C program for the PIC16F877 with a 4 MHz oscillator to display a continuously incrementing count on an LCD. A switch is connected to the External Interrupt pin INT of the PIC Microcontroller. When this switch is pressed, the microcontroller is interrupted and the ISR is executed. On each interrupt, increment the displayed value and clear the interrupt flag in software. Ensure that the LCD display is ON with a blinking cursor, the entire display is shifted to the right, and the cursor starts writing at the beginning of the second line. The counting should run indefinitely

```
#include <xc.h>

#define _XTAL_FREQ 4000000

#define RS RC0
#define EN RC1


void LCD_CMD(unsigned char);
void LCD_DATA(unsigned char);
void LCD_INIT(void);
void LCD_PRINT_NUMBER(unsigned int);
void LCD_DELAY(void);


unsigned int count = 0;


void __interrupt() ISR(void)
{
    if (INTF==1)

        count++;
        LCD_CMD(0xC0);
        LCD_PRINT_NUMBER(count);

        INTF = 0;
    }
}

void main(void)
{
    TRISD = 0x00;
    TRISC = 0x00;
    TRISB0 = 1;


    INTE = 1;
    GIE = 1;

    LCD_INIT();

    LCD_CMD(0xC0);
    LCD_PRINT_NUMBER(count);

    while (1)
    {

    }
}
```

```
void LCD_CMD(unsigned char cmd)
  {
      RS = 0;
      PORTD = cmd;
      EN = 1;
      __delay_ms(2);
      EN = 0;
      LCD_DELAY();
  }

void LCD_DATA(unsigned char data)
  {
      RS = 1;
      PORTD = data;
      EN = 1;
      __delay_ms(2);
      EN = 0;
      LCD_DELAY();
  }

 void LCD_INIT(void)
 {
     __delay_ms(20);

     LCD_CMD(0x38);
     LCD_CMD(0x0D);
     LCD_CMD(0x06);
     LCD_CMD(0x01);
     LCD_CMD(0x1C);
     __delay_ms(5);
 }

 void LCD_PRINT_NUMBER(unsigned int num)
 {
     unsigned int thousands = (num / 1000);
     unsigned int hundreds  = (num / 100) % 10;
     unsigned int tens      = (num / 10) % 10;
     unsigned int ones      = num % 10;

     LCD_DATA(thousands + '0');
     LCD_DATA(hundreds + '0');
     LCD_DATA(tens + '0');
         LCD_DATA(ones + '0');
    }

    void LCD_DELAY(void)
     {
       for (unsigned int i = 0; i < 300; i++);
     }
```

Design a C program for the PIC16F877 with a 4 MHz oscillator to display a continuously incrementing count on an LCD. Use the Timer overflow interrupt with the largest pre-scaler to detect overflows. On each interrupt, increment the displayed value and clear the interrupt flag in software. Ensure that the LCD display is ON with a blinking cursor, the entire display is shifted to the right, and the cursor starts writing at the beginning of the second line. The counting should run indefinitely.

```
#include <xc.h>

#define _XTAL_FREQ 4000000
#define RS RC0
#define EN RC1

void LCD_CMD(unsigned char);
void LCD_DATA(unsigned char);
void LCD_INIT(void);
void LCD_PRINT_NUM(unsigned int);
void LCD_DELAY(void);

volatile unsigned int count = 0;

void __interrupt() ISR(void)
{
    if (T0IF==1)          // If Timer0 overflow occurred
    {
        count++;       // Increment counter

        LCD_CMD(0xC0); // Move cursor to 2nd line, position 1
        LCD_PRINT_NUM(count);

        T0IF = 0;
    }
}

void main(void)
{
    TRISD = 0x00;      // LCD Data port
    TRISC = 0x00;      // LCD Control pins

    OPTION_REG = 0b00000111;  // Prescaler 1:256, TMR0 uses internal clock
                              // PSA=0, PS2:PS0 = 111

    T0IE = 1;          // Enable Timer0 interrupt
    GIE = 1;           // Enable global interrupts

    LCD_INIT();        // Initialize the LCD

    LCD_CMD(0xC0);     // Start at beginning of line 2
    LCD_PRINT_NUM(count);

    while(1)
   {
    }
}
```

```
void LCD_CMD(unsigned char cmd)
 {
     RS = 0;
     PORTD = cmd;
     EN = 1;
     __delay_ms(2);
     EN = 0;
     LCD_DELAY();
 }
 void LCD_DATA(unsigned char data)
 {
     RS = 1;
     PORTD = data;
     EN = 1;
     __delay_ms(2);
     EN = 0;
     LCD_DELAY();
 }
 void LCD_INIT(void)
 {
     __delay_ms(20);

     LCD_CMD(0x38);
     LCD_CMD(0x0D);
     LCD_CMD(0x06);
     LCD_CMD(0x01);
     __delay_ms(2);

     LCD_CMD(0x1C);
 }

 void LCD_PRINT_NUM(unsigned int num)
 {
     unsigned int thousands = num / 1000;
     unsigned int hundreds  = (num / 100) % 10;
     unsigned int tens      = (num / 10) % 10;
     unsigned int ones      = num % 10;

     LCD_DATA(thousands + '0');
     LCD_DATA(hundreds  + '0');
     LCD_DATA(tens      + '0');
     LCD_DATA(ones      + '0');
 }



 void LCD_DELAY(void)
 {
     for (unsigned int i = 0; i < 300; i++);
 }
```