```
//Anwar Kashem, Wei-Feng Li
//EE 476 Final Project.
//May 10th, 2001
//plotter.c

#include <90s8515.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <delay.h>

#define dir 1
#define step_per_rotation 12
#define pen_up 1
#define pen_down 0
#define resolution 1
#define maxBuffer 25
#define calibrateMode 0
#define pointMode 1
#define charMode 2


//Serial Receive (UART_RXC ISR) & Command Parsing Variables

unsigned char r_index;       //current string index
unsigned char r_buffer[16];            //input string
unsigned char r_ready;       //flag for receive done
unsigned char r_char;                  //current character
unsigned int drawMode;                 //global var indicating the drawmode.

//Motor Control Variables
int x0,y0;                   //current_x, current_y;
int font;       //TEXT size
int pause;                   //pause between steps (in ms)

//initialization methods
void initialize(void);

//Serial communication, command parsing methods
void setKeybdInt(void);
void execCommand(void);

//motor/pen control methods
void x_plus(int rotation);
void x_minus(int rotation);
void y_plus(int rotation, int pen);
void y_minus(int rotation, int pen);

//line drawing methods
void move_x(int rotation, int x_dir);                   //move x by some amount in given direction
(x0,y0)=(x0 +/- rotation, y0)
void move_y(int rotation, int y_dir, int pen);//move y by some amount in given direction (x0,y0)=(x0, yo
+/- rotation)
void move_by(int x, int y, int pen);            //move to (x0,y0) = (x0+x , y0+y) with pen up/down
void move_to(int x, int y, int pen);
//void home(void);
```

```
//void newline(int size);                         //void draw_line(int x0, int y0, int x1, int y1, int
pen_up);

//Character drawing methods
void wr_sp(int size);                             //hit space bar once
void wr_A(int size);                              //write letter"A" with given size
void wr_B(int size);                              //write letter"B" with given size
void wr_C(int size);                              //write letter"C" with given size
void wr_D(int size);                              //write letter"D" with given size
void wr_E(int size);                              //write letter"E" with given size
void wr_F(int size);                              //write letter"F" with given size
void wr_G(int size);                              //write letter"G" with given size
void wr_H(int size);                              //write letter"H" with given size
void wr_I(int size);                              //write letter"I" with given size
void wr_J(int size);                              //write letter"J" with given size
void wr_K(int size);                              //write letter"K" with given size
void wr_L(int size);                              //write letter"L" with given size
void wr_M(int size);                              //write letter"M" with given size
void wr_N(int size);                              //write letter"N" with given size
void wr_O(int size);                              //write letter"O" with given size
void wr_P(int size);                              //write letter"P" with given size
void wr_Q(int size);                              //write letter"Q" with given size
void wr_R(int size);                              //write letter"R" with given size
void wr_S(int size);                              //write letter"S" with given size
void wr_T(int size);                              //write letter"T" with given size
void wr_U(int size);                              //write letter"U" with given size
void wr_V(int size);                              //write letter"V" with given size
void wr_W(int size);                              //write letter"W" with given size
void wr_X(int size);                              //write letter"X" with given size
void wr_Y(int size);                              //write letter"Y" with given size
void wr_Z(int size);                              //write letter"Z" with given size
void wr_0(int size);                              //write letter"0" with given size
void wr_1(int size);                              //write letter"1" with given size
void wr_2(int size);                              //write number"2" with given size
void wr_3(int size);                              //write number"3" with given size
void wr_4(int size);                              //write number"4" with given size
void wr_5(int size);                              //write number"5" with given size
void wr_6(int size);                              //write number"6" with given size
void wr_7(int size);                              //write number"7" with given size
void wr_8(int size);                              //write number"8" with given size
void wr_9(int size);                              //write number"9" with given size



//UART character-ready ISR
interrupt [UART_RXC] void uart_rec(void)
{
        r_char=UDR;    //get a char
        UDR=r_char;    //then print it
        //build the input string
        if (r_char != '\r') {
                if (r_index < maxBuffer) {
                        r_buffer[r_index++]=r_char;
                }
        }
        else
        {
                putchar('\n');        //use putchar to avoid overwrite
                r_buffer[r_index]=0x00;   //zero terminate
                r_ready=1;                //signal cmd processor
                UCR.7=0;                  //stop rec ISR
        }
}

//This function sets up the UART receive ISR.
