For AVR :

DDR A : identifies the pins of port A

0 = input 1 = output

**Identifying pins using code :**

* DDRA = 0x00; // all pins are input
* (\*(uint8\_t \*) (0x24)) = 0x00; // pointer to address of port A pins in MC to set all pins as input
  + - * + 0xff // port A pins are all output pins
* using bit masking:

if its wanted to set pin 5 in port A = 1 : DDRA |= 1<<5;

* using MACROs:
  + #define set\_bit(reg,bit) (reg | = 1 << bit);

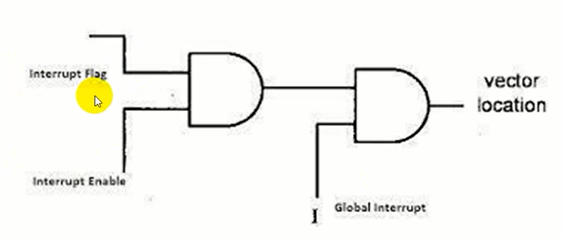
// set pin (bit) in DDR (port) = 1 🡪 bit is output pin

* + #define reset\_bit(reg,bit) ( reg &= ~(1<<bit) )

// set pin (bit) in DDR (port) =0 🡪bit is input pin

* + #define toggle\_bit(reg,bit) (reg^=1)

// toggling bits 1🡪0 & 0🡪1

**Interrupt:**

**Non-vectored interrupt:**

1. Enable global interrupt.
2. Enable interrupt type user needs
3. write ISR

N.B.:

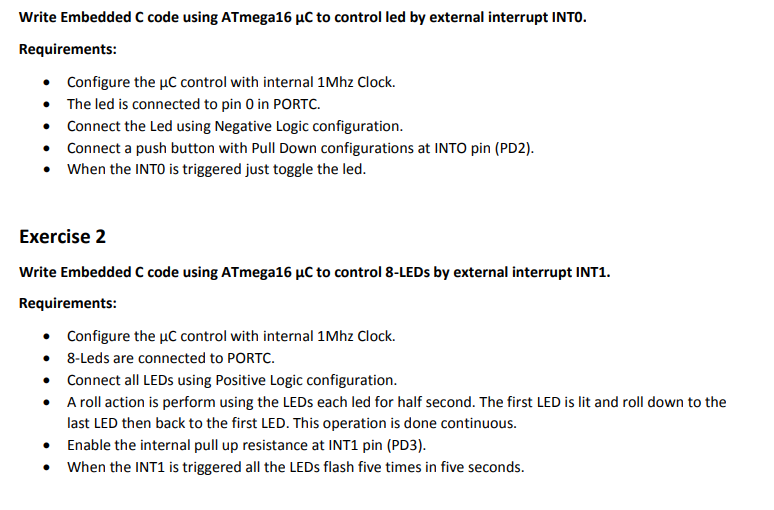
Before main loop define a function to be implemented is there is interrupt after that set the interrupt flag to zero to disable that interrupt.

**vectored interrupt:**

address and interrupt src is defined in interrupt vector table.

Examples:

**Exercise 1**



Challenge:

