```c
/*
7/21/2011 Ether

Here's a simple C function CWYdeg(x,y)
which returns an angle from 0 to 360 degrees
measured clockwise from the +Y Cartesian axis
when given Cartesian coordinates (x,y)

Using single-precision floats, the accuracy
is better than +/-0.17 degree over the
entire 0 to 360 range.

Using a similar technique, it is straightforward
to create a function which returns angles either
clockwise or counterclockwise, from the +X or -X
or +Y or -Y axis, in the range 0 to 360
or -180 to +180

The coefficients for the "helper function" f()
were determined using linear algebra techniques
to minimize the maximum absolute error.
*/


#include <stdio.h>


float CWYdeg(float x, float y){
float f(float t);
if(x>=0)if(y>=0)if(y>x)return f(x/y);else
if(x==0)return 0; else return 90-f(y/x);else
if(-y<=x)return 90+f(-y/x);else return 180-f(-x/y);else
if(y<=0)if(y<=x)return 180+f(x/y);else return 270-f(y/x);else
if(y<=-x)return 270+f(-y/x);else return 360-f(-x/y);}


float f(float t){
const float c=  -0.0802884041;
const float b=  60.81576;
const float a= -15.574181;
return  t*(a*t+b)+c;}


// here's code to test the function:

void test(float x, float y){printf("x=%f y=%f CWYdeg(x,y)=%f\n",x,y,CWYdeg(x,y));}

void main(void){
test(0,0);
test(0,1);
test(1,1);
test(1,0);
test(0,-1);
test(-1,-1);
test(-1,0);
test(-1,1);
test(-0.0001,1);
}
```