

```
/*
lab1.c
Christopher Brant
cbrant
ECE 2220, Fall 2016
MPI
```

*Purpose: The purpose of this machine problem is for an overall review of C programming.*

*Assumptions: There are bugs that must be fixed.*

*Bugs: Need to fix the algorithm so that it can check that a higher waveform is only taken if there is not another number of a higher frequency.*

*\*Bug fixed\*count initialization was placed inside second for loop\**

*To create a nicely formatted PDF file for printing install the enscript command. To create a PDF for "file.c" in landscape with 2 columns do:*  
*enscript file.c -G2rE -o - | ps2pdf - file.pdf*

```
*/
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#define MAXLINE 100
#define MAXSAMPLES 500
#define STOPCOUNT 3
#define MINTHRESH 3
```

```
int main()
{
    char line[MAXLINE];
    int corr_thresh = -1;
    int pos = -1;
    int val;
    int count;
    int maxVal = -1;
    int minVal;
    int samples[MAXSAMPLES] = {0};

    printf("What is the correlation threshold? ");

    fgets(line, MAXLINE, stdin);
    sscanf(line, "%d", &corr_thresh);

    while (corr_thresh < MINTHRESH)
    {
        printf("That is not a valid correlation threshold!\n");

        if (corr_thresh == -1)
        {
            printf("Goodbye\n");
            exit(1);
        }

        fgets(line, MAXLINE, stdin);
        sscanf(line, "%d", &corr_thresh);
    }

    printf("\nThe correlation threshold is equal to: %d\n", corr_thresh);

    printf("What is the minimum correlation value? ");
    fgets(line, MAXLINE, stdin);
```

```
    sscanf(line, "%d", &minVal);

    while (minVal <= 0)
    {
        if (minVal == -1)
        {
            printf("That is not a valid minimum correlation value.\nGoodbye\n");
            exit(1);
        }

        fgets(line, MAXLINE, stdin);
        sscanf(line, "%d", &minVal);
    }

    printf("\nThe minimum correlation value is equal to: %d\n", minVal);

    while (1 == 1)
    {
        //Next collect the samples.
        int sampleCount = 0;
        int i = 0;
        pos = -1;
        maxVal = -1;
        /*These variables are placed here so that they reset every time a set of samples is collected.*/

        printf("\nPlease insert the waveform sample values: ");

        while (sampleCount != STOPCOUNT && i < MAXSAMPLES)
        {
            fgets(line, MAXLINE, stdin);
            sscanf(line, "%d", &samples[i]);

            if (samples[0] == -1)
            {
                printf("Goodbye\n");
                exit(1);
            }

            if (i >= 3 && samples[i] == 0 && samples[i-1] == 0 && samples[i-2] == 0)
            {
                sampleCount = STOPCOUNT;
            }

            i++;
        }

        int j, k, valCount = 0;
        val = 0;
        //Work with this section to check for waveform
        for (j = 0; j <= i; j++)
        {
            count = 1;

            for (k = j + 1; k <= i; k++)
            {
                if (samples[j] >= minVal && samples[j] == samples[k])
                {
                    count++;
                }
            }

            if (count >= corr_thresh)
            {
                //do something
            }
        }
    }
}
```

```
        val = samples[j];
        if (val > maxVal)
        {
            maxVal = val;
            valCount = count;
        }
    }

    if (maxVal != -1)
        val = maxVal;

    count = 0;

    for (j = 0; j <= i; j++)
    {
        if (samples[j] == val && val != 0)
        {
            count++;

            if (count == 1)
            {
                pos = j + 1;
            }
        }
    }

    if (pos == -1)
    {
        printf("No waveform detected\n");
    }
    else
    {
        printf("Waveform detected at position %d with value %d and
appears %d times\n", pos, val, valCount);
    }
    exit(0);
}
```