```
#define          F_CPU   10000000UL

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>
#include <math.h>
/*
-----------------------------------------------
Inställningar
-----------------------------------------------

*/


void init(){
        DDRB = 0b11111111;
        DDRD = 0b00111111;
        DDRC = 0b11111110;

        PCICR |= 1<<PCIE2;

        PCMSK2 |= (1<<PCINT16);

}

void E_low(){
        PORTD &= ~(0b00000100);
}

void E_high(){

        PORTD |= 0b00000100;
}

void rw_high(){
        PORTD |= 0b00001000;

}

void rw_low(){
        PORTD &= ~(0b00001000);
```

```
}

void Di_high(){
	PORTD |= 0b00100000;
}

void Di_low(){
	PORTD &= ~(0b00100000);
}

void reset_off(){
	PORTD |= (0b00010000);
}

void reset_on(){

	PORTD &= ~(0b00010000);
}


void chip_1(){

	PORTD &= ~(0b00000010);
	PORTD |= (0b00000001);

}


void chip_2(){

	PORTD &= ~(0b00000001);
	PORTD |= (0b00000010);
}




/*
---------------------------------------------------------------------
Används för att draw
---------------------------------------------------------------------

*/
```

```
void SetYadrchip1(char y){
	chip_2();
	rw_low();
	Di_low();
	reset_off();
	PORTB = (0b01000000 | y);
	E_high();
	E_low();
}

void SetXadrchip1(char x){

	rw_low();
	Di_low();
	chip_2();
	reset_off();
	PORTB = (0b10111000 | (x/8));
	E_high();
	E_low();

}

void SetYadrchip2(char y){
	chip_1();
	rw_low();
	Di_low();
	reset_off();
	PORTB = (0b01000000 | y);
	E_high();
	E_low();
}

void SetXadrchip2(char x){

	rw_low();
	Di_low();
	chip_1();
	reset_off();
	PORTB = (0b10111000 | (x/8));
	E_high();
	E_low();

}
```

```
void DisplayOn(){
	PORTB = 0b00111111; // Display on PB0-PB4 = 1;
	rw_low();
	Di_low();
	reset_off();
	chip_1();
	E_high();
	E_low();
	chip_2();
	E_high();
	E_low();


}



void Displayoff(){
	PORTB |= 0b00111110; // Display on PB1-PB4 = 1 och PB0 = 0 ;
	rw_low();
	Di_low();
	reset_off();
	chip_1();
	E_high();
	E_low();
	chip_2();
	E_high();
	E_low();

}



void delay() {

	int k;
	for (k=0; k<10; k++) {
		asm volatile ("nop\n");
	}
}
```

```
void cleardisplayLeft(){
	chip_1();
	//delay();

	for(char i = 0; i<64 ; i++){
		for(char x = 0 ; x <64 ; x++){
			delay();
			SetYadrchip1(x);
			delay();
			SetXadrchip1(i);
			delay();
			rw_low();
			Di_high();
			PORTB = 0b00000000;
			reset_off();
			E_high();
			E_low();
		}
	}
}

void cleardisplayRight(){
	chip_2();
	//delay();

	for(char i = 0; i<64 ; i++){
		for(char x = 0 ; x <64 ; x++){
			delay();
			SetYadrchip2(x);
			delay();
			SetXadrchip2(i);
			delay();
			rw_low();
			Di_high();
			PORTB = 0b00000000;
			reset_off();
			E_high();
```

```
                E_low();
            }
        }
}

void startLeft(){
        chip_2();
        //delay();
        for(char i = 0; i<64 ; i++){
            for(char x = 0 ; x <64 ; x++){
                delay();
                SetYadrchip1(x);
                delay();
                SetXadrchip1(i);
                delay();
                rw_low();
                Di_high();
                PORTB = 0b11111111;
                reset_off();
                E_high();
                E_low();
            }
        }
}

void startRight(){
        chip_1();
        //delay();
        for(char i = 0; i<64 ; i++){
            for(char x = 0 ; x <64 ; x++){
                delay();
                SetYadrchip2(x);
                delay();
                SetXadrchip2(i);
                delay();
                rw_low();
                Di_high();
                PORTB = 0b11111111;
                reset_off();
                E_high();
                E_low();
            }
        }
```

```
}



/*
---------------------------------------------------------------------
Draw
---------------------------------------------------------------------
*/
char displayRight[8][64];
void Board(){
                chip_1();
                for(char i = 20 ; i<64; i++){
                        delay();
                        SetXadrchip2(i);
                        delay();
                        SetYadrchip2(0);
                        delay();
                        rw_low();
                        Di_high();
                        displayRight[(i/8)][0] = 0b11111111;
                        PORTB = 0b11111111;
                        reset_off();
                        E_high();
                        E_low();
                        delay();
                        displayRight[(i/8)][1] = 0b11111111;
                        E_high();
                        E_low();
                }


                for(char i = 20 ; i<64; i++){
                        delay();
                        SetXadrchip2(i);
                        delay();
                        SetYadrchip2(62);
                        delay();
                        rw_low();
                        Di_high();
                        displayRight[(i/8)][62] = 0b11111111;
                        PORTB = 0b11111111;
```

```
            reset_off();
            E_high();
            E_low();
            delay();
            displayRight[(i/8)][63] = 0b11111111;
            E_high();
            E_low();
        }
        char helpa = 1;

        for(char i = 2 ; i<62; i++){
            delay();
            SetXadrchip2(62);
            delay();
            SetYadrchip2(i);
            delay();
            rw_low();
            Di_high();
            helpa++;
            helpa = ((helpa -1) % 2)+1;

            if( helpa == 1){
                displayRight[62/8][i] = 0b01000000;
                PORTB = 0b01000000;
                }else{
                displayRight[62/8] [i] = 0b10000000;
                PORTB = 0b10000000;

            }
            reset_off();
            E_high();
            E_low();
        }




}
```

```
void drawRight(char x , char y){
	chip_1();
	SetYadrchip2(y);
	delay();
	SetXadrchip2(x);
	delay();
	rw_low();
	Di_high();
	displayRight[x/8][y]= displayRight[x/8][y]| (1<<(x%8));
	PORTB = displayRight[x/8][y];
	E_high();
	E_low();
	delay();
	E_high();
	E_low();
	delay();
	E_high();
	E_low();
	delay();
	E_high();
	E_low();

}

void cleargame(){
	chip_1();
	//delay();
	char helpb;
	for(char i = 2; i<62 ; i++){
		for(char x = 0 ; x <62 ; x++){
			delay();
			SetYadrchip2(i);
			delay();
			SetXadrchip2(x);
			delay();
			rw_low();
			Di_high();
			if(x == 61){
			helpb++;
			helpb = ((helpb -1) % 2)+1;
			}
			if( x == 61 && helpb == 1){
				displayRight[61/8][i] = 0b01000000;
```

```
                    PORTB = 0b01000000;
                }else if(x==61){
                    displayRight[61/8] [i] = 0b10000000;
                    PORTB = 0b10000000;

                }else{
                displayRight[x][i] = 0b00000000;
                    PORTB = 0b00000000;
                }
                reset_off();
                E_high();
                E_low();
            }

        }

}
```

```
char displayLeft[8][64];
void drawLeft(char x , char y){
    chip_2();
    SetYadrchip1(y);
    delay();
    SetXadrchip1(x);
    delay();
    rw_low();
    Di_high();
    displayLeft[x/8][y]= displayLeft[x/8][y]| (1<<(x%8));
    PORTB = displayLeft[x/8][y];
    E_high();
    E_low();
    delay();
    delay();
    displayLeft[x/8][y+1]= displayLeft[x/8][y+1]| (1<<(x%8));
    E_high();
```

```
	E_low();
	delay();


}
void drawspel( char x, char y){
	chip_1();
	SetYadrchip2(y);
	delay();
	SetXadrchip2(x);
	delay();
	rw_low();
	Di_high();
	displayRight[x/8][y]= displayRight[x/8][y]| (1<<(x%8));
	PORTB = displayRight[x/8][y];
	E_high();
	E_low();
	delay();
	delay();
