**Ejercicio 1:**

#define \_XTAL\_FREQ 4000000

#include <xc.h>

#include <stdint.h>

void setup(void);

void main(void) {

setup();

while(1){

PORTA++;

\_\_delay\_ms(1000);

}

return;

}

void setup(void){

ANSEL = 0x00;

ANSELH = 0x00;

OSCCONbits.IRCF = 0b0110; //4MHz

OSCCONbits.SCS = 1;

TRISA = 0x00;

PORTA = 0x00;

return;

}

**Ejercicio 2:**

#define \_XTAL\_FREQ 4000000

#define bit 1

#include <xc.h>

#include <stdint.h>

void setup(void);

void main(void) {

setup();

while(1){

for(uint8\_t i=0; i<=7;++i){

PORTA = bit <<i;

\_\_delay\_ms(1000);

}

}

return;

}

void setup(void){

ANSEL = 0x00;

ANSELH = 0x00;

OSCCONbits.IRCF = 0b0110; //4MHz

OSCCONbits.SCS = 1;

TRISA = 0x00;

PORTA = 0x00;

return;

}

**Ejercicio 3:**

#define \_XTAL\_FREQ 4000000

#include <xc.h>

#include <stdint.h>

void setup(void);

void main(void) {

setup();

int valores[7] = {1,127,95,36,15,253,63};

while(1){

for(uint8\_t i=0; i<7;++i){

PORTC = valores[i];

\_\_delay\_ms(1000);

}

}

return;

}

void setup(void){

ANSEL = 0x00;

ANSELH = 0x00;

OSCCONbits.IRCF = 0b0110; //4MHz

OSCCONbits.SCS = 1;

TRISC = 0x00;

PORTC = 0x00;

return;

}

**Ejercicio 4:**

#define \_XTAL\_FREQ 4000000

#include <xc.h>

#include <stdint.h>

uint8\_t contador=0;

uint8\_t comparar=0;

const char valores[10] = {19,7,3,10,15,1,6,18,20,14};

void setup(void);

void \_\_interrupt() isr (void){

if(INTCONbits.RBIF){ // Fue interrupción del PORTB

if (!PORTBbits.RB0) // Verificamos si fue RB0 quien generó la interrupción

++contador; // Incremento del PORTC

if(!PORTBbits.RB1)

--contador;

INTCONbits.RBIF = 0; // Limpiamos bandera de interrupción

}

return;

}

void main(void) {

setup();

while(1){

PORTA = contador;

//contador = (contador == 21)? 0:contador;

//contador = (contador == 255)? 20:contador;

if(contador==21)

contador=0;

if(contador==255)

contador=20;

for(uint8\_t i=0; i<7;++i){

if(contador == valores[i])

comparar = i;

}

PORTC = comparar;

}

return;

}

void setup(void){

ANSEL = 0x00;

ANSELH = 0x00;

OSCCONbits.IRCF = 0b0110; //4MHz

OSCCONbits.SCS = 1;

TRISA = 0x00;

TRISB = 0x00;

TRISC = 0x00;

PORTA = 0x00;

PORTB = 0x00;

PORTC = 0x00;

TRISBbits.TRISB0 = 1;

TRISBbits.TRISB1 = 1;

OPTION\_REGbits.nRBPU = 0;

WPUBbits.WPUB0 = 1;

WPUBbits.WPUB1 = 1;

IOCBbits.IOCB0 = 1;

IOCBbits.IOCB1 = 1;

INTCONbits.GIE = 1;

INTCONbits.RBIE = 1;

INTCONbits.RBIF = 0;

return;

}