void setKeybdInt(void)
{
  r_ready=0;
  r_index=0;
  UCR.7=1;
}


//Command Parser/Executor
void execCommand() {
        unsigned int i, x_1, y_1, x_2, y_2;
        unsigned char xfirst[5], yfirst[5], xnext[5], ynext[5];

        if(r_buffer[0] == '!') {                  //Set the DrawMode command in format "! mode"
                if(r_buffer[2] == 'p') {
                        drawMode = pointMode;
                        printf("Plotter is now in point-to-point drawing mode\r\n");
                } else
                if(r_buffer[2] == 'c') {
                        drawMode = charMode;
                        printf("Plotter is now in character drawing mode\r\n");
                } else
                if(r_buffer[2] == 'r') {
                        drawMode = calibrateMode;
                        printf("Plotter is now in recalibration drawing mode\r\n");
                }
        }
        else {
          if(drawMode == calibrateMode) { // -->format "XXXX,YYYY" X = xoffset, Y= yoffset
                strncpy(xfirst, &r_buffer[0], 4);
                strncpy(yfirst, &r_buffer[5], 4);
                move_by(atoi(xfirst), atoi(yfirst), pen_up);
                x0 = 0;
                y0 = 0;
          }
          if(drawMode == pointMode) {             //We are in point mode  -> format
"XXXX,YYYY,WWWW,ZZZZ"
                strncpy(xfirst, &r_buffer[0], 4);
                strncpy(yfirst, &r_buffer[5], 4);
                strncpy(xnext, &r_buffer[10], 4);
                strncpy(ynext, &r_buffer[15], 4);

                x_1 = atoi(xfirst);
                y_1 = atoi(yfirst);
                x_2 = atoi(xnext);
                y_2 = atoi(ynext);
                printf("%d %d %d %d\r\n", x_1, y_1, x_2, y_2);

                if(x_1 < 0) x_1 = 0;
                if(x_1 > 3000) x_1 = 3000;
                if(y_1 < 0) y_1 = 0;
                if(y_1 > 3000) y_1 = 3000;
                if(x_2 < 0) x_2 = 0;
                if(x_2 > 3000) x_2 = 3000;
                if(y_2 < 0) y_2 = 0;
                if(y_2 > 3000) y_2 = 3000;

                move_to(x_1, y_1, pen_up);
                move_to(x_2, y_2, pen_down);
      }
          if(drawMode == charMode) {        //We are in char mode
                 if(r_buffer[0] == 'f'){
                              font = atoi(&r_buffer[5]);
                              printf("Font size set to %d\r\n", font);
                 }
            if(r_buffer[0] == '>'){
             i = 1;
                       while(r_buffer[i] != 0){
                        if(r_buffer[i] == 'a')
                           wr_A(font);
                              if(r_buffer[i] == 'b')
                           wr_B(font);
                              if(r_buffer[i] == 'c')
                           wr_C(font);
                              if(r_buffer[i] == 'd')
                           wr_D(font);

                              if(r_buffer[i] == 'e')
                           wr_E(font);

                              if(r_buffer[i] == 'f')
                           wr_F(font);

                              if(r_buffer[i] == 'g')
                           wr_G(font);

                              if(r_buffer[i] == 'h')
                           wr_H(font);

                              if(r_buffer[i] == 'i')
                           wr_I(font);

                              if(r_buffer[i] == 'j')
                           wr_J(font);

                              if(r_buffer[i] == 'k')
                           wr_K(font);

                              if(r_buffer[i] == 'l')
                           wr_L(font);

                              if(r_buffer[i] == 'm')
```

```
wr_M(font);

      if(r_buffer[i] == 'n')
wr_N(font);

      if(r_buffer[i] == 'o')
wr_O(font);

      if(r_buffer[i] == 'p')
wr_P(font);

      if(r_buffer[i] == 'q')
wr_Q(font);

      if(r_buffer[i] == 'r')
