```
/* This program demonstrates the use of T0 interrupt. The code will count the
number of T0 timer overflows that occur while a slide switch is in Off position
Some editing is requited prior to running the code. Fill in the indicated blanks.
*/

#include <c8051_SDCC.h> // include files. You need to include stdio and c8051_SCDD.
#include <stdio.h>      // Add lines as needed

//------------------------------------------------------------
// Function PROTOTYPES
//------------------------------------------------------------
void T0_ISR (void) __interrupt 1; // Function Prototype for Interrupt Service Routine
void Port_Init(void);  // Initialize ports for input and output
void Timer_Init(void); // Initialize Timer 0
void Interrupt_Init(void);
void Counter_off(void);
void Counter_on(void); //

//------------------------------------------------------------
// Global variables
//------------------------------------------------------------

// one end of bicolor LED0 is associated with Port 3 Pin 3
__sbit __at 0xB3 Biled1;
// other end of bicolor LED0 is associated with Port 3 Pin 4
__sbit __at 0xB4 Biled2;
__sbit __at 0xA0 SW; // Slide Switch associated with Port 2 Pin 0

unsigned int Counts = 0;
```

```
//***************
void main(void)
{
    Sys_Init();    // System Initialization Always do this first.
    putchar(' ');  // line added to allow printf statements
    Port_Init();   // Initialize port 2 and 3
    Timer_Init();  // Initialize Timer 0
    Interrupt_Init();

    printf("Start\r\n");
    while (1)
    {
        Counter_off();
        Counter_on();
    }
}


void Port_Init(void)
{

    // Port 3
    P3MDOUT |= 0x18; // set output pins P3.3 and P3.4 in push-pull mode

    // Port 2
    P2MDOUT &= ~0x01; // set input pin P2.0 in open drain mode
    P2 |= 0x01; // set input pin P2.0 to high impedance state
}
```

```
void Interrupt_Init(void)
{
   IE |= 0x82; //enable Timer0 interrupts by setting the appropriate bit in the SFR
   EA = 1;  //enable all interrupts using an existing sbit label
}


void Timer_Init(void)
{
   CKCON &= ~0x08;  // Make T1 intact and T0 use SYSCLK/12
   TMOD &= 0xF0;   // Clear the 4 least significant bits
   TMOD |= 0x00;   // Leave T1 intact and set T0 mode 13bit
   TR0 = 0;    // Stop Timer0
// 2 ways to clear 16-bit T0 counter: use a single command for all 16 bits
   TMR0 = 0;    // Clear both bytes of T0
// or use 2 commands for low and high bytes separately
   //TL0 _____________;    // Clear low byte of register T0
   //TH0 _____________;    // Clear high byte of register T0
}

void T0_ISR ( void ) __interrupt 1 //Interrupt service routine
{
   TF0 = 0;  // clear interrupt request (not required - cleared automatically by hardware)
   Counts++; // increment overflow counter
}

void Counter_off(void) // turn the BILED off and stop the counter
{
   TR0 = 0;    // turn off the counter
   Counts = 0; // reset counts to 0
   Biled1 = 0; //on
   Biled2 = 0; //on
   TL0 = 0x00;
   TH0 = 0x00; // initialize the Timer to a 0 start value
   while(SW);  // while the switch is off, wait
}

void Counter_on(void) // turn the BILED on and count how long it the switch is on
{
   Biled1 = 1;
   Biled2 = 0;
   TR0 = 1;    // start the counter
   while(!SW); // while the switch is on, wait
   printf("\rNumber of Overflows = %d\n", Counts);
}
```