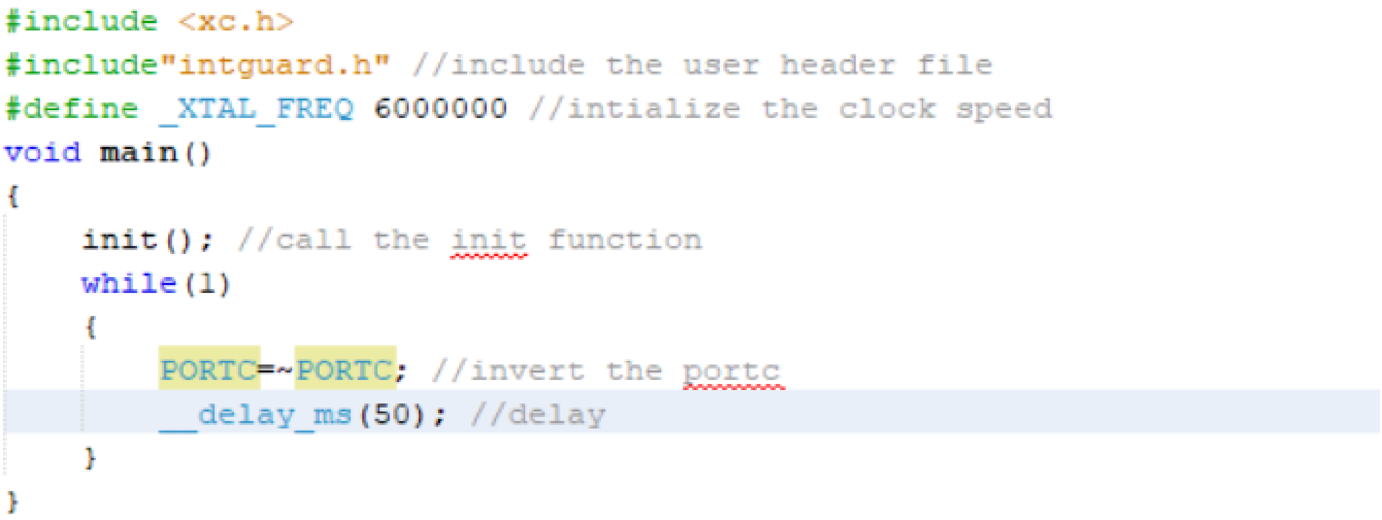
INTERRUPT

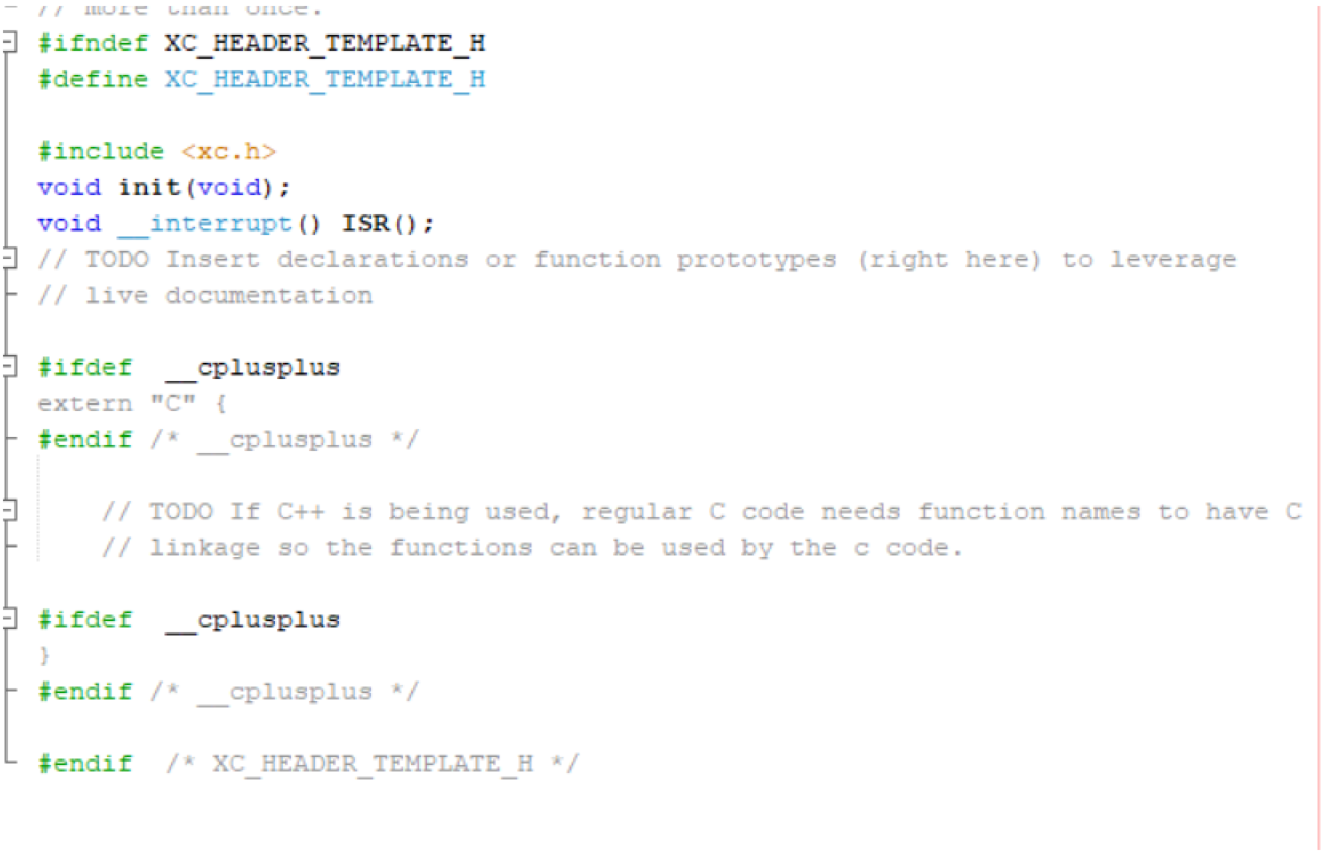
Task program

MAIN PROGRAM:-



```c
#include <xc.h>
#include"intguard.h" //include the user header file
#define _XTAL_FREQ 6000000 //intialize the clock speed
void main()
{
    init(); //call the init function
    while(1)
    {
        PORTC=~PORTC; //invert the portc
        __delay_ms(50); //delay
    }
}
```