wr_R(font);

      if(r_buffer[i] == 's')
wr_S(font);

      if(r_buffer[i] == 't')
wr_T(font);

      if(r_buffer[i] == 'u')
wr_U(font);
      if(r_buffer[i] == 'v')
wr_V(font);

      if(r_buffer[i] == 'w')
wr_W(font);

      if(r_buffer[i] == 'x')
wr_X(font);
      if(r_buffer[i] == 'y')
wr_Y(font);

      if(r_buffer[i] == 'z')
wr_Z(font);
                     if(r_buffer[i] == '0')
wr_0(font);

      if(r_buffer[i] == '1')
wr_1(font);

      if(r_buffer[i] == '2')
wr_2(font);

      if(r_buffer[i] == '3')
wr_3(font);

      if(r_buffer[i] == '4')
wr_4(font);

      if(r_buffer[i] == '5')
wr_5(font);
                             if(r_buffer[i] == '6')
```

```
                        wr_6(font);

                            if(r_buffer[i] == '7')
                        wr_7(font);

                            if(r_buffer[i] == '8')
                        wr_8(font);

                            if(r_buffer[i] == '9')
                        wr_9(font);
                            if(r_buffer[i] == ' ')
                        wr_sp(font);
                            i++;
                       }
          }
         }
       }
}

void x_plus(int rotation){
  int k;
  x0 = x0 + rotation;
  while(rotation>0){
    k = resolution;
    while(k>0){
      if(PORTA==0x9f){
        PORTA = 0xaf;
      }
      else if(PORTA==0xaf){
        PORTA = 0x6f;
      }
      else if(PORTA==0x6f){
        PORTA = 0x5f;
      }
      else if(PORTA==0x5f){
        PORTA = 0x9f;
      }
      k=k-1;
   }
   rotation=rotation-1;
 }
}


void x_minus(int rotation){
  int k;
  x0 = x0 - rotation;
  while(rotation>0){
    k = resolution;
    while(k>0){
      if(PORTA==0xaf){
        PORTA = 0x9f;
      }
      else if(PORTA==0x9f){
        PORTA = 0x5f;
      }
   else if(PORTA==0x5f){
    PORTA = 0x6f;
   }
   else if(PORTA==0x6f){
    PORTA = 0xaf;
   }
   k=k-1;
  }
  rotation=rotation-1;
 }
}


void y_plus(int rotation, int pen){
 int k;
 y0 = y0 + rotation;
 while(rotation>0){
  k = resolution;
  while(k>0){
   if(PORTC == 0x9f || PORTC == 0x90){
    if(pen == pen_down)  PORTC = 0xa0;
       if(pen == pen_up)        PORTC = 0xaf;
   }
   else if(PORTC == 0xaf || PORTC == 0xa0){
    if(pen == pen_down)  PORTC = 0x60;
       if(pen == pen_up)    PORTC = 0x6f;
   }
   else if(PORTC ==0x6f || PORTC == 0x60){
    if(pen == pen_down)   PORTC = 0x50;
    if(pen == pen_up)        PORTC = 0x5f;
   }
   else{
    if(pen == pen_down)   PORTC = 0x90;
              if(pen == pen_up)     PORTC = 0x9f;
   }
   k=k-1;
  }
  rotation=rotation-1;
 }
}

void y_minus(int rotation, int pen){
 int k;
 y0 = y0 - rotation;
 while(rotation>0){
  k = resolution;
  while(k>0){
   if(PORTC==0xaf || PORTC == 0xa0){
              if(pen == pen_down)           PORTC = 0x90;
              if(pen == pen_up)       PORTC = 0x9f;
   }
   else if(PORTC==0x9f || PORTC == 0x90){
    if(pen == pen_down)     PORTC = 0x50;
              if(pen == pen_up)              PORTC = 0x5f;
   }
   else if(PORTC==0x5f || PORTC == 0x50){
```

```
              if(pen == pen_down)                       PORTC = 0x60;
                            if(pen == pen_up)       PORTC = 0x6f;
      }
      else{
                            if(pen == pen_down)                     PORTC = 0xa0;
                            if(pen == pen_up)             PORTC = 0xaf;
      }
      k=k-1;
   }
   rotation=rotation-1;
 }
}

void move_x(int rotation, int x_dir){
  if(x_dir==1){
    x_plus(rotation);
  }
  else{
    x_minus(rotation);
  }
}

void move_y(int rotation, int y_dir, int pen){
  if(y_dir==1){
    y_plus(rotation, pen);
  }
  else{
    y_minus(rotation, pen);
  }
}

void move_by(int x1, int y1, int pen){
    int xt, yt;
     xt=x0+x1;
     yt=y0+y1;
     move_to(xt,yt,pen);
}

void move_to(int x1, int y1, int pen){

  int stepx, stepy,fraction,dx,dy;
  dy = y1 - y0;
  dx = x1 - x0;

