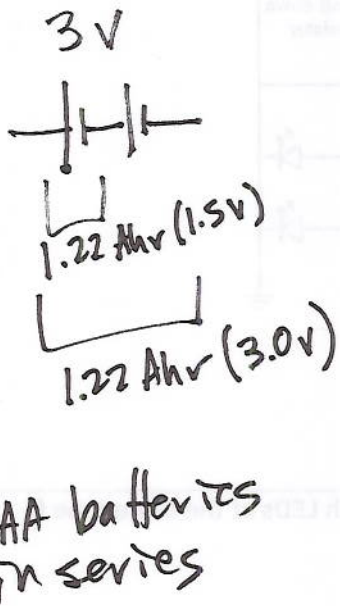
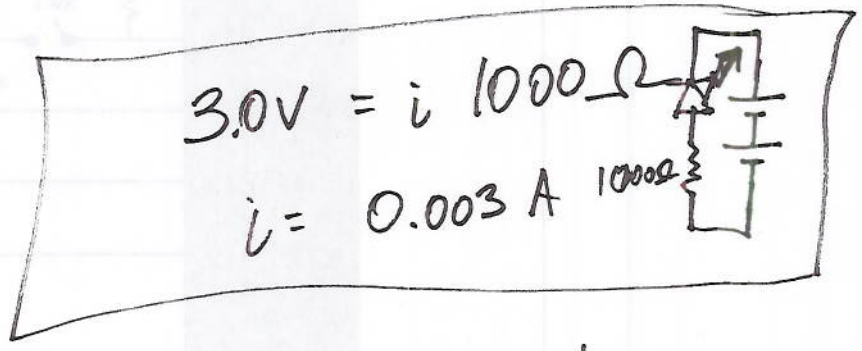
$V = IR$

$3.3V = i\ 1000\Omega$

$i = 0.003\ A$

= Capacity 1.220 Ahr (1.5v DC)





How long will the above circuit light the LED?

$$\frac{1.22\ Ahr}{0.003\ A} = 3660\ hr = 152.5\ days$$

Mobile phone call → 100 to 200 mA

Radio

Storage stick = 2.6 Ahrs (5v)

(2)

```
//*******************************************************************************
//
// Program lab4code.c
// Written by James Conrad, (some code from TI - i.e. beep, delay, march)
// Modified by:  XXXXXXXXXXXXXXXXX (you)! main function only
// Date Modified:
// Input: 4 slide switches - Port 2, pins 0 to 3
// Output:  speaker - Port 1, pin 1
// Plays audible tunes - either the Star Wars Empire March, Super Mario theme,
// Also Sprach Zarathustra, or an annoying beep.
//
//*******************************************************************************

// This macro identifies the ports available and provides the delay function
#include "msp430g2553.h"

//These macro define the frequencies for the tunes
#define e4 164
#define g3 196
#define bSH 207
#define aSH 233
#define a3 220
#define c 261
#define b3 246
#define d 294
#define eF 311
#define e 329
#define f 349
#define gF 370
#define g 391
#define gS 415
#define a 440
#define aS 455
#define b 466
#define cH 523
#define cSH 554
#define dH 587
#define dSH 622
#define eH 659
#define fH 698
#define fSH 740
#define gH 784
#define gSH 830
#define aH 880
#define bF5 932
#define c6 1046

// Prototype functions - always prototype function you use
// that means, identify in inputs and outputs
void delay_ms(unsigned int ms );
void delay_us(unsigned int us );
void beep(unsigned int note, unsigned int duration);
void freq(int x);
void march();
void mario();
void sprach();


//*******************************************************************************
```

PORT1 (PIN)

replaces the character string "aSH" with the number 233 every where in the code

```
//
// function main
// Input: Port 2 switches
// Output:  none - but calls beep which plays a tune on Port 1.1
// Main driver of Lab 4 - reads switches, calls
// Plays audible tunes - either the Star Wars Empire March or an annoying beep
//
//*****************************************************************************

int main( void ){
   // Disable the watchdog timer

   //Set the direction for the speaker port and switch ports


   //Loop forever

      //Read the slide switches, set the variables based on correct switch values
      //If all of switches are 0000, play the Empire March by calling march()
      //If all of switches are 0001, play the Sprach by calling sprach()
      //If all of switches are 0010, play the mario theme by calling mario()
      //Otherwise, play an annoying beep by calling freq(x), where x is the
      //   Port 2 input

   } // end of while loop

} // end of the main program


//YOU SHOULD NOT CHANGE ANY CODE BELOW HERE

//*****************************************************************************
//
// function delay_ms
// Input: int ms
// Output:  none
// delays for ms milliseconds
// Requires to be linked to the subroutine __delay_cycles (is in msp430g2553.h)
//
//*****************************************************************************

void delay_ms(unsigned int ms ){
   unsigned int i;
   for (i = 0; i<= ms; i++)
   __delay_cycles(500);
}


//*****************************************************************************
//
// function delay_us
// Input: int us
// Output:  none
// delays for us microseconds
// Requires to be linked to the subroutine __delay_cycles (is in msp430g2553.h)
//
//*****************************************************************************
```

```
//*****************************************************************************
//
// function march
// Input:  none
// Output: none (beep does this)
// Plays the Star Wars Empire March tune
// Requires to be in the same file as the subroutines beep and delay_ms
//
//*****************************************************************************

void march() {
    beep( a, 500);   beep(  a, 500); beep(  a, 500);    beep(  f, 350);
    beep(  cH, 150); beep(  a, 500); beep(  f, 350);    beep(  cH, 150);
    beep(  a, 650);  delay_ms(450);

    //first bit
    beep(  eH, 500);    beep(  eH, 500);    beep(  eH, 500);   beep(  fH, 350);
    beep(  cH, 150);    beep(  gS, 500);    beep(  f, 350);    beep(  cH, 150);
    beep(  a, 650);        delay_ms(450);

    //second bit...
    beep(  aH, 500);    beep(  a, 300);     beep(  a, 150);    beep(  aH, 400);
    beep(  gSH, 200);   beep(  gH, 200);    beep(  fSH, 125);  beep(  fH, 125);
    beep(  fSH, 250);   delay_ms(250);
    beep(  aS, 250);    beep(  dSH, 400);   beep(  dH, 200);   beep(  cSH, 200);

    //start of the interesting bit
    beep(cH, 125);    beep(b, 125);    beep(cH, 250);   delay_ms(250);
    beep(f, 125);     beep(gS, 500);   beep(f, 375);    beep(a, 125);
    beep(cH, 500);    beep(a, 375);    beep(cH, 125);   beep(eH, 650);

    //more interesting stuff (this doesn't quite get it right somehow)
    beep(aH, 500);    beep(a, 300);    beep(a, 150);    beep(aH, 400);
    beep(gSH, 200);   beep(gH, 200);   beep(fSH, 125);  beep(fH, 125);
    beep(fSH, 250);   delay_ms(250);

    beep(aS, 250);    beep(dSH, 400);  beep(dH, 200);   beep(cSH, 200);

    //repeat... repeat
    beep(cH, 125);    beep(b, 125);    beep(cH, 250);   delay_ms(250);

    beep(f, 250);     beep(gS, 500);   beep(f, 375);    beep(cH, 125);
    beep(a, 500);     beep(f, 375);    beep(cH, 125);   beep(a, 650);
}


//*****************************************************************************
//
// function mario
// Input:  none
// Output: none (beep does this)
// Plays the Super Mario theme song tune
// Requires to be in the same file as the subroutines beep and delay_ms
//
//*****************************************************************************

void mario(){
     beep (e,100);          beep (e,100);          beep (e,100);
```