	displayRight[x/8][y+1]= displayRight[x/8][y+1]| (1<<(x%8));
	PORTB = displayRight[x/8][y+1];
	E_high();
	E_low();
	delay();

}
char row;
char col;
char rotera = 1;
char sizeblock;
char offwidthL;
char offwidthR;
char width;
char hight;
char offhightUL; // Uppifrån från vänster
char offhightNL; // Nerifrån från vänster
char offhightUR;// Uppifrån från höger
char offhightNR;// Nerifrån från höger
char offwidthNedL;
char offwidthNedR;
char offwidthUppeL;
char offwidthUppeR;
```

```
void spelblock(char z){

	switch (z)
	{
		case 1:



		drawspel(row-3,col);
		drawspel(row-3,col+2);
		drawspel(row-2,col);
		drawspel(row-2,col+2);
		drawspel(row-1,col);
		drawspel(row-1,col+2);
		drawspel(row,col);
		drawspel(row,col+2);


		width = 4 ;
		hight = 4;
		offhightUL= 0; // Uppifrån från vänster
		offhightNL= 0; // Nerifrån från vänster
		offhightUR= 0;// Uppifrån från höger
		offhightNR = 0;
		offwidthNedL = 0;
		offwidthNedR = 0;
		offwidthUppeL = 0;
		offwidthUppeR = 0;



		break;

		case 2:
		if(rotera == 1 || rotera == 3){
			drawspel(row,col);
			drawspel(row-1,col);
			drawspel(row-2,col);
			drawspel(row-3,col);
			drawspel(row-4,col);
			drawspel(row-5,col);
			drawspel(row-6,col);
```

```
            drawspel(row-7,col);
            drawspel(row-8,col);
            drawspel(row-9,col);

            width = 2 ;
            hight = 10;
            offhightUL= 0; // Uppifrån från vänster
            offhightNL= 0; // Nerifrån från vänster
            offhightUR= 0;// Uppifrån från höger
            offhightNR = 0;
            offwidthNedL = 0;
            offwidthNedR = 0;
            offwidthUppeL = 0;
            offwidthUppeR = 0;

            }else{
            drawspel(row,col);
            drawspel(row-1,col);
            drawspel(row,col+2);
            drawspel(row-1,col+2);
            drawspel(row,col+4);
            drawspel(row-1,col+4);
            drawspel(row,col+6);
            drawspel(row-1,col+6);
            drawspel(row,col+8);
            drawspel(row-1,col+8);
            width = 10 ;
            hight = 2;
            offhightUL= 0; // Uppifrån från vänster
            offhightNL= 0; // Nerifrån från vänster
            offhightUR= 0;// Uppifrån från höger
            offhightNR = 0;
            offwidthNedL = 0;
            offwidthNedR = 0;
            offwidthUppeL = 0;
            offwidthUppeR = 0;


}


break;
```

```
case 3:
if(rotera == 1 || rotera == 3 ){
        drawspel(row,col);
        drawspel(row-1,col);
        drawspel(row-2,col);
        drawspel(row-3,col);

        drawspel(row-2,col+2);
        drawspel(row-3,col+2);
        drawspel(row-4,col+2);
        drawspel(row-5,col+2);


        width = 4;
        hight = 6;
        offhightUL= 0; // Uppifrån från vänster
        offhightNL= 2; // Nerifrån från vänster
        offhightUR= 2;// Uppifrån från höger
        offhightNR = 0;
        offwidthNedL = 0;
        offwidthNedR = 2;
        offwidthUppeL = 2;
        offwidthUppeR = 0;
        }else{
        drawspel(row-2, col);
        drawspel(row-3, col);
        drawspel(row-2, col+2);
        drawspel(row-3, col+2);

        drawspel(row, col+2);
        drawspel(row-1, col+2);
        drawspel(row, col+4);
        drawspel(row-1, col+4);


        width = 6 ;
        hight = 4;
        offhightUL= 2; // Uppifrån från vänster
        offhightNL= 0; // Nerifrån från vänster
        offhightUR= 0;// Uppifrån från höger
        offhightNR = 2;
        offwidthNedL = 2;
        offwidthNedR = 0;
```

```
        offwidthUppeL = 0;
        offwidthUppeR = 2;

}

break;

case 4:
if(rotera == 1 || rotera == 3){
drawspel(row,col+2);
drawspel(row-1,col+2);
drawspel(row-2,col);
drawspel(row-2,col+2);
drawspel(row-3,col);
drawspel(row-3,col+2);
drawspel(row-4,col);
drawspel(row-5,col);

        width = 4;
        hight = 6;
        offhightUL= 2; // Uppifrån från vänster
        offhightNL= 0; // Nerifrån från vänster
        offhightUR= 0;// Uppifrån från höger
        offhightNR = 2;
        offwidthNedL = 2;
        offwidthNedR = 0;
        offwidthUppeL = 0;
        offwidthUppeR = 2;
break;
}else{
                drawspel(row,col);
                drawspel(row-1,col);
                drawspel(row,col+2);
                drawspel(row-1,col+2);

        drawspel(row-2,col+2);
        drawspel(row-3,col+2);
                drawspel(row-2,col+4);
                drawspel(row-3,col+4);
                        width = 6;
                        hight = 4;
                        offhightUL= 0; // Uppifrån från vänster
                        offhightNL= 2; // Nerifrån från vänster
```

```
                        offhightUR= 2;// Uppifrån från höger
                        offhightNR = 0;
                        offwidthNedL = 0;
                        offwidthNedR = 2;
                        offwidthUppeL = 2;
                        offwidthUppeR = 0;

        break;
        case 5:
if(rotera == 1){
        drawspel(row,col);
        drawspel(row-1,col);
                drawspel(row,col+2);
                drawspel(row-1,col+2);
                        drawspel(row-2,col+2);
                        drawspel(row-3,col+2);
                                drawspel(row,col+4);
                                drawspel(row-1,col+4);
                                width = 6;
                                hight = 4;
                                offhightUL= 0;
                                offhightNL= 2;
                                offhightUR= 0;
                                offhightNR = 2;
                                offwidthNedL = 0;
                                offwidthNedR = 0;
                                offwidthUppeL = 2;
                                offwidthUppeR = 2;
                                break;
}else if(rotera == 2){

        drawspel(row,col);
        drawspel(row-1,col);
        drawspel(row-2,col);
        drawspel(row-3,col);
        drawspel(row-4,col);
        drawspel(row-5,col);
        drawspel(row-2,col+2);
        drawspel(row-3,col+2);

                width = 4;
                hight = 6;
                offhightUL= 0;
```

```
                    offhightNL= 0;
                    offhightUR= 2;
                    offhightNR = 2;
                    offwidthNedL = 0;
                    offwidthNedR = 2;
                    offwidthUppeL = 0;
                    offwidthUppeR = 2;
                    break;

        }else if(rotera  == 3){

        drawspel(row-2,col);
        drawspel(row-3,col);
        drawspel(row-2,col+2);
        drawspel(row-3,col+2);
        drawspel(row-2,col+4);
        drawspel(row-3,col+4);
        drawspel(row,col+2);
        drawspel(row-1,col+2);
                width = 6;
                hight = 4;
                offhightUL= 2;
                offhightNL= 0;
                offhightUR= 2;
                offhightNR = 0;
                offwidthNedL = 2;
                offwidthNedR = 2;
                offwidthUppeL = 0;
                offwidthUppeR = 0;

        break;

}else{

drawspel(row-2,col);
        drawspel(row-3,col);
        drawspel(row,col+2);
        drawspel(row-1,col+2);
        drawspel(row-2,col+2);
        drawspel(row-3,col+2);
        drawspel(row-4,col+2);
        drawspel(row-5,col+2);
```

```
            width = 4;
            hight = 6;
            offhightUL= 2;
            offhightNL= 2;
            offhightUR= 0;
            offhightNR = 0;
            offwidthNedL = 2;
            offwidthNedR = 0;
            offwidthUppeL = 2;
            offwidthUppeR = 0;

            break;
}