  if (dy < 0) { dy = -dy;  stepy = -1; } else { stepy = 1; }
  if (dx < 0) { dx = -dx;  stepx = -1; } else { stepx = 1; }
  dy <<= 1;                                             // dy is now 2*dy
  dx <<= 1;                                             // dx is now 2*dx

  if (dx > dy) {
    fraction = dy - (dx >> 1);                               //2*dy - dx
    while (x0 != x1) {
      printf("           \r");
      if (fraction >= 0) {
        move_y(1,stepy, pen);                                              //y0 += stepy;
        fraction -= dx;                                                    //fraction -= 2*dx
      }
      move_x(1,stepx);                                          //x0 += stepx;
      fraction += dy;                                      //fraction -= 2*dy
    }
  }
  else {
    fraction = dx - (dy >> 1);
    while (y0 != y1) {
//      printf("y0 and y1 %d %d\r\n", x0, x1);
      printf("           \r");
      if (fraction >= 0) {
        move_x(1,stepx);                                        //x0 += stepx;
        fraction -= dy;
      }
      move_y(1,stepy, pen);                                     //y0 += stepy;
      fraction += dx;
    }
  }
}

void initialize(void){

  font=1;
  pause = 10;
  x0 = 0;
  y0 = 0;
  drawMode = pointMode;

  //Motor Control Port Setup
  DDRA=0xff;           //horizontal
  DDRC=0xff;                //veritical + pen control
  PORTA=0x9f;
  PORTC=0x9f;

  //Serial Stuff
  UBRR  = 25;               //Set Baud Rate = 9600
  UCR   = 0x18;             //Tx/Rx enable

  //Set up Keyboard Interrupt Routine
  setKeybdInt();

  //Set up interrupts
  #asm
          sei
  #endasm
}

void main(void)
{
  initialize();
  while(1) {
          if(r_ready == 1) {                //Execute the keyboard command if its ready
          execCommand();
          setKeybdInt();
    }
  }
```

```
}

//Character Drawing Functions

void wr_sp(int size){                                 //hit space bar once
  int four  =4;
  four<<= size;
  move_by (four,0,pen_up);
}

void wr_A(int size){                                  //write letter"A" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(two,             four,    pen_down);
  move_by(two,             -four,   pen_down);
  move_by(-three,          two,     pen_up);
  move_by(two,             0,       pen_down);
  move_by(three,           -two,    pen_up);

}
void wr_B(int size){                                  //write letter"B" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(four,            0,       pen_down);
  move_by(0,               four,    pen_down);
  move_by(-four,  0,       pen_down);
  move_by(one,             0,       pen_up);
  move_by(0,               -four,   pen_down);
  move_by(0,               two,     pen_up);
  move_by(two,             0,       pen_down);
  move_by(0,0,pen_up);
  move_by(three,  -two,    pen_up);
}

void wr_C(int size){                                  //write letter"C" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(three,  three,   pen_up);
  move_by(0,               one,      pen_down);
  move_by(-three,0,        pen_down);
  move_by(0,               -four,    pen_down);
  move_by(three,  0,       pen_down);
  move_by(0,               one,      pen_down);
  move_by(three,  -one,    pen_up);
}

void wr_D(int size){                                     //write letter"D" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,             four,     pen_up);
  move_by(0,               -four,    pen_down);
  move_by(-one,            four,     pen_up);
  move_by(four,            0,                  pen_down);
  move_by(0,               -four,    pen_down);
  move_by(-four,  0,                 pen_up);
  move_by(four,            0,                  pen_down);
  move_by(two,             0,                  pen_up);
}

void wr_E(int size){                                     //write letter"E" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(four,            four,     pen_up);
