//final code

#include<math.h>

#include<Servo.h>

Servo my;

#define r2 0.9

void zd()

{

my.write(180);

}

void zu()

{

my.write(100);

}

void num0()

{

zd();

ymoveb(4);

xmovef(4);

ymovef(4);

xmoveb(4);

southeast(4\*r2);

zu();

ymovef(4);

}

void num1()

{

ymoveb(2);

zd();

northeast(2\*r2);

ymoveb(4);

zu();

northeast(2\*r2);

ymovef(2);

}

void num2()

{

zd();

xmovef(4);

ymoveb(2);

xmoveb(4);

ymoveb(2);

xmovef(4);

zu();

ymovef(4);

}

void num3()

{

zd();

xmovef(4);

ymoveb(2);

xmoveb(4);

zu();

ymoveb(2);

zd();

xmovef(4);

ymovef(4);

zu();

}

void num4()

{

xmovef(4);

zd();

ymoveb(4);

zu();

ymovef(2);

zd();

xmoveb(2);

northeast(2\*r2);

zu();

}

void num5()

{

zd();

ymoveb(2);

xmovef(4);

ymoveb(2);

xmoveb(4);

zu();

ymovef(4);

zd();

xmovef(4);

zu();

}

void num6()

{

zd();

ymoveb(4);

xmovef(4);

ymovef(2);

xmoveb(4);

zu();

northeast(2\*r2);

xmovef(2);

}

void num7()

{

zd();

xmovef(4);

southwest(4\*r2);

zu();

northeast(4\*r2);

}

void num8()

{

alpO();

ymoveb(2);

zd();

xmoveb(4);

zu();

northeast(2\*r2);

xmovef(2);

}

void num9()

{

zd();

ymoveb(2);

xmovef(4);

ymovef(2);

ymoveb(4);

xmoveb(4);

zu();

ymovef(4);

zd();

xmovef(4);

zu();

}

void alpA()

{

ymoveb(4);

zd();

ymovef(2);

northeast(2\*r2);

southeast(2\*r2);

ymoveb(2);

zu();

ymovef(2);

zd();

xmoveb(4);

zu();

ymovef(2);

xmovef(4);

}

void alpB(){

zd();

ymoveb(4);

xmovef(4);

northwest(2\*r2);

northeast(2\*r2);

xmoveb(4);

zu();

xmovef(4);

}

void alpC(){

zd();

ymoveb(4);

xmovef(4);

zu();

ymovef(4);

zd();

xmoveb(4);

zu();

xmovef(4);

}

void alpD(){

zd();

ymoveb(4);

xmovef(2);

northeast(2\*r2);

northwest(2\*r2);

xmoveb(2);

zu();

xmovef(4);

}

void alpE(){

zd();

xmovef(4);

xmoveb(4);

ymoveb(2);

xmovef(2);

xmoveb(2);

ymoveb(2);

xmovef(4);

zu();

ymovef(4);

}

void alpF(){

zd();

xmovef(4);

xmoveb(4);

ymoveb(2);

xmovef(2);

xmoveb(2);

ymoveb(2);

zu();

xmovef(4);

ymovef(4);

}

void alpG(){

zd();

xmovef(4);

xmoveb(4);

ymoveb(4);

xmovef(4);

ymovef(2);

xmoveb(2);

xmovef(2);

zu();

ymovef(2);

}

void alpH(){

zd();

ymoveb(4);

ymovef(2);

xmovef(4);

ymoveb(2);

ymovef(4);

zu();

}

void alpI(){

zd();

xmovef(4);

xmoveb(2);

ymoveb(4);

xmoveb(2);

xmovef(4);

zu();

ymovef(4);

}

void alpJ(){

zd();

xmovef(4);

xmoveb(2);

ymoveb(4);

xmoveb(2);

ymovef(2);

zu();

xmovef(4);

ymovef(2);

}

void alpK()

{

zd();

ymoveb(4);

ymovef(2);

northeast(2\*r2);

southwest(2\*r2);

southeast(2\*r2);

zu();

ymovef(4);

}

void alpZ()

{

zd();

xmovef(4);

southwest(4\*r2);

xmovef(4);

zu();

ymovef(4);

}

void alpY()

{

zd();

southeast(2\*r2);

northeast(2\*r2);

zu();

ymoveb(4);

xmoveb(2);

zd();

ymovef(2);

zu();

ymovef(2);

xmovef(2);

}

void alpX()

{

zd();

southeast(4\*r2);

zu();

xmoveb(4);

zd();

northeast(4\*r2);

zu();

}

void alpW()

{

zd();

ymoveb(4);

northeast(2\*r2);

southeast(2\*r2);

ymovef(4);

zu();

}

void alpV()

{

zd();

ymoveb(2);

southeast(2\*r2);

northeast(2\*r2);

ymovef(2);

zu();