                    case 6:
                    if(rotera == 1){
                    drawspel(row,col);
                    drawspel(row-1,col);
                    drawspel(row-2,col);
                    drawspel(row-3,col);
                    drawspel(row-4,col);
                    drawspel(row-5,col);
                    drawspel(row,col +2);
                    drawspel(row-1,col +2);

                    width = 4;
                    hight = 6;
                    offhightUL= 0;
                    offhightNL= 0;
                    offhightUR= 0;
                    offhightNR = 4;
                    offwidthNedL = 0;
                    offwidthNedR = 0;
                    offwidthUppeL = 0;
                    offwidthUppeR = 2;

                    break;
                    }else if(rotera == 2){
                    drawspel(row,col);
                    drawspel(row-1,col);
                    drawspel(row-2,col);
                    drawspel(row-3,col);
                    drawspel(row-2,col+2);
                    drawspel(row-3,col+2);
```

```
drawspel(row-2,col+4);
drawspel(row-3,col+4);
width = 6;
hight = 4;
offhightUL= 0;
offhightNL= 0;
offhightUR= 2;
offhightNR = 0;
offwidthNedL = 0;
offwidthNedR = 4;
offwidthUppeL = 0;
offwidthUppeR = 0;
        break;
}else if(rotera == 3){
drawspel(row-4, col);
drawspel(row-5,col);
drawspel(row,col+2);
drawspel(row-1,col+2);
drawspel(row-2,col+2);
drawspel(row-3,col+2);
drawspel(row-4,col+2);
drawspel(row-5,col+2);

width = 4;
hight = 6;
offhightUL= 4;
offhightNL= 0;
offhightUR= 0;
offhightNR = 0;
offwidthNedL = 2;
offwidthNedR = 0;
offwidthUppeL = 0;
offwidthUppeR = 2;
break;
}else{
        drawspel(row,col);
        drawspel(row-1,col);
drawspel(row,col+2);
drawspel(row-1,col+2);
        drawspel(row,col+4);
        drawspel(row-1,col+4);
        drawspel(row-2,col+4);
        drawspel(row-3,col+4);
```

```
                    width = 6;
                    hight = 4;
                    offhightUL= 0;
                    offhightNL= 2;
                    offhightUR= 0;
                    offhightNR = 0;
                    offwidthNedL = 0;
                    offwidthNedR = 0;
                    offwidthUppeL = 4;
                    offwidthUppeR = 0;
                    break;
                }
        case 7:

        if(rotera == 1){
        drawspel(row,col);
        drawspel(row-1,col);
drawspel(row,col+2);
drawspel(row-1,col+2);
drawspel(row-2,col+2);
drawspel(row-3,col+2);
drawspel(row-4,col+2);
drawspel(row-5,col+2);
width = 4;
hight = 6;
offhightUL= 0;
offhightNL= 4;
offhightUR= 0;
offhightNR = 0;
offwidthNedL = 0;
offwidthNedR = 0;
offwidthUppeL = 2;
offwidthUppeR = 0;

break;
        }else if(rotera == 2){

                drawspel(row,col);
                drawspel(row-1,col);
                drawspel(row-2,col);
                drawspel(row-3,col);
```

```
        drawspel(row,col+2);
        drawspel(row-1,col+2);
        drawspel(row,col+4);
        drawspel(row-1,col+4);

                width = 6;
                hight = 4;
                offhightUL= 0;
                offhightNL= 0;
                offhightUR= 0;
                offhightNR = 2;
                offwidthNedL = 0;
                offwidthNedR = 0;
                offwidthUppeL = 0;
                offwidthUppeR = 4;
                break;

}else if(rotera == 3){

drawspel(row,col);
drawspel(row-1,col);
        drawspel(row-2,col);
        drawspel(row-3,col);
        drawspel(row-4,col);
        drawspel(row-5,col);
        drawspel(row-4,col+2);
        drawspel(row-5,col+2);

        width = 4;
        hight = 6;
        offhightUL= 0;
        offhightNL= 0;
        offhightUR= 4;
        offhightNR = 0;
        offwidthNedL = 0;
        offwidthNedR = 2;
        offwidthUppeL = 0;
        offwidthUppeR = 0;
        break;

}else{
        drawspel(row-2,col);
        drawspel(row-3,col);
```

```
                        drawspel(row-2,col+2);
                        drawspel(row-3,col+2);
                        drawspel(row-2,col+4);
                        drawspel(row-3,col+4);
                        drawspel(row,col+4);
                        drawspel(row-1,col+4);
                        width = 6;
                        hight = 4;
                        offhightUL= 2;
                        offhightNL= 0;
                        offhightUR= 0;
                        offhightNR = 0;
                        offwidthNedL = 4;
                        offwidthNedR = 0;
                        offwidthUppeL = 0;
                        offwidthUppeR = 0;
                        break;

                }



                }
        }
}

void deleteispel(char x , char y){
        chip_1();
        SetYadrchip2(y);
        delay();
        SetXadrchip2(x);
        delay();
        rw_low();
        Di_high();
        displayRight[x/8][y]= displayRight[x/8][y]& ~(1<<(x%8));
        PORTB = displayRight[x/8][y];
        E_high();
        E_low();
        delay();
        delay();
        displayRight[x/8][y+1]= displayRight[x/8][y+1]& ~(1<<(x%8));
        PORTB = displayRight[x/8][y+1];
        E_high();
```

```
	E_low();
	delay();


}



void deleteblock(char z){
	switch (z)
	{
		case 1:

		deleteispel(row-3,col);
		deleteispel(row-3,col+2);
		deleteispel(row-2,col);
		deleteispel(row-2,col+2);
		deleteispel(row-1,col);
		deleteispel(row-1,col+2);
		deleteispel(row,col);
		deleteispel(row,col+2);

		break;

		case 2:
		if(rotera == 1 || rotera == 3){
			deleteispel(row,col);
			deleteispel(row-1,col);
			deleteispel(row-2,col);
			deleteispel(row-3,col);
			deleteispel(row-4,col);
			deleteispel(row-5,col);
			deleteispel(row-6,col);
			deleteispel(row-7,col);
			deleteispel(row-8,col);
			deleteispel(row-9,col);
			}else{
			deleteispel(row,col);
			deleteispel(row-1,col);
			deleteispel(row,col+2);
			deleteispel(row-1,col+2);
			deleteispel(row,col+4);
```

```
            deleteispel(row-1,col+4);
            deleteispel(row,col+6);
            deleteispel(row-1,col+6);
            deleteispel(row,col+8);
            deleteispel(row-1,col+8);

}




break;

case 3:

if(rotera == 1 || rotera ==3){
            deleteispel(row,col);
            deleteispel(row-1,col);
            deleteispel(row-2,col);
            deleteispel(row-3,col);

            deleteispel(row-2,col+2);
            deleteispel(row-3,col+2);
            deleteispel(row-4,col+2);
            deleteispel(row-5,col+2);

            }else{
            deleteispel(row-2, col);
            deleteispel(row-3, col);
            deleteispel(row-2, col+2);
            deleteispel(row-3, col+2);

            deleteispel(row, col+2);
            deleteispel(row-1, col+2);
            deleteispel(row, col+4);
            deleteispel(row-1, col+4);
}

break;
```

```
case 4:
                    if(rotera == 1 || rotera == 3){
        deleteispel(row,col+2);

        deleteispel(row-1,col+2);
        deleteispel(row-2,col);
        deleteispel(row-2,col+2);
        deleteispel(row-3,col);
        deleteispel(row-3,col+2);
        deleteispel(row-4,col);
        deleteispel(row-5,col);
        break;
                    }else{

                        deleteispel(row,col);
                        deleteispel(row-1,col);
                        deleteispel(row,col+2);
                        deleteispel(row-1,col+2);
                        deleteispel(row-2,col+2);
                        deleteispel(row-3,col+2);
                        deleteispel(row-2,col+4);
                        deleteispel(row-3,col+4);
