

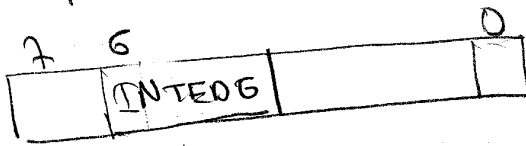
(1)

Ex³

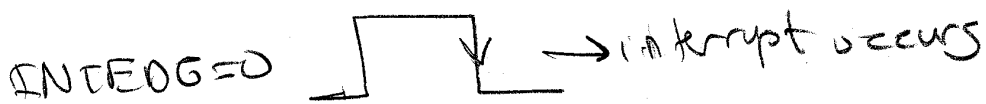
Write a program using interrupt subroutine such that when a falling pulse at RB0/INT occurs RBL is toggled (i.e, if it is ON it is turned OFF and vice-versa)

Sln.

OPTION REG



↓
interrupt edge bit



list p=16f84A

include "p16f84A.INC"

-config -WDT-OFF & PWRTE-OFF & XT-Osc

org 0x000;

goto main

org 0x004;

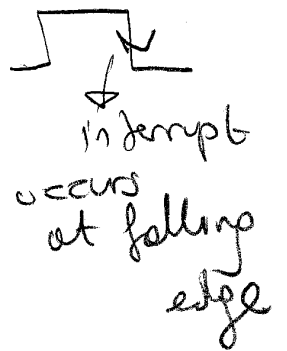
goto interrupt-subroutine;

2

```

main
clr PORTB,
banksel TRISB; goto Bank1 → bsf STATUS, RA0
movlw 0x00; OR elrw
movwf TRISB; ) OR clr TRISB
bcf OPTION_REG, INTE0G;
banksel PORTB; → bcf STATUS, RPO
bcf INTCON, INTF;

```



```

bsf INTCON, GIE → clear interrupt flag
bsf INTCON, INTE → Enable all interrupts.
                    → Enable interrupts at RB0/INT
loop goto loop

```

interrupt_subroutine

```

bcf INTCON, INTE; → disable other interrupts from RB0/INT while inside interrupt subroutine.
movlw b'00000010';
xorwf PORTB, F
bcf INTCON, INTF;
bsf INTCON, INTE; → toggle RBL
ret frc; → clear interrupt flag
            → Enable new interrupts
            → Return from interrupt subroutine.

```

③
Ex

Trace the following program

```
list p16f84A
include "p16f84A.INC"
config -WDT-OFF & -XT-OSC
org 0x000;
goto main;
org 0x004;
btfss PORTA, RA1;
goto intsb1;
btfss PORTA, RA2;
goto intsb2;
goto intsb3;
```

```
main
  clrf PORTB;
  bsf STATUS, RPO;
  clrf PORTB;
  movlw 0xFF;
  movwf TRISA;
  bcf OPTION-REG, INTEDG;
  bcf STATUS, RPO;
  bcf INTCON, INTF;
  bsf INTCON, GIE;
  bsf INTCON, INTE;
loop goto loop
```

```
intsb1
  bcf INTCON, INTE;
  bsf PORTB, RB1;
  bcf INTCON, INTF;
  bsf INTCON, INTE;
  retfie;
```

④

intsbr2

```
bcf INTCON, INTE,  
bsf PORTB, RB2,  
bcf INTCON, INTF,  
bsf INTCON, INTE,  
retfie;
```

intsbr3

```
bcf INTCON, INTE,  
bsf PORTB, RB3,  
bcf INTCON, INTF,  
bsf INTCON, INTE,  
retfie;
```