/\*

PROGRAM DESCRIPTION : THIS PROGRAM PERFORMS THE INVERSE KINEMATIC

ANALYSIS OF A FOUR AXIS ADEPT-1 SCARA ROBOT

\*/

#include<stdio.h>

#include<conio.h>

#include<math.h>

void display\_q(double\*,int );

void main()

{

double arm[4][4],a[4],d[4],w[6],q[4];

double A,B;

int i,j,k;

char ch;

clrscr();

//display the title of the program

printf("\n\t ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ» ");

printf("\n\t º INVERSE ARM KINEMATICS OF A FOUR AXIS ADEPT-1 SCARA ROBOT º ");

printf("\n\t ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼ ");

//Input the Joint Distance Vector

printf("\n\n\t\t\t Enter the Joint Distances Vector(in mm)\n ");

for(i=0;i<4;i++)

if(i!=1)

{

printf("\t\t\t\t Enter d%d : ",i+1);

scanf("%lf",&d[i]);

}

d[1] = 0;

printf("\t\t\t\t (d2 = 0)\n");

//Input the Link Length Vector

printf("\n\t\t\t Enter the Link Length Vector(in mm)\n ");

for(i=0;i<2;i++)

{

printf("\t\t\t\t Enter a%d : ",i+1);

scanf("%lf",&a[i]);

}

a[2] = a[3] = 0;

printf("\t\t\t\t (a2 = a3 = 0)\n");

getch();

clrscr();

//Input the Arm Matrix

printf("\n\t\t Enter the Arm Matrix :\n ");

gotoxy(13,5);

printf("Ú ¿");

for(i=0;i<7;i++)

{

gotoxy(13,6+i);

printf("³");

gotoxy(54,6+i);

printf("³");

}

gotoxy(13,6+i);

printf("À Ù");

for(i=0;i<4;i++)

for(j=0;j<4;j++)

{

gotoxy(15+10\*j,6+2\*i);

scanf("%lf",&arm[i][j]);

}

//Input the Tool Configuration Vector

printf("\n\n\t\t Enter Tool Configuration Vector - ");

for(i=0;i<4;i++)

w[i] = arm[i][3];

w[3] = w[4] = 0;

printf("\n\t\t\t Enter w6 : ");

scanf("%lf",&w[5]);

printf("\t ( w1 = %2.2f , w2 = %2.2f , w3 = %2.2f ",w[0],w[1],w[2]);

printf(", w4 = w5 = 0 ) ");

getch();

clrscr();

//Display the Arm Matrix

printf("\n\n\t\t\t THE ARM MATRIX \n");

printf("\t\t\t ÍÍÍÍÍÍÍÍÍÍÍÍÍÍ");

printf("\n\t\t Ú ¿");

for(i=0;i<4;i++)

{

printf("\n\t\t ³");

k=0;

for(j=0;j<4;j++)

{

if(arm[i][j] < 0 )

{

k = k + 6;

gotoxy(15+k,6+i);

k++;

}

else

{

k = k + 7;

gotoxy(15+k,6+i);

}

if(arm[i][j])

if(arm[i][j] -(int)arm[i][j] == 0)

printf("%0.0f ",arm[i][j]);

else

printf("%0.2f ",arm[i][j]);

else

printf("%0.0f ",arm[i][j]);

}

gotoxy(15+k+7,6+i);

printf("³");

}

printf("\n\t\t À Ù");

gotoxy(12,7);

printf(" 4");

gotoxy(12,8);

printf("T = ");

gotoxy(12,9);

printf(" 0");

//Display the Tool Configuration Vector

gotoxy(12,12);

printf("\n\t\t THE TOOL CONFIGURATION VECTOR ");

printf("\n\t\t ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ ");

printf("\n\t\t\t Ú ¿");

for(i=0;i<6;i++)

{

printf("\n\t\t\t ³ %3.2f ",w[i]);

gotoxy(41,16+i);

printf("³");

}

printf("\n\t\t\t À Ù");

gotoxy(25,18);

printf("w(q) = ");

//Compute the Joint Angle Vectors i.e perform IK

//Compute one set of solution using positive value of q1

q[1] = acos((w[0]\*w[0]+w[1]\*w[1]-a[0]\*a[0]-a[1]\*a[1])/(2\*a[0]\*a[1]));

A = a[1]\*sin(q[1]);

B = a[0] + a[1]\*cos(q[1]);

q[0] = atan((A\*w[0] + B\*w[1])/(B\*w[0]-A\*w[1]));

q[2] = d[0] - d[3] -w[2];

q[3] = 3.14159\*log(-w[5]);

getch();

clrscr();

printf("\n\n\n\n\n\t\t\t JOINT ANGLE VECTOR \n");

printf("\t\t\t ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ \n\n");

//Display the Joint Angle Vector(1st solution)

gotoxy(13,12);

printf("q(w) =");

gotoxy(34,12);

printf(" OR ");

display\_q(q,11);

//Compute second set of solution using negative value of q2

q[1] = -acos((w[0]\*w[0]+w[1]\*w[1]-a[0]\*a[0]-a[1]\*a[1])/(2\*a[0]\*a[1]));

A = a[1]\*sin(q[1]);

B = a[0] + a[1]\*cos(q[1]);

q[0] = atan((A\*w[0] + B\*w[1])/(B\*w[0]-A\*w[1]));

//display the Joint Angle Vectors(2nd solution)

display\_q(q,33);

getch();

}

//Module for displaying the Joint Angle Vector

void display\_q(double\* q,int x)

{

int i,F = 180/3.14;

gotoxy(9+x,9);

printf("Ú ¿");

gotoxy(10+x,10);

for(i=0;i<4;i++)

{

gotoxy(9+x,10+i);

printf("³ ³");

if(q[i]<0)

gotoxy(11+x,10+i);

else

gotoxy(12+x,10+i);

if(i!=2)

printf("%3.2f ",q[i]\*F);

}

gotoxy(9+x,10+i);

printf("À Ù");

gotoxy(10+x,12);

printf(" %3.2f",q[2]);

}











