```
/*Combines the shift left/right operations with AND, OR, Exclusive-OR and NOT to set, clear and toggle
individual bits of registers. Access to all hardware components (i.e. Timer, USART, WDT, IO) is via registers.
Therefore these functions enable us to control and interrogate these components.
 */

#include "Shift_bits_header.h"

int main (void) {
  char op_code;
  char digits[8];
    char keypress;
  unsigned char lfsr;

int test = 0;
  char PRN_counter = 0;

 setup_HW;
for(int m = 0; m <=7; m++)digits[m] = 0;
   sei();
 lfsr = (PRN_8bit_GEN ());

  String_to_PC_Basic
  ("\r\n\r\nSelect mode 1 to 6? Then AK to continue or x to exit (when allowed)\r\n");

  while (1)
  { String_to_PC_Basic("Mode?\t");

    op_code = waitforkeypress_Basic();

switch (op_code)
{ case '1':   String_to_PC_Basic("shift number left\r\n"); break;
  case '2':   String_to_PC_Basic("shift number right\r\n"); break;
  case '3':   String_to_PC_Basic("set a bit\r\n"); break;
  case '4':   String_to_PC_Basic("clear a bit\r\n"); break;
  case '5':   String_to_PC_Basic("toggle a bit\r\n"); break;
  case '6':   String_to_PC_Basic("test a bit\r\n"); break;
  default:    newline_Basic(); continue; }

digits[1] = 0;
digits[2] = 0;

do
{ if (digits[1] == 0)
  {  digits[0] = lfsr;
    digits[2] = digits[0];

    I2C_Tx_BWops(digits);
    lfsr = PRN_8bit_GEN ();
  }

  digits[2] = logical_op(digits[1], digits[0], op_code);

  I2C_Tx_BWops(digits);
  keypress = waitforkeypress_Basic();

  if (op_code < '3')
  { if (++digits[1] == 9)digits[1] = 0;
  }
  else
  { if (!(digits[1]))digits[1] = 1;
  else digits[1] = (((byte)digits[1] << 1) % 256);}

} while (keypress != 'x');}}


/*****************************************************************************************/
char logical_op(char X, char Y, char op_code) {
  char result = 0;
  switch (op_code) {

    case '1': result = Y << X; break;
    case '2': result = (byte)Y >> X; break;
    case '3': result = Y | X; break;
    case '4': result = Y & (~(X)); break;
    case '5': result = Y ^ X; break;
    case '6': result = Y & X; break;}
  return result;}



/*****************************************************************/
```