                        break;

                    }

        case 5:
        if(rotera == 1){
        deleteispel(row,col);
        deleteispel(row-1,col);
        deleteispel(row,col+2);
        deleteispel(row-1,col+2);
        deleteispel(row-2,col+2);
        deleteispel(row-3,col+2);
        deleteispel(row,col+4);
        deleteispel(row-1,col+4);

        }else if(rotera == 2){
```

```
                                deleteispel(row,col);
                                deleteispel(row-1,col);
                                deleteispel(row-2,col);
                                deleteispel(row-3,col);
                                deleteispel(row-4,col);
                                deleteispel(row-5,col);
                                deleteispel(row-2,col+2);
                                deleteispel(row-3,col+2);
                        break;
        }else if(rotera  == 3){

        deleteispel(row-2,col);
        deleteispel(row-3,col);
        deleteispel(row-2,col+2);
        deleteispel(row-3,col+2);
        deleteispel(row-2,col+4);
        deleteispel(row-3,col+4);
        deleteispel(row,col+2);
        deleteispel(row-1,col+2);


        break;

}else{

deleteispel(row-2,col);
deleteispel(row-3,col);
deleteispel(row,col+2);
deleteispel(row-1,col+2);
deleteispel(row-2,col+2);
deleteispel(row-3,col+2);
deleteispel(row-4,col+2);
deleteispel(row-5,col+2);
break;
}

case 6:
if(rotera == 1){
deleteispel(row,col);
deleteispel(row-1,col);
deleteispel(row-2,col);
deleteispel(row-3,col);
```

```
deleteispel(row-4,col);
deleteispel(row-5,col);
deleteispel(row,col +2);
deleteispel(row-1,col +2);
break;
}else if(rotera == 2){
deleteispel(row,col);
deleteispel(row-1,col);
deleteispel(row-2,col);
deleteispel(row-3,col);
deleteispel(row-2,col+2);
deleteispel(row-3,col+2);
deleteispel(row-2,col+4);
deleteispel(row-3,col+4);
break;
}else if(rotera == 3){
deleteispel(row-4, col);
deleteispel(row-5,col);
deleteispel(row,col+2);
deleteispel(row-1,col+2);
deleteispel(row-2,col+2);
deleteispel(row-3,col+2);
deleteispel(row-4,col+2);
deleteispel(row-5,col+2);
break;
}else{
deleteispel(row,col);
deleteispel(row-1,col);
deleteispel(row,col+2);
deleteispel(row-1,col+2);
deleteispel(row,col+4);
deleteispel(row-1,col+4);
deleteispel(row-2,col+4);
deleteispel(row-3,col+4);
break;
}
case 7:
if(rotera == 1){
deleteispel(row,col);
deleteispel(row-1,col);
deleteispel(row,col+2);
deleteispel(row-1,col+2);
deleteispel(row-2,col+2);
```

```
deleteispel(row-3,col+2);
deleteispel(row-4,col+2);
deleteispel(row-5,col+2);
break;
}else if(rotera == 2){

deleteispel(row,col);
deleteispel(row-1,col);
deleteispel(row-2,col);
deleteispel(row-3,col);
deleteispel(row,col+2);
deleteispel(row-1,col+2);
deleteispel(row,col+4);
deleteispel(row-1,col+4);

break;


}else if(rotera == 3){

deleteispel(row,col);
deleteispel(row-1,col);
deleteispel(row-2,col);
deleteispel(row-3,col);
deleteispel(row-4,col);
deleteispel(row-5,col);
deleteispel(row-4,col+2);
deleteispel(row-5,col+2);
break;

}else{
deleteispel(row-2,col);
deleteispel(row-3,col);
deleteispel(row-2,col+2);
deleteispel(row-3,col+2);
deleteispel(row-2,col+4);
deleteispel(row-3,col+4);
deleteispel(row,col+4);
deleteispel(row-1,col+4);
break;

}
```

```
                        }

}




char OnOff(char x, char y){
        return (displayRight[x/8][y] & 1<<(x%8));
}


char Xclearrow;
char notclear;
void movedownrow(){
        char b;
        //cr = 61,
        for(char i = Xclearrow; i  >19; i--){
                for(char c = 2; c<62; c++){
                        if(OnOff(i-1,c)){
                                displayRight[i/8][c] = displayRight[i/8][c]| 1<<(i%8);
                                }else{
                                displayRight[i/8][c] = displayRight[i/8][c] & ~(1<<(i%8));
                        }
                }

        }

        for(char t = 0; t <8; t++){
                for(char u = 0; u < 64; u++){
                        SetXadrchip2(t*8);
                        delay();
                        SetYadrchip2(u);
                        delay();
                        rw_low();
                        Di_high();
                        PORTB = displayRight[t][u];
                        E_high();
                        E_low();
                        delay();
                }
```

```
	}

}
char displaycopy[8][64];




uint8_t roteratest(){
	char hit = 0;
	if(rotera == 1 || rotera == 3){
		for(int i = width; i<hight; i++){
			for(int p = 0; p<hight; p++){
				if(displayRight[(row-p)/8][col+i] & 1<<((row-p)%8)){
					hit = 1;
				}
			}
		}

		}else{

		for(int i = hight; i<width; i++){
			for(int p = 0; p < width; p++ ){
				if(displayRight[(row-i)/8][col+p] & 1<<((row-i)%8)){
					hit = 1;
				}
			}
		}
	}

	if(hit){
		return 0;
	}
	return 1;

}
char l;

void nyknapp(){

	char hit = 0;
```

```
if(PINA & (1<<3) && !(PINA & 1<< 2)&& !l){ // vänster

	// kollar vänster sidan
	for(int i = offhightUL; i<(hight-offhightNL); i++){
		if(displayRight[(row-i)/8][col-1] & (1<<((row-i)%8))){
			hit= 1;
		}
	}

	if(offhightNL){
		for(int i = hight-offhightNL; i<hight; i++){
			if(displayRight[(row-i)/8][col+offwidthUppeL-1] & 1<<(row-i)%8){
				hit = 1;
			}
		}
	}

	if (offhightUL){
		for(int i = 0; i<offhightUL; i++){
			if(displayRight[(row-i)/8][col-1+offwidthNedL] & 1<<(row+i)%8){
				hit = 1;
			}
		}
	}
	if(!hit){
		col-=2;
	}
}//klar

if(PINA & (1<<1) && !PINC && !(PINA & 1<<2) && !l ){
	hit = 0;


	//Kollar höger
	for(int i = offhightUR; i<(hight-offhightNR); i++){
		if(displayRight[(row-i)/8][col+width] & (1<<((row-i)%8))){
			hit= 1;
		}
	}



	if(offhightNR){
```

```
                        for(int i = hight-offhightNR; i<hight; i++){
                                if(displayRight[(row-i)/8][col+width-offwidthUppeR] &
1<<(row-i)%8){
                                        hit = 1;
                                }
                        }
                }

                if (offhightUR){
                        for(int i = 0; i<offhightUR; i++){
                                if(displayRight[(row-i)/8][col+width-offwidthNedR] &
1<<(row-i)%8){
                                        hit = 1;
                                }
                        }
                }

                if(!hit){

                        col+=2;
                }
        }


        if(PINA & (1<<2) && !PINC){

                l=1;
        }



}


void Startmeny(){


        drawLeft(9,3); //T
        drawLeft(9,4);
        drawLeft(9,5);
        drawLeft(9,6);
        drawLeft(9,7);
        drawLeft(9,8);
```

```
drawLeft(9,9);
drawLeft(9,10);
drawLeft(9,11);
drawLeft(10,3);
drawLeft(10,4);
drawLeft(10,5);
drawLeft(10,6);
drawLeft(10,7);
drawLeft(10,8);
drawLeft(10,9);
drawLeft(10,10);
drawLeft(10,11);
drawLeft(10,7);
drawLeft(11,7);
drawLeft(12,7);
drawLeft(13,7);
drawLeft(14,7);
drawLeft(15,7);
drawLeft(16,7);
drawLeft(17,7);
drawLeft(18,7);
//-------------------------------------------------------------------------------


drawLeft(9,15); //E
drawLeft(9,16);
drawLeft(9,17);
drawLeft(9,18);
drawLeft(9,19);
drawLeft(9,20);
drawLeft(10,15);
drawLeft(10,16);
drawLeft(10,17);
drawLeft(10,18);
drawLeft(10,19);
drawLeft(10,20);
drawLeft(11,15);
drawLeft(12,15);
drawLeft(13,17);
drawLeft(13,15);
drawLeft(14,17);
drawLeft(14,15);
drawLeft(15,15);
```

```
drawLeft(16,15);
drawLeft(17,17);
drawLeft(17,15);
drawLeft(18,15);
drawLeft(18,17);
drawLeft(17,19);
drawLeft(18,19);
drawLeft(17,20);
drawLeft(18,20);
//-------------------------------------------------------------------------------