}

void alpU()

{

zd();

ymoveb(4);

xmovef(4);

ymovef(4);

zu();

}

void alpT()

{

zd();

xmovef(4);

zu();

xmoveb(2);

zd();

ymoveb(4);

zu();

xmovef(2);

ymovef(4);

}

void alpS()

{

ymoveb(4);

zd();

xmovef(4);

ymovef(2);

xmoveb(4);

ymovef(2);

xmovef(4);

zu();

}

void alpR()

{

zd();

ymoveb(4);

zu();

xmovef(2);

zd();

northwest(2\*r2);

xmovef(2);

ymovef(2);

xmoveb(2);

zu();

xmovef(4);

}

void alpQ()

{

alpO();

southwest(2\*r2);

zd();

southeast(4\*r2);

zu();

ymovef(5);

}

void alpP()

{

zd();

ymoveb(4);

zu();

ymovef(2);

zd();

xmovef(4);

ymovef(2);

xmoveb(4);

zu();

xmovef(4);

}

void alpO()

{

zd();

ymoveb(4);

xmovef(4);

ymovef(4);

xmoveb(4);

zu();

xmovef(4);

}

void alpN()

{

ymoveb(4);

zd();

ymovef(4);

southeast(4\*r2);

ymovef(4);

zu();

}

void alpM()

{

ymoveb(4);

zd();

ymovef(4);

southeast(2\*r2);

northeast(2\*r2);

ymoveb(4);

zu();

ymovef(4);

}

void alpL()

{

zd();

ymoveb(4);

xmovef(4);

zu();

ymovef(4);

}

void moveats(int delay1, int t)

{

float ta;

ta=tan(t);

}

int fact(int n)

{

int i;

int sum=1;

for(i=n;i>0;i--)

{

sum=sum\*i;

}

return sum;

}

void xmovef(int delay1)

{

int i;

for(i=0;i<delay1\*1000/4;i++)

{

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,LOW);

digitalWrite(6,HIGH);

delay(1);

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,HIGH);

digitalWrite(6,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(4,HIGH);

digitalWrite(5,LOW);

digitalWrite(6,LOW);

delay(1);

digitalWrite(3,HIGH);

digitalWrite(4,LOW);

digitalWrite(5,LOW);

digitalWrite(6,LOW);

delay(1);

}

}

void xmoveb(int delay1)

{

int i;

for(i=0;i<delay1\*1000/4;i++)

{

digitalWrite(3,HIGH);

digitalWrite(4,LOW);

digitalWrite(5,LOW);

digitalWrite(6,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(4,HIGH);

digitalWrite(5,LOW);

digitalWrite(6,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,HIGH);

digitalWrite(6,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,LOW);

digitalWrite(6,HIGH);

delay(1);

}

}

void ymovef(int delay1)

{

int i;

for(i=0;i<delay1\*1000/4;i++)

{

digitalWrite(7,LOW);

digitalWrite(8,LOW);

digitalWrite(9,LOW);

digitalWrite(10,HIGH);

delay(1);

digitalWrite(7,LOW);

digitalWrite(8,LOW);

digitalWrite(9,HIGH);

digitalWrite(10,LOW);

delay(1);

digitalWrite(7,LOW);

digitalWrite(8,HIGH);

digitalWrite(9,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(7,HIGH);

digitalWrite(8,LOW);

digitalWrite(9,LOW);

digitalWrite(10,LOW);

delay(1);

}

}

void ymoveb(int delay1)

{

int i;

for(i=0;i<delay1\*1000/4;i++)

{

digitalWrite(7,HIGH);

digitalWrite(8,LOW);

digitalWrite(9,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(7,LOW);

digitalWrite(8,HIGH);

digitalWrite(9,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(7,LOW);

digitalWrite(8,LOW);

digitalWrite(9,HIGH);

digitalWrite(10,LOW);

delay(1);

digitalWrite(7,LOW);

digitalWrite(8,LOW);

digitalWrite(9,LOW);

digitalWrite(10,HIGH);

delay(1);

}

}

void northwest(float delay1)

{

float i;

for(i=0;i<delay1\*1000/4;i++)

{

digitalWrite(3,HIGH);

digitalWrite(7,LOW);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,LOW);

digitalWrite(10,HIGH);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,HIGH);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,HIGH);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,LOW);

digitalWrite(8,HIGH);

digitalWrite(5,HIGH);

digitalWrite(9,LOW);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,HIGH);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,HIGH);

digitalWrite(10,LOW);

delay(1);

}

}

void southeast(float delay1)

{

float i;

for(i=0;i<delay1\*1000/4;i++)

{

digitalWrite(3,LOW);

digitalWrite(7,HIGH);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,HIGH);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,LOW);

digitalWrite(8,HIGH);

digitalWrite(5,HIGH);

digitalWrite(9,LOW);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,HIGH);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,HIGH);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,HIGH);

digitalWrite(7,LOW);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,LOW);

digitalWrite(10,HIGH);

delay(1);