  move_by(-three,0,                  pen_down);
  move_by(0,               -four,    pen_down);
  move_by(three,  0,                 pen_down);
  move_by(-three,two,      pen_up);
  move_by(two,             0,                  pen_down);
  move_by(three,  -two,    pen_up);
}

void wr_F(int size){                                     //write letter"F" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,             0,                  pen_up);
  move_by(0,               four,     pen_down);
  move_by(3,                  0,                  pen_down);
  move_by(-three,-two,        pen_up);
  move_by(two,                0,                  pen_down);
  move_by(three, -two,        pen_up);
}

void wr_G(int size){                                        //write letter"G" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(three, three,       pen_up);
  move_by(-three,0,                     pen_down);
  move_by(0,                  -four,    pen_down);
  move_by(three, 0,                     pen_down);
  move_by(0,                  two,      pen_down);
  move_by(-one,               0,                  pen_down);
  move_by(four,               -two,     pen_up);
}

void wr_H(int size){                                        //write letter"H" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,                0,                  pen_up);
  move_by(0,                  four,     pen_down);
  move_by(three, 0,                     pen_up);
  move_by(0,                  -four,    pen_down);
  move_by(-three,two,         pen_up);
  move_by(three, 0,                     pen_down);
  move_by(two,                -two,     pen_up);
}
void wr_I(int size){                                        //write letter"I" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,                0,                  pen_up);
  move_by(two,                0,                  pen_down);
  move_by(-two,               four,     pen_up);
  move_by(two,                0,                  pen_down);
  move_by(-one,               0,                  pen_up);
  move_by(0,                  -four,    pen_down);
  move_by(four,            0,                 pen_up);
}

void wr_J(int size){                                  //write letter"J" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,             one,     pen_up);
  move_by(0,               -one,    pen_down);
  move_by(two,             0,                 pen_down);
  move_by(0,               four,    pen_down);
  move_by(-one,            one,     pen_up);
  move_by(two,             0,                 pen_down);
  move_by(two,             -four,   pen_up);
}

void wr_K(int size){                                  //write letter"K" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,             0,                 pen_up);
  move_by(0,               four,    pen_down);
  move_by(three,  0,                pen_up);
  move_by(-three,-two,     pen_down);
  move_by(three,  -two,    pen_down);
  move_by(two,             0,                 pen_up);
}

void wr_L(int size){                                  //write letter"L" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,             four,    pen_up);
  move_by(0,               -four,   pen_down);
  move_by(three,  0,                pen_down);
  move_by(two,             0,                 pen_up);
}

void wr_M(int size){                                  //write letter"M" with given size
  int one   = 1;
  int two   = 2;
```

```
 int three = 3;
 int four  = 4;
 one     <<= size;
 two     <<= size;
 three   <<= size;
 four    <<= size;
 move_by(0,              four,   pen_down);
 move_by(two,            0,              pen_down);
 move_by(0,              -two,   pen_down);
 move_by(0,              two,    pen_up);
 move_by(two,            0,              pen_down);
 move_by(0,              -four,  pen_down);
