Object description of paddles, ball etc. will be defined.
This will eliminate most of the arbit variable names.

DEFINE VARIABLES:

float vX - X velocity of ball
float vY - Y velocity of ball
float v1 - Y velocity of paddle 1
float v2 - Y velocity of paddle 2
float x - X coordinate of ball
float y - Y coordinate of ball
float y1 - y coordinate of center of paddle 1
float y2 - y coordinate of center of paddle 2
float l1 - length of paddle 1
float l2 - length of paddle 2
int h1 - health of paddle 1
int h2 - health of paddle 2
power1[3] - array of powers with the player 1
power2[3] - array of powers with the player 2
char[5] input1 - stores last five inputs for paddle 1
char[5] input2 - stores last five inputs for paddle2
const int HMAX - max health of paddle
const float P - breadth of paddle
const float vM - minimum velocity of paddle
const float T - time interval after which window is updated
const float L - length of arena
const float B - breadth of arena
arena is rectangular

METHODS:

1. method ballMotion()

```
1.-1 START
1.0 if(h1<=0 || h2<=0)
  1.0.1 goto 1.10
1.1 print (x,y)
1.2 x += vX*T
1.3 if(x<0)            // collision with wall behind paddle 1
  1.3.1 x*= -1
  1.3.2 vX*= -1
  1.3.3 h1-=.1*HMAX
1.4 if(x>L)            // collision with wall behind paddle 2
```

```
    1.4.1 x= 2*L-x
    1.4.2 vX*= -1
    1.4.3h2-=HMAX
1.5 y+=Vy*T
1.6 if(y<0)
    1.6.1 y*= -1
    1.6.2 vY*= -1
    1.6.3 if(x>x1&&x<x2)
         // ball hits area of special power dispensation
       1.6.3.1 assign random power to power array if array has empty spaces
1.7 if(y>B)
    1.7.1 y=2*B-y
    1.7.2 vY*=-1
    1.7.3 if(x>x1&&x<x2)
         // ball hits area of special power dispensation
       1.7.3.1 assign random power to power array if array has empty spaces
1.8 if( x<=P && (y<y1+l1/2 && y>y1-l1/2) )
    1.8.1 execute paddleReflect(1)
1.9 if( x>=L-P && (y<y2+l2/2 && y>y2-l2/2))
    1.9.1 execute paddleReflect(2)
1.10 if(h1==0)
    1.10.1 print PLAYER 2 WINS
    1.10.2 goto STOP
1.11 print PLAYER 1 WINS
1.12 STOP
```

```
2. method paddleReflect(int paddleNumber)

2.0 START
2.1 if(paddleNumber==1)
    2.1.1 int u1 = -1*(vY-v1)
    2.1.2 vX*=-1
    2.1.2 vY= u1 + v1
    2.1.3 y+=v1*T
2.2 if(paddleNumber==2)
    2.2.1 int u1 = -1*(vY-v2)
    2.2.2 vX*=-1
    2.2.2 vY= u1 + v2
    2.2.3y+= v2*T
2.3 STOP
```

3. method paddleMotion(int paddleNumber)

```
3.0 START
3.1 if(paddleNumber==1)
  3.1.1 input char in
  3.1.2 if(in=='w'||in=='s')
    3.1.2.1 int ctr=0
    3.1.2.2 if(input1[ctr]==in)
      3.1.2.2.1 ctr++
      3.1.2.2.2 if(ctr<5)
        3.1.2.2.2.1 goto 3.1.2.2
    3.1.2.3 v1= (1+.2*c)*vM
    3.1.2.4 if(in=='w')
      3.1.2.4.1 v1*=-1
    3.1.2.5 if(! ((y1==l/2 && v1<0)||(y1==B-l/2 && v1>0) )
      3.1.2.5.1 y1+= v1*T
    3.1.2.6 int i=0
    3.1.2.7 if(i<3)
      3.1.2.7 input1[i+1]=input1[i]
    3.1.2.8 input[0]=in

3.2 similarly for paddle 2
3.3 STOP
```

4. method 4 to do with execution of special powers