight);
      break;
    }
    wait1Msec(iterationTime);
    if (errorLeft<0 || errorRight<0)break;
  }
  SensorValue[LeftEncoder] = 0;
  SensorValue[RightEncoder] = 0;
}

task main(){
  clearDebugStream();
  move(1,50,360); //Move forwards at speed of 50 for 360 pulses
  writeDebugStreamLine("Finished first step");
  move(4,50,219); //Rotate 90° right at speed of 50
  writeDebugStreamLine("Finished second step");
  move(3,50,438); //Rotate 180° left at speed of 50
  writeDebugStreamLine("Finished third step");
  move(2,50,360); //Move backwards at speed of 50 for 360 pulses
  writeDebugStreamLine("Finished!");
}
```