 move_by(two,            0,              pen_up);
}

void wr_N(int size){                             //write letter"N" with given size
 int one   = 1;
 int two   = 2;
 int three = 3;
 int four  = 4;
 one     <<= size;
 two     <<= size;
 three   <<= size;
 four    <<= size;
 move_by(0,              four,   pen_down);
 move_by(four,           -four,  pen_down);
 move_by(0,              four,   pen_down);
 move_by(two,            -four,  pen_up);
}

void wr_O(int size){                             //write letter"O" with given size
 int one   = 1;
 int two   = 2;
 int three = 3;
 int four  = 4;
 one     <<= size;
 two     <<= size;
 three   <<= size;
 four    <<= size;
 move_by(0,              four,   pen_down);
 move_by(three,  0,              pen_down);
 move_by(0,              -four,  pen_down);
 move_by(-three,0,               pen_up);
 move_by(three,  0,              pen_down);
 move_by(three,  0,              pen_up);
}

void wr_P(int size){                             //write letter"P" with given size
 int one   = 1;
 int two   = 2;
 int three = 3;
 int four  = 4;
 int five  = 5;
 one     <<= size;
 two     <<= size;
 three   <<= size;
```

```
  four    <<= size;
  five    <<= size;
  move_by(one,          0,              pen_up);
  move_by(0,            four,   pen_down);
  move_by(two,          0,      pen_down);
  move_by(0,            -two,   pen_down);
  move_by(-two,         0,              pen_down);
  move_by(five,         -two,   pen_up);
}

void wr_Q(int size){                            //write letter"Q" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  wr_O(size);
  move_by(-four,  one,  pen_up);
  move_by(two,          -one,   pen_down);
  move_by(two,          0,              pen_up);
}

void wr_R(int size){                            //write letter"R" with given size
  int two   = 2;
  int three = 3;
  int four  = 4;
  int five  = 5;
  two     <<=size;
  three   <<=size;
  four    <<=size;
  five    <<=size;
  wr_P(size);
  move_by(-five,  two,  pen_up);
  move_by(two,          -two,   pen_down);
  move_by(three,  0,            pen_up);
}
void wr_S(int size){                            //write letter"S" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,          0,              pen_up);
  move_by(three,  0,            pen_down);
  move_by(0,            two,    pen_down);
  move_by(-three,0,             pen_down);
  move_by(0,            two,    pen_down);
  move_by(three,  0,            pen_down);
  move_by(two,          -four,  pen_up);
}
```

```
void wr_T(int size){                                    //write letter"T" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(0,-three,pen_up);
  move_by(0,one,pen_down);
  move_by(four,0,pen_down);
  move_by(0,-one,pen_down);
  move_by(-two,one,pen_up);
  move_by(0,-four,pen_down);
  move_by(four,0,pen_up);
}
void wr_U(int size){                                    //write letter"U" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(0,four,pen_up);
  move_by(0,-four,pen_down);
  move_by(three,0,pen_down);
  move_by(0,four,pen_down);
  move_by(three,-four,pen_up);
}
void wr_V(int size){                                    //write letter"V" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(0,four,pen_up);
  move_by(0,-four,pen_down);
  move_by(three,four,pen_down);
  move_by(two,-four,pen_up);
}

void wr_W(int size){                                    //write letter"W" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(0,four,pen_up);
  move_by(0,-four,pen_down);
  move_by(two,0,pen_down);
  move_by(0,two,pen_up);
  move_by(0,-two,pen_down);
  move_by(two,0,pen_down);
  move_by(0,four,pen_down);
  move_by(two,-four,pen_up);
}
void wr_X(int size){                                //write letter"X" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;;
  move_by(four,four,pen_down);
  move_by(-four,0,pen_up);
  move_by(four,-four,pen_down);
  move_by(two,0,pen_up);
