/\*

PROGRAM DESCRIPTION : THIS PROGRAM PERFORMS THE DIRECT KINEMATIC

ANALYSIS OF A FOUR AXIS ADEPT-1 SCARA ROBOT

\*/

# include<stdio.h>

# include<conio.h>

# include<graphics.h>

# include<math.h>

# include<iostream.h>

# include<iomanip.h>

class HCTM

{

public :

double matrix[4][4];

double d,theta,a,alfa;

HCTM();

HCTM(double d,double theta,double a,double alfa);

HCTM mul(HCTM h);

void mul(double h[],double result[]);

void displayMatrix();

void displayrot();

void displayPosition();

};

HCTM :: HCTM(){

d = 1;theta=0;a=1;alfa=0;

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

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

if (i==j matrix[i][j] = 1;

else matrix[i][j] = 0;

}

HCTM :: HCTM(double d,double theta,double a,double alfa){

this->d = d;

this->theta = theta;

this->a = a;

this->alfa = alfa;

double ctheta,stheta,calfa,salfa;

ctheta = cos(theta);

if( theta < 0 )

stheta = -sin(-theta);

else

stheta = sin(theta);

calfa = cos(alfa);

if( alfa < 0 )

salfa = -sin(-alfa);

else

salfa = sin(alfa);

matrix[0][0] = ctheta;

matrix[0][1] = -stheta\*calfa;

matrix[0][2] = stheta\*salfa;

matrix[0][3] = a\*ctheta;

matrix[1][0] = stheta;

matrix[1][1] = ctheta\*calfa;

matrix[1][2] = -ctheta\*salfa;

matrix[1][3] = a\*stheta;

matrix[2][0] = 0;

matrix[2][1] = salfa;

matrix[2][2] = calfa;

matrix[2][3] = d;

matrix[3][0] = 0;

matrix[3][1] = 0;

matrix[3][2] = 0;

matrix[3][3] = 1;

}

HCTM HCTM::mul(HCTM h)

{

HCTM result;

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

{

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

{

double sum = 0;

for(int k=0;k<4;k++)

{

sum+=matrix[i][k]\*h.matrix[k][j];

}

result.matrix[i][j] = sum;

}

}

return result;

}

void HCTM :: mul(double h[],double result[])

{

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

{

double sum=0;

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

{

sum += matrix[i][j]\*h[j];

}

result[i] = sum;

}

}

void HCTM :: displayMatrix()

{

cout<<"\n";

cout<<"\t"<<'Ú'<<"\t\t\t\t\t\t "<<'¿';

cout<<"\n";

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

{

cout<<"\t"<<'³';

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

{

if ( (matrix[i][j] \* matrix[i][j] ) <= 0.00005)

matrix[i][j] = 0;

printf(" %10.3f ", matrix[i][j] );

}

cout<<" "<<'³';

cout<<"\n";

}

cout<<"\t"<<'À'<<"\t\t\t\t\t\t "<<'Ù';

}

void HCTM ::displayrot()

{

cout<<"\n";

cout<<"\t"<<'Ú'<<"\t\t\t\t\t"<<'¿';

cout<<"\n";

for(int i=0;i<3;i++)

{

cout<<"\t"<<'³';

for(int j=0;j<3;j++)

{

if ( (matrix[i][j] \* matrix[i][j] ) <= 0.00005)

matrix[i][j] = 0;

printf(" %10.3f ", matrix[i][j] );

}

cout<<" "<<'³';

cout<<"\n";

}

cout<<"\t"<<'À'<<"\t\t\t\t\t"<<'Ù';

}

void HCTM :: displayPosition()

{

cout<<"\n\n\n";

cout<<"\t"<<'Ú'<<"\t\t\t\t\t\t "<<"\t¿T";

cout<<"\n";

cout<<"\t"<<'³';

for(int j=0;j<3;j++)

printf("\t %10.3f ",matrix[j][3]);

cout<<"\t³";

cout<<"\n\t"<<'À'<<"\t\t\t\t\t\t "<<"\tÙ";

cout<<"\n";

}

int main(void)

{

clrscr();

double theta1,theta2,theta3,theta4,theta5;

double d1,d3,d4,d5,a1,a2,a3,a4;

double factor = M\_PI/180; // to convert from degree to radians

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

printf("\n\t º DIRECT KINEMATIC ANALYSIS OF A FOUR AXIS ADEPT-1 SCARA ROBOT º ");

printf("\n\t ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼ ");

cout<<"\n\t\tGeometric link parameters \n";

cout<<"\t\t\tEnter a1 : ";

cin>>a1;

cout<<"\t\t\tEnter a2 : ";

cin>>a2;

cout<<"\t\t\tEnter d1 : ";

cin>>d1;

cout<<"\t\t\tEnter d4 : ";

cin>>d4;

cout<<"\n\t\tJoint angle vector \n";

cout<<"\t\t\tEnter d3 : ";

cin>>d3;

int tempd3=d3;

cout<<"\t\t\tEnter @1 : ";

cin>>theta1;

cout<<"\t\t\tEnter @2 : ";

cin>>theta2;

cout<<"\t\t\tEnter @4 : ";

cin>>theta4;

clrscr();

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

printf("\n\t º DIRECT KINEMATIC ANALYSIS OF A FOUR AXIS ADEPT-1 SCARA ROBOT º ");

printf("\n\t ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼ ");

cout<<"\n\n\n\t\t\t The Intermediate Matrices . . . ";

cout<<"\n\t\t\t\t 1"<<"\n\t\t\t\tT ="<<"\n\t\t\t\t 0";

HCTM t10(d1,theta1\*factor,a1,M\_PI);

t10.displayMatrix();

getch();

cout<<"\n\n\n\t\t\t\t 2"<<"\n\t\t\t\tT ="<<"\n\t\t\t\t 1";

HCTM t21(0,theta2\*factor,a2,0);

t21.displayMatrix();

getch();

clrscr();

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

printf("\n\t º DIRECT KINEMATIC ANALYSIS OF A FOUR AXIS ADEPT-1 SCARA ROBOT º ");

printf("\n\t ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼ ");

cout<<"\n\n\n\t\t\t The Intermediate Matrices . . . ";

cout<<"\n\t\t\t\t 3"<<"\n\t\t\t\tT ="<<"\n\t\t\t\t 2";

HCTM t32(d3,0,0,0);

t32.displayMatrix();

getch();

cout<<"\n\n\n\t\t\t\t 4"<<"\n\t\t\t\tT ="<<"\n\t\t\t\t 3";

HCTM t43(d4,theta4\*factor,0,0);

t43.displayMatrix();

getch();

clrscr();

HCTM t40,t30,t20;

t20 = t10.mul(t21);

t30 = t20.mul(t32);

t40 = t30.mul(t43);

clrscr();

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

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

printf("\n\t ÈÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ¼ ");

cout<<"\n\t\t The Final HCTM Is Given As Follows :";

cout<<"\n\t\t\t\t 4"<<"\n\t\t\t\tT ="<<"\n\t\t\t\t 0";

t40.displayMatrix();

cout<<"\n\n\n\t\t\t Rotation Matrix . . . ";

t40.displayrot();

cout<<"\n\n\n\t\t The Physical Co-ordinates Are . . . ";

t40.displayPosition();

getch();

return 0;

}