drawLeft(9,23); //T
drawLeft(9,24);
drawLeft(9,25);
drawLeft(9,26);
drawLeft(9,27);
drawLeft(9,28);
drawLeft(9,29);
drawLeft(9,30);
drawLeft(9,31);
drawLeft(10,23);
drawLeft(10,24);
drawLeft(10,25);
drawLeft(10,26);
drawLeft(10,27);
drawLeft(10,28);
drawLeft(10,29);
drawLeft(10,30);
drawLeft(10,31);
drawLeft(10,27);
drawLeft(11,27);
drawLeft(12,27);
drawLeft(13,27);
drawLeft(14,27);
drawLeft(15,27);
drawLeft(16,27);
drawLeft(17,27);
drawLeft(18,27);
//-----------------------------------------------------------------------------------


drawLeft(9,34); //R
```

```
drawLeft(10,34);
drawLeft(11,34);
drawLeft(12,34);
drawLeft(13,34);
drawLeft(14,34);
drawLeft(15,34);
drawLeft(16,34);
drawLeft(17,34);
drawLeft(18,34);
drawLeft(9,40);
drawLeft(10,40);
drawLeft(11,40);
drawLeft(12,40);
drawLeft(13,40);
drawLeft(14,40);
drawLeft(9,36);
drawLeft(10,36);
drawLeft(9,38);
drawLeft(10,38);
drawLeft(13,36);
drawLeft(14,36);
drawLeft(13,38);
drawLeft(14,38);
drawLeft(15,38);
drawLeft(16,38);
drawLeft(17,40);
drawLeft(18,40);

//-----------------------------------------------------------------------------------------

drawLeft(9,44);  // l
drawLeft(10,44);
drawLeft(11,44);
drawLeft(12,44);
drawLeft(13,44);
drawLeft(14,44);
drawLeft(15,44);
drawLeft(16,44);
drawLeft(17,44);
drawLeft(18,44);

//-------------------------------------------------------------------------------
```

```
	drawLeft(9, 48); // S
	drawLeft(10, 48);
	drawLeft(9, 50);
	drawLeft(10, 50);
	drawLeft(9, 52);
	drawLeft(10, 52);
	drawLeft(9, 54);
	drawLeft(10, 54);
	drawLeft(11, 48);
	drawLeft(12, 48);
	drawLeft(13, 48);
	drawLeft(14, 48);
	drawLeft(13, 50);
	drawLeft(14, 50);
	drawLeft(13, 52);
	drawLeft(14, 52);
	drawLeft(13, 54);
	drawLeft(14, 54);
	drawLeft(15, 54);
	drawLeft(16, 54);
	drawLeft(17, 48);
	drawLeft(18, 48);
	drawLeft(17, 50);
	drawLeft(18, 50);
	drawLeft(17, 52);
	drawLeft(18, 52);
	drawLeft(17, 54);
	drawLeft(18, 54);
}

void TextGame(char x, char y) {

	//---------------------------------------- G
	drawspel(x,y);
	drawspel(x-1,y);
	drawspel(x-2,y);
	drawspel(x-3,y);
	drawspel(x-4,y);
	drawspel(x-5,y);
	drawspel(x-6,y);
	drawspel(x-7,y);
	drawspel(x-8,y);
```

```
drawspel(x-9,y);
drawspel(x-10,y);
drawspel(x-11,y);


drawspel(x-10,y+4);
drawspel(x-10,y+2);
drawspel(x-10,y+6);
drawspel(x-10,y+8);
drawspel(x-11,y+4);
drawspel(x-11,y+2);
drawspel(x-11,y+6);
drawspel(x-11,y+8);

drawspel(x,y+4);
drawspel(x,y+2);
drawspel(x,y+6);
drawspel(x,y+8);
drawspel(x-1,y+4);
drawspel(x-1,y+2);
drawspel(x-1,y+6);
drawspel(x-1,y+8);

drawspel(x,y+8);
drawspel(x-1,y+8);
drawspel(x-2,y+8);
drawspel(x-3,y+8);
drawspel(x-4,y+8);
drawspel(x-5,y+8);


drawspel(x-5,y+6);
drawspel(x-5,y+8);
drawspel(x-4,y+6);
drawspel(x-4,y+8);


//------------------------------------------ A

drawspel(x,y+12);
drawspel(x-1,y+12);
drawspel(x-2,y+12);
drawspel(x-3,y+12);
```

```
drawspel(x-4,y+12);
drawspel(x-5,y+12);
drawspel(x-6,y+12);
drawspel(x-7,y+12);

drawspel(x-8,y+14);
drawspel(x-9,y+14);

drawspel(x-8,y+20);
drawspel(x-9,y+20);

drawspel(x-10,y+16);
drawspel(x-11,y+16);


drawspel(x-10,y+18);
drawspel(x-11,y+18);

drawspel(x,y+22);
drawspel(x-1,y+22);
drawspel(x-2,y+22);
drawspel(x-3,y+22);
drawspel(x-4,y+22);
drawspel(x-5,y+22);
drawspel(x-6,y+22);
drawspel(x-7,y+22);


drawspel(x-4,y+14);
drawspel(x-4,y+16);
drawspel(x-4,y+18);
drawspel(x-4,y+20);


drawspel(x-5,y+14);
drawspel(x-5,y+16);
drawspel(x-5,y+18);
drawspel(x-5,y+20);

// ---------------------------------------------------- M

drawspel (x,y+26);
drawspel (x-1,y+26);
```

```
drawspel (x-2,y+26);
drawspel (x-3,y+26);
drawspel (x-4,y+26);
drawspel (x-5,y+26);
drawspel (x-6,y+26);
drawspel (x-7,y+26);
drawspel (x-8,y+26);
drawspel (x-9,y+26);
drawspel (x-10,y+26);
drawspel (x-11,y+26);


drawspel (x-8,y+28);
drawspel (x-9,y+28);

drawspel (x-6,y+30);
drawspel (x-7,y+30);


drawspel (x-8,y+32);
drawspel (x-9,y+32);

drawspel (x,y+34);
drawspel (x-1,y+34);
drawspel (x-2,y+34);
drawspel (x-3,y+34);
drawspel (x-4,y+34);
drawspel (x-5,y+34);
drawspel (x-6,y+34);
drawspel (x-7,y+34);
drawspel (x-8,y+34);
drawspel (x-9,y+34);
drawspel (x-10,y+34);
drawspel (x-11,y+34);

// ---------------------------------------------- E

drawspel (x,y+38);
drawspel (x-1,y+38);
drawspel (x-2,y+38);
drawspel (x-3,y+38);
drawspel (x-4,y+38);
drawspel (x-5,y+38);
```

```
drawspel (x-6,y+38);
drawspel (x-7,y+38);
drawspel (x-8,y+38);
drawspel (x-9,y+38);
drawspel (x-10,y+38);
drawspel (x-11,y+38);


drawspel (x,y+40);
drawspel (x,y+42);
drawspel (x,y+44);
drawspel (x,y+46);

drawspel (x-1,y+40);
drawspel (x-1,y+42);
drawspel (x-1,y+44);
drawspel (x-1,y+46);

drawspel (x-6,y+40);
drawspel (x-6,y+42);
drawspel (x-6,y+44);


drawspel (x-5,y+40);
drawspel (x-5,y+42);
drawspel (x-5,y+44);

drawspel (x-11,y+40);
drawspel (x-11,y+42);
drawspel (x-11,y+44);
drawspel (x-11,y+46);

drawspel (x-10,y+40);
drawspel (x-10,y+42);
drawspel (x-10,y+44);
drawspel (x-10,y+46);
```

```
}

void TextOver(char x, char y){

        //---------------------------------------- O

        drawspel(x-2,y);
        drawspel(x-3,y);
        drawspel(x-4,y);
        drawspel(x-5,y);
        drawspel(x-6,y);
        drawspel(x-7,y);
        drawspel(x-8,y);
        drawspel(x-9,y);

        drawspel(x-10,y+2);
        drawspel(x-10,y+4);
        drawspel(x-10,y+6);
        drawspel(x-11,y+2);
        drawspel(x-11,y+4);
        drawspel(x-11,y+6);

        drawspel(x,y+2);
        drawspel(x,y+4);
        drawspel(x,y+6);
        drawspel(x-1,y+2);
        drawspel(x-1,y+4);
```

```
drawspel(x-1,y+6);

drawspel(x-2,y+8);
drawspel(x-3,y+8);
drawspel(x-4,y+8);
drawspel(x-5,y+8);
drawspel(x-6,y+8);
drawspel(x-7,y+8);
drawspel(x-8,y+8);
drawspel(x-9,y+8);


// ------------------------------------------- V

drawspel(x-6,y+12);
drawspel(x-7,y+12);
drawspel(x-8,y+12);
drawspel(x-9,y+12);
drawspel(x-10,y+12);
drawspel(x-11,y+12);

drawspel(x-5,y+14);
drawspel(x-4,y+14);
drawspel(x-3,y+14);
drawspel(x-2,y+14);

drawspel(x-1,y+16);
drawspel(x,y+16);

drawspel(x-5,y+18);
drawspel(x-4,y+18);
drawspel(x-3,y+18);
drawspel(x-2,y+18);

drawspel(x-6,y+20);
drawspel(x-7,y+20);
drawspel(x-8,y+20);
drawspel(x-9,y+20);
drawspel(x-10,y+20);
drawspel(x-11,y+20);


//------------------------------------------- E
```

```
drawspel(x,y+24);
drawspel(x-1,y+24);
drawspel(x-2,y+24);
drawspel(x-3,y+24);
drawspel(x-4,y+24);
drawspel(x-5,y+24);
drawspel(x-6,y+24);
drawspel(x-7,y+24);
drawspel(x-8,y+24);
drawspel(x-9,y+24);
drawspel(x-10,y+24);
drawspel(x-11,y+24);

drawspel(x,y+24);
drawspel(x,y+26);
drawspel(x,y+28);
drawspel(x,y+30);

drawspel(x-1,y+24);
drawspel(x-1,y+26);
drawspel(x-1,y+28);
drawspel(x-1,y+30);

drawspel(x-5,y+24);
drawspel(x-5,y+26);
drawspel(x-5,y+28);

drawspel(x-6,y+24);
drawspel(x-6,y+26);
drawspel(x-6,y+28);


drawspel(x-10,y+24);
drawspel(x-10,y+26);
drawspel(x-10,y+28);
drawspel(x-10,y+30);

drawspel(x-11,y+24);
drawspel(x-11,y+26);
drawspel(x-11,y+28);
drawspel(x-11,y+30);
//---------------------------------------------- R
```

```
drawspel(x,y+34);
drawspel(x-1,y+34);
drawspel(x-2,y+34);
drawspel(x-3,y+34);
drawspel(x-4,y+34);
drawspel(x-5,y+34);
drawspel(x-6,y+34);
drawspel(x-7,y+34);
drawspel(x-8,y+34);
drawspel(x-9,y+34);
drawspel(x-10,y+34);
drawspel(x-11,y+34);

drawspel(x-10,y+36);
drawspel(x-11,y+36);
drawspel(x-10,y+38);
drawspel(x-11,y+38);
drawspel(x-10,y+40);


drawspel(x-8,y+40);
drawspel(x-9,y+40);

drawspel(x-7,y+40);
drawspel(x-6,y+40);


drawspel(x-5,y+38);
drawspel(x-6,y+38);
drawspel(x-5,y+36);
drawspel(x-6,y+36);

drawspel(x-10,38);

drawspel(x-4,y+38);