GUARD PROGRAM:-

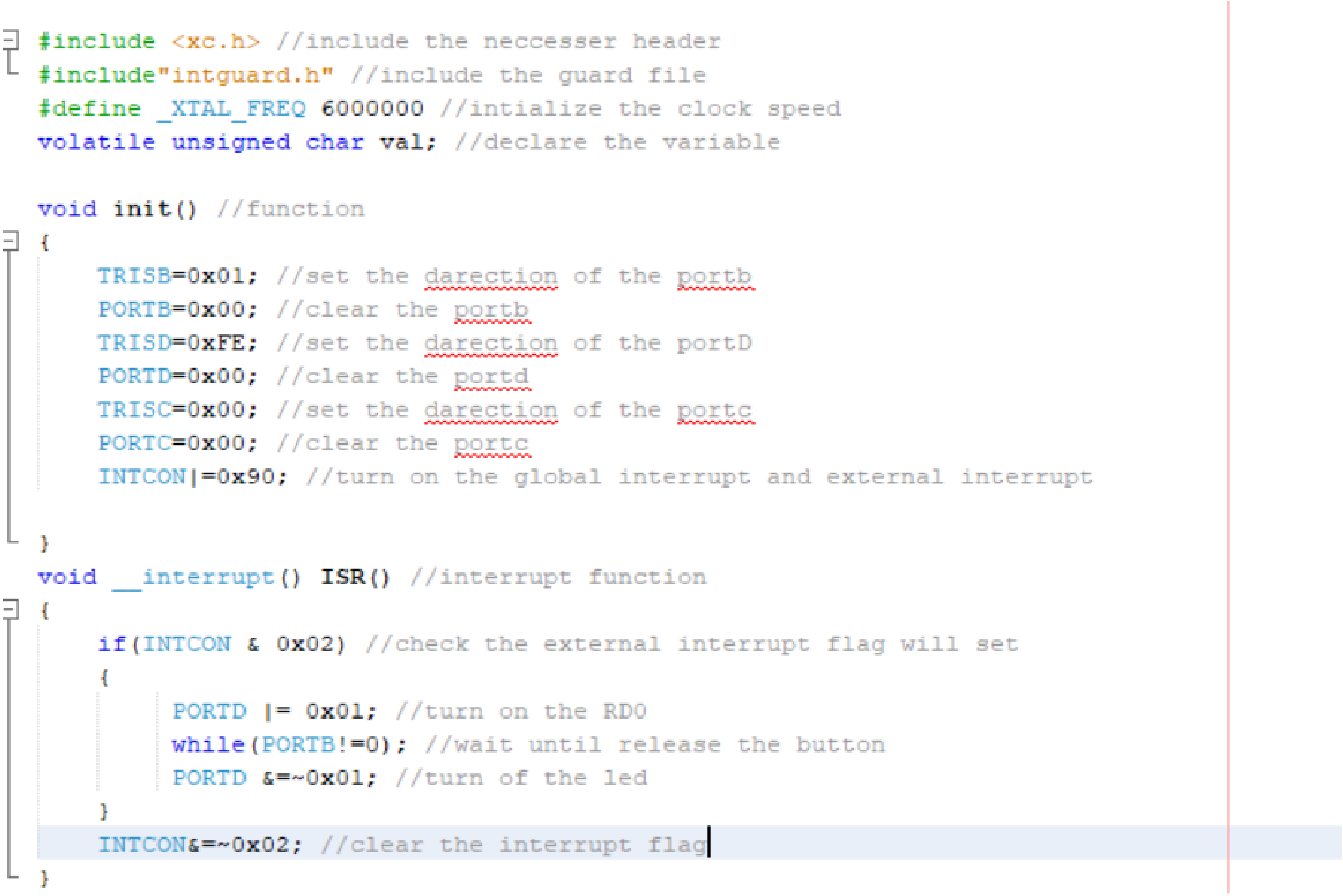


```c
// more than once.
#ifndef XC_HEADER_TEMPLATE_H
#define XC_HEADER_TEMPLATE_H

#include <xc.h>
void init(void);
void __interrupt() ISR();
// TODO Insert declarations or function prototypes (right here) to leverage
// live documentation

#ifdef  __cplusplus
extern "C" {
#endif /* __cplusplus */

    // TODO If C++ is being used, regular C code needs function names to have C
    // linkage so the functions can be used by the c code.

#ifdef  __cplusplus
}
#endif /* __cplusplus */

#endif  /* XC_HEADER_TEMPLATE_H */
```

# INTERRUPT

Task program

FUNCTION PROGRAM:



```c
#include <xc.h> //include the neccesser header
#include"intguard.h" //include the guard file
#define _XTAL_FREQ 6000000 //intialize the clock speed
volatile unsigned char val; //declare the variable

void init() //function
{
    TRISB=0x01; //set the darection of the portb
    PORTB=0x00; //clear the portb
    TRISD=0xFE; //set the darection of the portD
    PORTD=0x00; //clear the portd
    TRISC=0x00; //set the darection of the portc
    PORTC=0x00; //clear the portc
    INTCON|=0x90; //turn on the global interrupt and external interrupt

}
void __interrupt() ISR() //interrupt function
{
    if(INTCON & 0x02) //check the external interrupt flag will set
    {
        PORTD |= 0x01; //turn on the RD0
        while(PORTB!=0); //wait until release the button
        PORTD &=~0x01; //turn of the led
    }
    INTCON&=~0x02; //clear the interrupt flag
}
```