}

void wr_Y(int size){                                //write letter"Y" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(two,0,pen_up);
  move_by(0,two,pen_down);
  move_by(-two,two,pen_down);
  move_by(four,0,pen_up);
  move_by(-two,-two,pen_down);
  move_by(four,-two,pen_up);
}
void wr_Z(int size){                                //write letter"Z" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(0,three,pen_up);
  move_by(0,one,pen_down);
  move_by(four,0,pen_down);
  move_by(-four,-four,pen_down);
  move_by(four,0,pen_down);
  move_by(0,one,pen_down);
  move_by(two,-one,pen_up);
```

```
}
void wr_0(int size){                                //write letter"0" with given size
  int one = 1;
  int two = 2;
  int three = 3;
  int four = 4;
  one<<=size;
  two<<=size;
  three<<=size;
  four<<=size;
  move_by(one,0,pen_up);
  move_by(0,four,pen_down);
  move_by(two,0,pen_down);
  move_by(0,-four,pen_down);
  move_by(-two,0,pen_down);
  move_by(-one,0,pen_up);
  move_by(four,four,pen_down);
  move_by(two,-four,pen_up);
}
void wr_1(int size){                                //write number "1" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,three,pen_up);
  move_by(one,one,pen_down);
  move_by(0,-four,pen_down);
  move_by(-one,0,pen_up);
  move_by(two,0,pen_down);
  move_by(three,0,pen_up);
}
void wr_2(int size){                                //write number"2" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,four,pen_up);
  move_by(two,0,pen_down);
  move_by(0,-two,pen_down);
  move_by(-two,0,pen_down);
  move_by(0,-two,pen_down);
  move_by(two,0,pen_down);
  move_by(three,0,pen_up);
}
void wr_3(int size){                                //write number"3" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,three,pen_up);
  move_by(two,0,pen_down);
  move_by(0,-four,pen_down);
  move_by(-two,0,pen_down);
  move_by(0,two,pen_up);
  move_by(two,0,pen_down);
  move_by(three,-two,pen_up);
}
void wr_4(int size){                                   //write number"4" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(three,four,pen_up);
  move_by(-two,-three,pen_down);
  move_by(three,0,pen_down);
  move_by(-one,three,pen_up);
  move_by(0,-four,pen_down);
  move_by(three,0,pen_up);
}
void wr_5(int size){                                   //write number"5" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,0,pen_up);
  move_by(two,0,pen_down);
  move_by(0,two,pen_down);
  move_by(-two,0,pen_down);
  move_by(0,two,pen_down);
  move_by(two,0,pen_down);
  move_by(three,-four,pen_up);
}
void wr_6(int size){                                   //write number"6" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,two,pen_up);
  move_by(two,0,pen_down);
  move_by(0,-two,pen_down);
  move_by(-two,0,pen_down);
  move_by(0,four,pen_down);
  move_by(two,0,pen_down);
  move_by(three,-four,pen_up);
}
void wr_7(int size){                                    //write number"7" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;;
  move_by(one,three,pen_up);
  move_by(0,one,pen_down);
  move_by(two,0,pen_down);
  move_by(0,-four,pen_down);
  move_by(three,0,pen_up);
}
void wr_8(int size){                                    //write number"8" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,two,pen_up);
  move_by(two,0,pen_down);
  move_by(0,two,pen_down);
  move_by(-two,0,pen_down);
  move_by(0,-four,pen_down);
  move_by(two,0,pen_down);
  move_by(0,two,pen_down);
  move_by(three,-two,pen_up);
}
void wr_9(int size){                                    //write number"9" with given size
  int one   = 1;
  int two   = 2;
  int three = 3;
  int four  = 4;
  one     <<= size;
  two     <<= size;
  three   <<= size;
  four    <<= size;
  move_by(one,0,pen_up);
  move_by(two,0,pen_down);
  move_by(0,four,pen_down);
  move_by(-two,0,pen_down);
  move_by(0,-two,pen_down);
  move_by(two,0,pen_down);
  move_by(three,-two,pen_up);
```

```
}
```