drawspel(x-5,y+38);

drawspel(x-3,y+40);
drawspel(x-2,y+40);
```

```
        drawspel(x-1,y+40);
        drawspel(x,y+40);









}

void Press(char x, char y){
        drawLeft(x,y);
        drawLeft(x-1,y);
        drawLeft(x-2,y);
        drawLeft(x-3,y);
        drawLeft(x-4,y);
        drawLeft(x-5,y);
        drawLeft(x-6,y);
        drawLeft(x-7,y);
        drawLeft(x-8,y);
        drawLeft(x-9,y);
        drawLeft(x-10,y);
        drawLeft(x-11,y);

        drawLeft(x-10,y+2);
        drawLeft(x-11,y+2);
        drawLeft(x-10,y+4);
        drawLeft(x-10,y+6);
        drawLeft(x-11,y+4);
```

```
drawLeft(x-6,y+6);
drawLeft(x-7,y+6);
drawLeft(x-8,y+6);
drawLeft(x-9,y+6);

drawLeft(x-6,y+2);
drawLeft(x-5,y+2);
drawLeft(x-6,y+4);
drawLeft(x-5,y+4);

// ------------------------------------------- R

drawLeft(x,y+10);
drawLeft(x-1,y+10);
drawLeft(x-2,y+10);
drawLeft(x-3,y+10);
drawLeft(x-4,y+10);
drawLeft(x-5,y+10);
drawLeft(x-6,y+10);
drawLeft(x-7,y+10);
drawLeft(x-8,y+10);
drawLeft(x-9,y+10);
drawLeft(x-10,y+10);
drawLeft(x-10,y+16);
drawLeft(x-11,y+10);

drawLeft(x-10,y+12);
drawLeft(x-11,y+12);
drawLeft(x-10,y+14);
drawLeft(x-11,y+14);

drawLeft(x-6,y+16);
drawLeft(x-7,y+16);
drawLeft(x-8,y+16);
drawLeft(x-9,y+16);


drawLeft(x-6,y+14);
drawLeft(x-5,y+14);
drawLeft(x-6,y+12);
drawLeft(x-5,y+12);

drawLeft(x-4,y+14);
```

```
drawLeft(x-5,y+14);

drawLeft(x-3,y+16);
drawLeft(x-2,y+16);

drawLeft(x-1,y+16);
drawLeft(x,y+16);


// ------------------------------------------------------ E

drawLeft(x,y+20);
drawLeft(x-1,y+20);
drawLeft(x-2,y+20);
drawLeft(x-3,y+20);
drawLeft(x-4,y+20);
drawLeft(x-5,y+20);
drawLeft(x-6,y+20);
drawLeft(x-7,y+20);
drawLeft(x-8,y+20);
drawLeft(x-9,y+20);
drawLeft(x-10,y+20);
drawLeft(x-10,y+20);
drawLeft(x-11,y+20);

drawLeft(x,y+22);
drawLeft(x-1,y+22);
drawLeft(x,y+24);
drawLeft(x-1,y+24);
drawLeft(x,y+26);
drawLeft(x-1,y+26);

drawLeft(x-6,y+22);
drawLeft(x-5,y+22);
drawLeft(x-6,y+24);
drawLeft(x-5,y+24);

drawLeft(x-10,y+22);
drawLeft(x-11,y+22);
drawLeft(x-10,y+24);
drawLeft(x-11,y+24);
drawLeft(x-10,y+26);
drawLeft(x-11,y+26);
```

```
//----------------------------------- S

drawLeft(x,y+30);
drawLeft(x,y+32);
drawLeft(x,y+34);
drawLeft(x,y+36);

drawLeft(x-1,y+30);
drawLeft(x-1,y+32);
drawLeft(x-1,y+34);
drawLeft(x-1,y+36);


drawLeft(x-2,y+36);
drawLeft(x-3,y+36);
drawLeft(x-4,y+36);

drawLeft(x-6,y+36);
drawLeft(x-5,y+36);
drawLeft(x-6,y+32);
drawLeft(x-5,y+32);
drawLeft(x-6,y+34);
drawLeft(x-5,y+34);
drawLeft(x-6,y+30);
drawLeft(x-5,y+30);

drawLeft(x-6,y+30);
drawLeft(x-7,y+30);
drawLeft(x-8,y+30);
drawLeft(x-9,y+30);
drawLeft(x-10,y+30);
drawLeft(x-11,y+30);

drawLeft(x-10,y+32);
drawLeft(x-11,y+32);
drawLeft(x-10,y+34);
drawLeft(x-11,y+34);
drawLeft(x-10,y+36);
drawLeft(x-11,y+36);
```

```
        // ------------------------------------------------ S

        drawLeft(x,y+30+10);
        drawLeft(x,y+32+10);
        drawLeft(x,y+34+10);
        drawLeft(x,y+36+10);

        drawLeft(x-1,y+30+10);
        drawLeft(x-1,y+32+10);
        drawLeft(x-1,y+34+10);
        drawLeft(x-1,y+36+10);


        drawLeft(x-2,y+36+10);
        drawLeft(x-3,y+36+10);
        drawLeft(x-4,y+36+10);

        drawLeft(x-6,y+36+10);
        drawLeft(x-5,y+36+10);
        drawLeft(x-6,y+32+10);
        drawLeft(x-5,y+32+10);
        drawLeft(x-6,y+34+10);
        drawLeft(x-5,y+34+10);
        drawLeft(x-6,y+30+10);
        drawLeft(x-5,y+30+10);

        drawLeft(x-6,y+30+10);
        drawLeft(x-7,y+30+10);
        drawLeft(x-8,y+30+10);
        drawLeft(x-9,y+30+10);
        drawLeft(x-10,y+30+10);
        drawLeft(x-11,y+30+10);

        drawLeft(x-10,y+32+10);
        drawLeft(x-11,y+32+10);
        drawLeft(x-10,y+34+10);
        drawLeft(x-11,y+34+10);
        drawLeft(x-10,y+36+10);
        drawLeft(x-11,y+36+10);

}

void LR(char x, char y){
```

```
// ---------------------------------- L

drawLeft(x,y);
drawLeft(x-1,y);
drawLeft(x-2,y);
drawLeft(x-3,y);
drawLeft(x-4,y);
drawLeft(x-5,y);
drawLeft(x-6,y);
drawLeft(x-7,y);
drawLeft(x-8,y);
drawLeft(x-9,y);
drawLeft(x-10,y);
drawLeft(x-11,y);

drawLeft(x,y+2);
drawLeft(x-1,y+2);
drawLeft(x,y+4);
drawLeft(x-1,y+4);
drawLeft(x,y+6);
drawLeft(x-1,y+6);

// ---------------------------------- +


drawLeft(x-2,y+12);
drawLeft(x-3,y+12);
drawLeft(x-4,y+12);
drawLeft(x-5,y+12);
drawLeft(x-6,y+12);
drawLeft(x-7,y+12);

drawLeft(x-5,y+10);
drawLeft(x-4,y+10);
drawLeft(x-5,y+14);
drawLeft(x-4,y+14);
drawLeft(x-5,y+12);
drawLeft(x-4,y+12);

// ----------------------------  R
char z = 10 ;
drawLeft(x,y+10+z);
```

```
        drawLeft(x-1,y+10+z);
        drawLeft(x-2,y+10+z);
        drawLeft(x-3,y+10+z);
        drawLeft(x-4,y+10+z);
        drawLeft(x-5,y+10+z);
        drawLeft(x-6,y+10+z);
        drawLeft(x-7,y+10+z);
        drawLeft(x-8,y+10+z);
        drawLeft(x-9,y+10+z);
        drawLeft(x-10,y+10+z);
        drawLeft(x-10,y+16+z);
        drawLeft(x-11,y+10+z);

        drawLeft(x-10,y+12+z);
        drawLeft(x-11,y+12+z);
        drawLeft(x-10,y+14+z);
        drawLeft(x-11,y+14+z);

        drawLeft(x-6,y+16+z);
        drawLeft(x-7,y+16+z);
        drawLeft(x-8,y+16+z);
        drawLeft(x-9,y+16+z);


        drawLeft(x-6,y+14+z);
        drawLeft(x-5,y+14+z);
        drawLeft(x-6,y+12+z);
        drawLeft(x-5,y+12+z);

        drawLeft(x-4,y+14+z);
        drawLeft(x-5,y+14+z);
        drawLeft(x-3,y+16+z);
        drawLeft(x-2,y+16+z);

        drawLeft(x-1,y+16+z);
        drawLeft(x,y+16+z);

}

void deleteLeft(char x , char y){
        chip_2();
        SetYadrchip1(y);
        delay();
```

```
        SetXadrchip1(x);
        delay();
        rw_low();
        Di_high();
        displayLeft[x/8][y]= displayLeft[x/8][y] & ~(1<<(x%8));
        PORTB = displayLeft[x/8][y];
        E_high();
        E_low();
        delay();
        delay();
        displayLeft[x/8][y+1]= displayLeft[x/8][y+1] & ~(1<<(x%8));
        E_high();
        E_low();
        delay();


}
void DeletePress(char x, char y){
        deleteLeft(x,y);
        deleteLeft(x-1,y);
        deleteLeft(x-2,y);
        deleteLeft(x-3,y);
        deleteLeft(x-4,y);
        deleteLeft(x-5,y);
        deleteLeft(x-6,y);
        deleteLeft(x-7,y);
        deleteLeft(x-8,y);
        deleteLeft(x-9,y);
        deleteLeft(x-10,y);
        deleteLeft(x-11,y);

        deleteLeft(x-10,y+2);
        deleteLeft(x-11,y+2);
        deleteLeft(x-10,y+4);
        deleteLeft(x-10,y+6);
        deleteLeft(x-11,y+4);

        deleteLeft(x-6,y+6);
        deleteLeft(x-7,y+6);
        deleteLeft(x-8,y+6);
        deleteLeft(x-9,y+6);

        deleteLeft(x-6,y+2);
```

```
deleteLeft(x-5,y+2);
deleteLeft(x-6,y+4);
deleteLeft(x-5,y+4);

// ------------------------------------------ R

deleteLeft(x,y+10);
deleteLeft(x-1,y+10);
deleteLeft(x-2,y+10);
deleteLeft(x-3,y+10);
deleteLeft(x-4,y+10);
deleteLeft(x-5,y+10);
deleteLeft(x-6,y+10);
deleteLeft(x-7,y+10);
deleteLeft(x-8,y+10);
deleteLeft(x-9,y+10);
deleteLeft(x-10,y+10);
deleteLeft(x-10,y+16);
deleteLeft(x-11,y+10);

deleteLeft(x-10,y+12);
deleteLeft(x-11,y+12);
deleteLeft(x-10,y+14);
deleteLeft(x-11,y+14);

deleteLeft(x-6,y+16);
deleteLeft(x-7,y+16);
deleteLeft(x-8,y+16);
deleteLeft(x-9,y+16);


deleteLeft(x-6,y+14);
deleteLeft(x-5,y+14);
deleteLeft(x-6,y+12);
deleteLeft(x-5,y+12);

deleteLeft(x-4,y+14);
deleteLeft(x-5,y+14);

deleteLeft(x-3,y+16);
deleteLeft(x-2,y+16);

deleteLeft(x-1,y+16);
```

```
deleteLeft(x,y+16);


// ------------------------------------------------------ E

deleteLeft(x,y+20);
deleteLeft(x-1,y+20);
deleteLeft(x-2,y+20);