}

}

void southwest(float delay1)

{

float i;

for(i=0;i<delay1\*1000/4;i++)

{

digitalWrite(3,HIGH);

digitalWrite(7,HIGH);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,HIGH);

digitalWrite(8,HIGH);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,HIGH);

digitalWrite(9,HIGH);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,HIGH);

digitalWrite(10,HIGH);

delay(1);

}

}

void northeast(float delay1)

{

float i;

for(i=0;i<delay1\*1000/4;i++)

{

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,HIGH);

digitalWrite(10,HIGH);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,HIGH);

digitalWrite(9,HIGH);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,LOW);

digitalWrite(7,LOW);

digitalWrite(4,HIGH);

digitalWrite(8,HIGH);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

digitalWrite(3,HIGH);

digitalWrite(7,HIGH);

digitalWrite(4,LOW);

digitalWrite(8,LOW);

digitalWrite(5,LOW);

digitalWrite(9,LOW);

digitalWrite(6,LOW);

digitalWrite(10,LOW);

delay(1);

}

}

void space()

{

xmovef(1);

}

void setup()

{

int i;

Serial1.begin(9600);

for(i=3;i<11;i++)

{

pinMode(i,INPUT);

}

my.attach(11);

}

void loop()

{

char ch;

ch=Serial1.read();

if(ch=='a')

my.write(100);

else if(ch=='b')

my.write(180);

else if(ch=='1')

xmoveb(1);

else if(ch=='2')

xmovef(1);

else if(ch=='3')

ymoveb(1);

else if(ch=='4')

ymovef(1);

else if(ch=='5')

{

String message = "KLE";

// Iterate through each character in the message

for (int i = 0; i < message.length(); i++) {

char letter = toupper(message.charAt(i));

// Call the corresponding function based on the letter or number

switch (letter) {

case 'A':

alpA();

break;

case 'B':

alpB();

break;

case 'C':

alpC();

break;

case 'D':

alpD();

break;

case 'E':

alpE();

break;

case 'F':

alpF();

break;

case 'G':

alpG();

break;

case 'H':

alpH();

break;

case 'I':

alpI();

break;

case 'J':

alpJ();

break;

case 'K':

alpK();

break;

case 'L':

alpL();

break;

case 'M':

alpM();

break;

case 'N':

alpN();

break;

case 'O':

alpO();

break;

case 'P':

alpP();

break;

case 'Q':

alpQ();

break;

case 'R':

alpR();

break;

case 'S':

alpS();

break;

case 'T':

alpT();

break;

case 'U':

alpU();

break;

case 'V':

alpV();

break;

case 'W':

alpW();

break;

case 'X':

alpX();

break;

case 'Y':

alpY();

break;

case 'Z':

alpZ();

break;

case '0':

num0();

break;

case '1':

num1();

break;

case '2':

num2();

break;

case '3':

num3();

break;

case '4':

num4();

break;

case '5':

num5();

break;

case '6':

num6();

break;

case '7':

num7();

break;

case '8':

num8();

break;

case '9':

num9();

break;

case ' ':

space();

break;

default:

// Handle any other characters or actions if needed

break;

}

//delay(1000); // Delay between each letter drawing

space(); // Call the space function after each letter

// Call ymoveb(6) and xmoveb(25) after every 5 characters

if ((i + 1) % 5 == 0) {

ymoveb(7);

xmoveb(25);

}

}

}

else if(ch=='6')

{

//button2- square spiral

//start at center position

zd();

ymovef(2);

xmoveb(2);

ymoveb(3);

xmovef(3);

ymovef(4);

xmoveb(4);

ymoveb(5);

xmovef(5);

ymovef(6);

xmoveb(6);

ymoveb(7);

xmovef(7);

zu();

}

else if(ch=='7')

{

//button 3-random pattern

//start at home

zd();

xmovef(4);

southeast(4\*r2);

xmovef(4);

zu();

ymovef(4);

zd();

xmoveb(4);

southwest(4\*r2);

xmoveb(4);

zu();

}

else if(ch=='8')

{

//button 4- box with slant lines

zd();

alpO();

zd();

southwest(4\*r2);

zu();

xmovef(4);

zd();

northwest(4\*r2);

zu();

xmovef(3);

zd();

southwest(3\*r2);

ymovef(1);

northeast(2\*r2);

xmovef(2);

ymoveb(1);

southwest(3\*r2);

xmovef(1);

northeast(2\*r2);

zu();

}

else if(ch=='9')

{

//button 5- simple square

//give decent space

int sq=4;

xmovef(sq);

ymoveb(sq);

xmoveb(sq);

ymovef(sq);

  }

}