/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* HeeChan Kang

\* CSC 431 - AI Robotics

\* Assignment Two - Arm Localization

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int checkAngle (double alpha, double beta);

void main() {

double alpha, beta, x, y, x\_temp;

double base = 100.0, l1 = 80.0, l2 = 80.0, limit = 7\*M\_PI/8;

int i, j;

int row = 80, column = 80;

for (i = 0; i < row; i++) {

beta = limit - i\*(2\*limit)/row;

for (j = 0; j < column; j++) {

alpha = -limit + j\*(2\*limit)/column;

printf("%d", checkAngle(alpha, beta));

}

printf("\n");

}

}

**/\* Method that returns 0 if unreachable and 1 if reachable. \*/**

int checkAngle (double alpha, double beta) {

double base = 100.0, l1 = 80.0, l2 = 80.0;

double x, y, x\_temp;

x = -l2\*sin(alpha + beta) - l1\*sin(alpha);

y = l2\*cos(alpha + beta) + l1\*cos(alpha) + base;

**/\* Check if arm goes underground. \*/**

if (y < 0) { return 0; }

**/\* Check if arm crosses its base. \*/**

if (y < base) {

/\* Compute location of "elbow". \*/

x\_temp = l1\*cos(alpha + M\_PI/2);

**/\* Value is negative if "elbow" and "wrist" is on the opposite side. \*/**

if (x \* x\_temp < 0) { return 0; }

}

**/\* Otherwise, we are good to go. \*/**

return 1;

}

