1. vi was actually used to write the entire homework.

2.
Makefile:
```
TARGET=sinewave
OBJS=main.o gpio.o
CFLAGS=-g -Wall

$(TARGET):$(OBJS)
	$(CC) -o $(TARGET) $(OBJS)

clean:
	rm -f $(TARGET) $(OBJS)
```

gpio.h:
```
#ifndef GPIO_H
#define GPIO_h

int export_pin(char *num);
int unexport_pin(char *num);
int set_dir(char *num, char *dir);
int set_val(char *num, char *val);

#endif
```

gpio.c:

```
#include <stdio.h>
#include <fcntl.h>      // open
#include <unistd.h>     // close
#include <string.h>

// Open the export file to GPIO pin # specified by num
int export_pin(char *num)
{
	int fe = open("/sys/class/gpio/export", O_WRONLY);
	if (fe < 0) {
		fprintf(stderr, "\tError enabling\n");
		return -1;
	}
	write(fe, num, strlen(num));
	close(fe);
	usleep(1000000);        // Wait 1s for export process
	return 0;
}

// Disable GPIO pin specified by num
int unexport_pin(char *num)
{
	int fe = open("/sys/class/gpio/unexport", O_WRONLY);
	if (fe < 0) {
		fprintf(stderr, "\tError disabling\n");
		return -1;
	}
	write(fe, num, strlen(num));
	close(fe);
	usleep(1000000);        // Wait 1 sec
	return 0;
}

// Configure pin direction as "in" or "out"
int set_dir(char *num, char *dir)
{
	char *path = "/sys/class/gpio/gpio";
	char direction[11] = "/direction";
```

```
        int fd;

        // Build the direction file path
        strcat(path, num);
        strcat(path, direction);

        // Open file and set pin direction
        fd = open(path, O_WRONLY);
        if(fd < 0) {
                fprintf(stderr, "\tError direction!\n");
                return -1;
        }
        write(fd, dir, strlen(dir));
        close(fd);

        return 0;
}

// Set pin high or low depending on val
int set_val(char *num, char *val)
{
        char *path = "/sys/class/gpio/gpio";
        char value[7] = "/value";
        int fv;

        // Build the value file path
        strcat(path, num);
        strcat(path, value);

        // Open value file and check if it opened
        fv = open(path, O_WRONLY);
        if(fv < 0) {
                fprintf(stderr, "\tError writing value!\n");
                return -1;
        }

        // Set pin
        write(fv, val, strlen(val));
        return fv;
}
```

main.c:

```
#include <stdio.h>
#include "gpio.h"

int main(int argc, char *argv[])
{
        int b = 0b11111111;

        // Export 8 GPIO pins for the DAC
        export_pin("4");
        export_pin("17");
        export_pin("18");
        export_pin("27");
        export_pin("22");
        export_pin("23");
        export_pin("24");
        export_pin("25");

        // Configure all 8 DAC GPIO as output
        set_dir("4", "out");
        set_dir("17", "out");
        set_dir("18", "out");
        set_dir("27", "out");
        set_dir("22", "out");
```

```
        set_dir("23", "out");
        set_dir("24", "out");
        set_dir("35", "out");

        return 0;
}
```

3. I've learned my lesson, start the homeworks earlier, especially when you write LONG. You really mean it.

4. enscript --header='$n %D $C|$%|Tyler Punch'
        hw.05.txt -o temp | ps2pdf temp Punch-Tyler-ECE331-HW05.pdf