1. PWM:
   a) Write a function **void pwmPin(uint8_t brightness, uint8_t pin)** that generate PWM signal using the ARM processor to produce a particular brightness (form level 0 – 255) on a LED connected to a given pin of the output portB. Use the f_{pwm} = 1/(10*255)
   b) Then program AVR using the function to change the brightness of two LEDs connected to pin 0, 3 of portB in opposite direction subsequently. At first the brightness increases (from 1 – 254) for the first LED and decreases (from 254 – 1) for the second LED and then they follows the opposite orders and repeat indefinitely such that summation of total brightness levels for two LEDs is 255.

Solution:

```
// ------- Preamble -------- //
#include <avr/io.h>       /* Defines pins, ports, etc */
#include <util/delay.h> /* Functions to waste time */
#include "pinDefines.h"
#define LED_DELAY 10  /* microseconds */

void pwmPin(uint8_t brightness, uint8_t pin) {
        uint8_t i;
        PORTB |= 1<<pin;        /* turn on */
        for (i = 0; i < 255; i++) {
                if (i >= brightness) { /* once it's been on long enough */
                PORTB &= ~(1<<pin); /* turn off */
        }
        _delay_us(LED_DELAY);
        }
}

int main(void) {
        uint8_t brightness = 0;
        int8_t direction = 1;
        // -------- Inits --------- //
        // Init all LEDs
        DDRB = 0x09;
        // ------ Event loop ------ //
        while (1) {  // Brighten and dim
                if (brightness == 0) {
                        direction = 1;
                }
                if (brightness == 255) {
                        direction = -1;
                }
                brightness += direction;
                pwmPin(brightness, 0);
                pwmPin(255 - brightness, 3);
        } /* End event loop */
        return (0); /* This line is never reached */
}
```