deleteLeft(x-3,y+20);
deleteLeft(x-4,y+20);
deleteLeft(x-5,y+20);
deleteLeft(x-6,y+20);
deleteLeft(x-7,y+20);
deleteLeft(x-8,y+20);
deleteLeft(x-9,y+20);
deleteLeft(x-10,y+20);
deleteLeft(x-10,y+20);
deleteLeft(x-11,y+20);

deleteLeft(x,y+22);
deleteLeft(x-1,y+22);
deleteLeft(x,y+24);
deleteLeft(x-1,y+24);
deleteLeft(x,y+26);
deleteLeft(x-1,y+26);

deleteLeft(x-6,y+22);
deleteLeft(x-5,y+22);
deleteLeft(x-6,y+24);
deleteLeft(x-5,y+24);

deleteLeft(x-10,y+22);
deleteLeft(x-11,y+22);
deleteLeft(x-10,y+24);
deleteLeft(x-11,y+24);
deleteLeft(x-10,y+26);
deleteLeft(x-11,y+26);


//------------------------------------ S

deleteLeft(x,y+30);
deleteLeft(x,y+32);
```

```
deleteLeft(x,y+34);
deleteLeft(x,y+36);

deleteLeft(x-1,y+30);
deleteLeft(x-1,y+32);
deleteLeft(x-1,y+34);
deleteLeft(x-1,y+36);


deleteLeft(x-2,y+36);
deleteLeft(x-3,y+36);
deleteLeft(x-4,y+36);

deleteLeft(x-6,y+36);
deleteLeft(x-5,y+36);
deleteLeft(x-6,y+32);
deleteLeft(x-5,y+32);
deleteLeft(x-6,y+34);
deleteLeft(x-5,y+34);
deleteLeft(x-6,y+30);
deleteLeft(x-5,y+30);

deleteLeft(x-6,y+30);
deleteLeft(x-7,y+30);
deleteLeft(x-8,y+30);
deleteLeft(x-9,y+30);
deleteLeft(x-10,y+30);
deleteLeft(x-11,y+30);

deleteLeft(x-10,y+32);
deleteLeft(x-11,y+32);
deleteLeft(x-10,y+34);
deleteLeft(x-11,y+34);
deleteLeft(x-10,y+36);
deleteLeft(x-11,y+36);


// ---------------------------------------------- S

deleteLeft(x,y+30+10);
deleteLeft(x,y+32+10);
deleteLeft(x,y+34+10);
deleteLeft(x,y+36+10);
```

```
        deleteLeft(x-1,y+30+10);
        deleteLeft(x-1,y+32+10);
        deleteLeft(x-1,y+34+10);
        deleteLeft(x-1,y+36+10);


        deleteLeft(x-2,y+36+10);
        deleteLeft(x-3,y+36+10);
        deleteLeft(x-4,y+36+10);

        deleteLeft(x-6,y+36+10);
        deleteLeft(x-5,y+36+10);
        deleteLeft(x-6,y+32+10);
        deleteLeft(x-5,y+32+10);
        deleteLeft(x-6,y+34+10);
        deleteLeft(x-5,y+34+10);
        deleteLeft(x-6,y+30+10);
        deleteLeft(x-5,y+30+10);

        deleteLeft(x-6,y+30+10);
        deleteLeft(x-7,y+30+10);
        deleteLeft(x-8,y+30+10);
        deleteLeft(x-9,y+30+10);
        deleteLeft(x-10,y+30+10);
        deleteLeft(x-11,y+30+10);

        deleteLeft(x-10,y+32+10);
        deleteLeft(x-11,y+32+10);
        deleteLeft(x-10,y+34+10);
        deleteLeft(x-11,y+34+10);
        deleteLeft(x-10,y+36+10);
        deleteLeft(x-11,y+36+10);

}

void DeleteLR(char x, char y){

        // ------------------------------------ L

        deleteLeft(x,y);
        deleteLeft(x-1,y);
        deleteLeft(x-2,y);
```

```
deleteLeft(x-3,y);
deleteLeft(x-4,y);
deleteLeft(x-5,y);
deleteLeft(x-6,y);
deleteLeft(x-7,y);
deleteLeft(x-8,y);
deleteLeft(x-9,y);
deleteLeft(x-10,y);
deleteLeft(x-11,y);

deleteLeft(x,y+2);
deleteLeft(x-1,y+2);
deleteLeft(x,y+4);
deleteLeft(x-1,y+4);
deleteLeft(x,y+6);
deleteLeft(x-1,y+6);

// ---------------------------------- +


deleteLeft(x-2,y+12);
deleteLeft(x-3,y+12);
deleteLeft(x-4,y+12);
deleteLeft(x-5,y+12);
deleteLeft(x-6,y+12);
deleteLeft(x-7,y+12);

deleteLeft(x-5,y+10);
deleteLeft(x-4,y+10);
deleteLeft(x-5,y+14);
deleteLeft(x-4,y+14);
deleteLeft(x-5,y+12);
deleteLeft(x-4,y+12);

// ----------------------------  R
char z = 10 ;
deleteLeft(x,y+10+z);
deleteLeft(x-1,y+10+z);
deleteLeft(x-2,y+10+z);
deleteLeft(x-3,y+10+z);
deleteLeft(x-4,y+10+z);
deleteLeft(x-5,y+10+z);
deleteLeft(x-6,y+10+z);
```

```
        deleteLeft(x-7,y+10+z);
        deleteLeft(x-8,y+10+z);
        deleteLeft(x-9,y+10+z);
        deleteLeft(x-10,y+10+z);
        deleteLeft(x-10,y+16+z);
        deleteLeft(x-11,y+10+z);

        deleteLeft(x-10,y+12+z);
        deleteLeft(x-11,y+12+z);
        deleteLeft(x-10,y+14+z);
        deleteLeft(x-11,y+14+z);

        deleteLeft(x-6,y+16+z);
        deleteLeft(x-7,y+16+z);
        deleteLeft(x-8,y+16+z);
        deleteLeft(x-9,y+16+z);


        deleteLeft(x-6,y+14+z);
        deleteLeft(x-5,y+14+z);
        deleteLeft(x-6,y+12+z);
        deleteLeft(x-5,y+12+z);

        deleteLeft(x-4,y+14+z);
        deleteLeft(x-5,y+14+z);
        deleteLeft(x-3,y+16+z);
        deleteLeft(x-2,y+16+z);

        deleteLeft(x-1,y+16+z);
        deleteLeft(x,y+16+z);

}
```

///// NY

```
const char zero[10][8] ={
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1}
};

const char ONE[10][8] ={
        {0,0,1,1,1,0,0,0},
        {0,1,1,1,1,0,0,0},
        {1,1,0,1,1,0,0,0},
        {1,0,0,1,1,0,0,0},
        {0,0,0,1,1,0,0,0},
        {0,0,0,1,1,0,0,0},
        {0,0,0,1,1,0,0,0},
        {0,0,0,1,1,0,0,0},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1}
};

const char TWO[10][8] ={
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {0,0,0,0,0,1,1,1},
        {0,0,0,0,0,1,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,0,0,0,0,0,0},
        {1,1,0,0,0,0,0,0},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1}
};
```

```
const char THREE[10][8] ={
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {0,0,0,0,0,1,1},
        {0,0,0,0,0,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {0,0,0,0,0,1,1},
        {0,0,0,0,0,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1}
};
const char FOUR[10][8] ={
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {0,0,0,0,0,1,1},
        {0,0,0,0,0,1,1},
        {0,0,0,0,0,1,1},
        {0,0,0,0,0,1,1}
};
const char FIVE[10][8] ={
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,0,0,0,0,0,0},
        {1,1,0,0,0,0,0,0},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {0,0,0,0,0,1,1},
        {0,0,0,0,0,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1}
};
const char SIX[10][8] ={
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,0,0,0,0,0,0},
        {1,1,0,0,0,0,0,0},
        {1,1,1,1,1,1,1,1},
```

```
        {1,1,1,1,1,1,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1}
};
const char SEVEN[10][8] ={
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {0,0,0,0,1,1,0},
        {0,0,0,1,1,0,0},
        {0,0,0,1,1,0,0},
        {0,0,1,1,1,0,0},
        {0,1,1,1,0,0,0},
        {0,1,1,0,0,0,0},
        {0,1,1,0,0,0,0},
        {0,1,1,0,0,0,0}
};


const char EIGHT[10][8] ={
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1}
};

const char NINE[10][8] ={
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1,1},
        {0,0,0,0,0,1,1},
        {0,0,0,0,0,1,1},
        {0,0,0,0,0,1,1},
```

```
        {0,0,0,0,0,1,1}
};

const char S[10][8] ={
        {1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1},
        {1,1,0,0,0,0,0},
        {1,1,0,0,0,0,0},
        {1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1},
        {0,0,0,0,0,1,1},
        {0,0,0,0,0,1,1},
        {1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1}
};
const char C[10][8] ={
        {0,0,1,1,1,1,1},
        {0,1,1,1,1,1,1},
        {0,1,1,0,0,0,0},
        {1,1,0,0,0,0,0},
        {1,1,0,0,0,0,0},
        {1,1,0,0,0,0,0},
        {1,1,0,0,0,0,0},
        {1,1,1,0,0,0,0},
        {0,1,1,1,1,1,1},
        {0,0,1,1,1,1,1}
};

const char O[10][8] ={
        {0,1,1,1,1,1,0},
        {1,1,1,1,1,1,1},
        {1,1,1,0,1,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,1,0,1,1,1},
        {1,1,1,1,1,1,1},
        {0,1,1,1,1,1,0}
};

const char R[10][8] ={
        {1,1,1,1,1,1,1},
```

```
        {1,1,1,1,1,1,1,1},
        {1,1,0,0,0,1,1},
        {1,1,0,0,0,1,1},
        {1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1},
        {1,1,1,1,0,0,0},
        {1,1,0,1,1,0,0},
        {1,1,0,1,1,1,1},
        {1,1,0,0,1,1,1}
};

const char E[10][8] ={
        {1,1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1},
        {1,1,0,0,0,0,0},
        {1,1,0,0,0,0,0},
        {1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1},
        {1,1,0,0,0,0,0},
        {1,1,0,0,0,0,0},
        {1,1,1,1,1,1,1},
        {1,1,1,1,1,1,1}
};



void drawText(char x , char y){
        chip_2();
        SetYadrchip1(y);
        delay();
        SetXadrchip1(x);
        delay();
        rw_low();
        Di_high();
        displayLeft[x/8][y]= displayLeft[x/8][y]| (1<<(x%8));
        PORTB = displayLeft[x/8][y];
        E_high();
        E_low();
        delay();



}
```

```
void DrawList(char w[10][8], char Startx, char Starty){


	char y = Starty;
	char x = Startx;

	for(char i = 0; i<10; i++){
		y = Starty;
		x = Startx+i;
		for(char b = 0; b<8 ; b++){
		y = Starty + b;


		if(w[i][b]){
			drawText(x,y);
		}
		}

	}
}

char score[5];
void DrawScore(char startx, char starty){/// skriver ut scoren som är i matrisen

	for(int i = 0 ; i<5; i++){

		switch(score[i]){
			case 0:


			DrawList(zero, startx, starty);
			break;
			case 1:
			DrawList(ONE, startx, starty);
			break;
			case 2:
			DrawList(TWO, startx, starty);
			break;
			case 3:
			DrawList(THREE, startx, starty);
			break;
			case 4:
```

```
                    DrawList(FOUR, startx, starty);
                    break;
                    case 5:
                    DrawList(FIVE, startx, starty);
                    break;
                    case 6:
                    DrawList(SIX, startx, starty);
                    break;
                    case 7:
                    DrawList(SEVEN, startx, starty);
                    break;
                    case 8:
                    DrawList(EIGHT, startx, starty);
                    break;
                    case 9:
                    DrawList(NINE, startx, starty);
                    break;


                }
                starty +=10;


        }
}

void deleteTextArea(){
        for(int x = 3; x<8 ; x++){
                for(int y = 0; y<64 ; y++){
                        chip_2();
                        SetYadrchip1(y);
                        delay();
                        SetXadrchip1(x*8);
                        delay();
                        rw_low();
                        Di_high();
                        displayLeft[x][y]= 0b00000000;
                        PORTB = displayLeft[x][y];
                        E_high();
                        E_low();
                        delay();
                }
        }
```

```
}

void DrawTextScore(char startx, char starty){ // skriver ut score;
	DrawList(S, startx, starty);
	starty+=10;
	DrawList(C, startx, starty);
	starty+=10;
	DrawList(O, startx, starty);
	starty+=10;
	DrawList(R, startx, starty);
	starty+=10;
	DrawList(E, startx, starty);
}

void AddToScore(char num){
	for(int i = 0; i <num; i++ ){
		score[4]++;
		if(score[4]>9){
			score[4] = 0;
			score[3]++;
		}
		if(score[3]>9){
			score[3] = 0;
			score[2]++;
		}
		if (score[2] >9){
			score[2] = 0;
			score[1]++;
		}
		if (score[1]>9){
			score[1] = 0;
			score[0]++;
		}

	}
}
void scoreadd(char x){
deleteTextArea();
DrawTextScore(30,8);
DrawScore(45,8);
AddToScore(x);
```

```
}


//SLUT NY
void clearrow(){
	Xclearrow = 61;
	char a;
	a = 0;
	for( int k = Xclearrow; k>0;  k--){
		for(char y = 2; y<62 ; y++){
			if(displayRight[k/8][y] & 1<<(k%8)){
				notclear = 0;
				Xclearrow = k;
				}else{
				notclear = 1;
				break;
			}
		}
		if(notclear == 0){
			for(int i = 2; i<61; i++){
				deleteispel(Xclearrow,i);
			}
			a++;
			movedownrow();
			scoreadd(a*pow(a,a));
			k++;
		}


	}

}

char nykollision(){
	char hit;
	hit = 0;

	char last1;
	char last2;
	char res;
```

```
        for(int i = offwidthNedL; i <(width-offwidthNedR); i++) {// int i = offwidthL; i
<(width-offwidthR); i++

                if(displayRight[(row+1)/8][col+i] & 1<<((row+1)%8)){  //
if(displayRight[(row+1)/8][col] & 1<<((row+1)%8))
                        clearrow();
                        return 0;
                }
        }
        if(offhightUL){
                for(int i = 0; i < offwidthNedL ; i++){

                        if(displayRight[(row+1-offhightUL)/8][col+i] & 1<<((row+1-offhightUL)%8)){
                                clearrow();
                                return 0;
                        }



                }
        }

        if(offhightUR){
                for(int i = (width-offwidthNedR); i < width ; i++){
                        last1 = displayRight[(row+1-offhightUR)/8][col+i];
                        last2 = 1<<((row+1-offhightUL)%8);
                        res = displayRight[(row+1-offhightUR)/8][col+i] &
1<<((row+1-offhightUR)%8);


                        if(displayRight[(row+1-offhightUR)/8][col+i] &
1<<((row+1-offhightUR)%8)){
                                clearrow();
                                return 0;


                        }

                }



        }
```

```
	return 1;

}
volatile uint8_t v;
char c;
char block;
char blockvar;
void rotate(){

	deleteblock(block);
	rotera++;
	rotera = ((rotera - 1) % 4) + 1;
	spelblock(block); //block
}









//------------------------------------------------------------------------------

int main(void)
{
	blockvar = 7;
	init();
	DisplayOn();
	cleardisplayRight();
	cleardisplayLeft();
	Board();

	Startmeny();

	//_delay_ms(1000);
	cleargame();
	sei();
	Press(40,6);
	LR(54,16);
char showscore;
```

```
row = 20; // star x
col = 30; // star y
char spelar;
int p;
char spela = 0;
char help;
rotera = 1;
char gameovernu = 0;

while(1){

	if(showscore){
		for (int i = 0 ; i<5; i ++)
		{
			score[i] = 0;
		}

		_delay_ms(3000);
		deleteTextArea();
		showscore = 0;
	}
	if(!showscore){
	Press(40,6);
	LR(54,16);
	}
	if(gameovernu){
		TextGame(25,8);
		TextOver(40,10);

	}
	if(PINA & 1<<3 && PINA &(1<<1) && !(PINA & 1<<2)){

		spelar = 1;
		help = 0;
		cleargame();
		gameovernu =1;
		DeletePress(40,6);
		DeleteLR(54,16);

	}

	while(spelar){
		while(1){
```

```
deleteTextArea();
DrawTextScore(30,8);
DrawScore(45,8);
AddToScore(1);
        block = (rand() % 7 +1);
        row = 9; // star x
        col = 30; // star y

        spelblock(block);       // random
        spela = nykollision();
        l = 0;
        while (spela){
                if(l ==1){
                        _delay_ms(10);
                }else{

                        _delay_ms(80);
                        cli();
                        if (v == 1){
                                rotate();
                                //Rotera
                                v = 0;
                        }
                        sei();
                }
                deleteblock(block);
                row++;
                nyknapp();
                spelblock(block);

                //Button


                spela = nykollision();



        }
        if(row < 20+hight || ((PINA & 1<<3) && (PINA & 1<<2) && (PINA &
1<<1))){
                cleargame();
                spelar = 0;
                help = 1;
```

```
                                        TextGame(25,8);
                                        TextOver(40,10);

                                        showscore = 1;
                                        break;
                                }
                                if(help){
                                        break;
                                }

                        }
                }

        }
        return 1;
}



ISR(PCINT2_vect){
        _delay_ms(10);
        if(PINC){
                if (roteratest()){
                        v = 1;
                }
        }

}
```