Dylan Thornburg

Problem 1:
Write a program that includes an array of 100 unsigned integers. Fill that array with one cycle of a sine wave scaled for use with an A/D converter such that the A/D converter input value ranges from 0-4095.
Next, print out that array as a C data structure to a file called sindat.h – an initialized array of integers, such that it can be included in another C program. The output is source code that could be included directly into another C program. Test the output to make sure it is proper C code.

**CODE**

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

int main() {
   unsigned int sin_wave[100];

   // Fill the array with sine values
   int i;
   for (i = 0; i < 100; i++)
   {
      double sin_value = sin(2 * M_PI * i / 100);
      sin_wave[i] = (unsigned int)(2047.5 * (sin_value+1)); // Scale and offset to fit 0-4095 range using half of 4095
   }

   FILE *file = fopen("sindat.h", "w");
   if (file == NULL)
   {
      fprintf(stderr, "ERROR. \n");
      return 1;
   }

   // Write the array as a C data structure to the file
   fprintf(file, "unsigned int sin_data[%d] = {\n", 100);
   for (i = 0; i < 100; i++)
   {
      fprintf(file, "%u", sin_wave[i]);
      if (i != 99)
      {
         fprintf(file, ",");
      }
      fprintf(file, "\n");
   }
   fprintf(file, "};\n");
   fclose(file);
   return 0;
}
```

**Sundat.h output:**

```
unsigned int sin_data[100] = {
2047,
2176,
2304,
2431,
2556,
2680,
2801,
2919,
3033,
3144,
3250,
3352,
3449,
3540,
3625,
3703,
3776,
3841,
3900,
3951,
3994,
4030,
4058,
4078,
4090,
4095,
4090,
4078,
4058,
4030,
3994,
3951,
3900,
3841,
3776,
3703,
3625,
3540,
3449,
3352,
3250,
3144,
3033,
2919,
2801,
2680,
```

2556,
2431,
2304,
2176,
2047,
1918,
1790,
1663,
1538,
1414,
1293,
1175,
1061,
950,
844,
742,
645,
554,
469,
391,
318,
253,
194,
143,
100,
64,
36,
16,
4,
0,
4,
16,
36,
64,
100,
143,
194,
253,
318,
391,
469,
554,
645,
742,
844,
950,
1061,
1175,
1293,

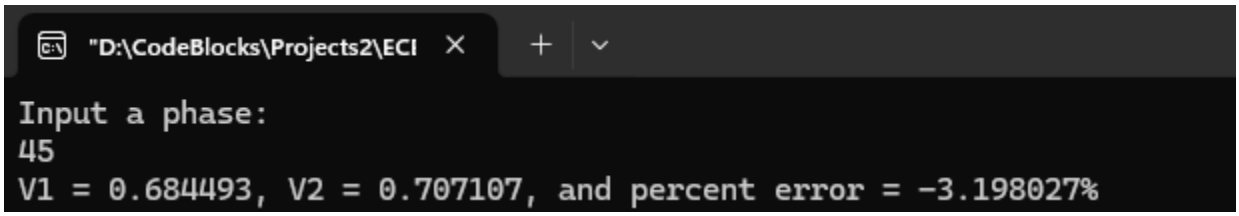
```
1414,
1538,
1663,
1790,
1918
};
```

Problem 2:
Write a program that uses a #include statement to include the sindat.h file from your problem 2. This new program should allow you to:
1) Input an integer between 0 and 359 representing phase in degrees. This is I.
2) Use the lookup table in sindat.h to approximate the sine of the input value I. This is V1.
3) Use the sine function in C to calculate the sine of the input value I. This is V2
4) Print out V1, V2, and the percentage error in V1 as compared to V2.
Example: If I is 45, the output values should be something like:

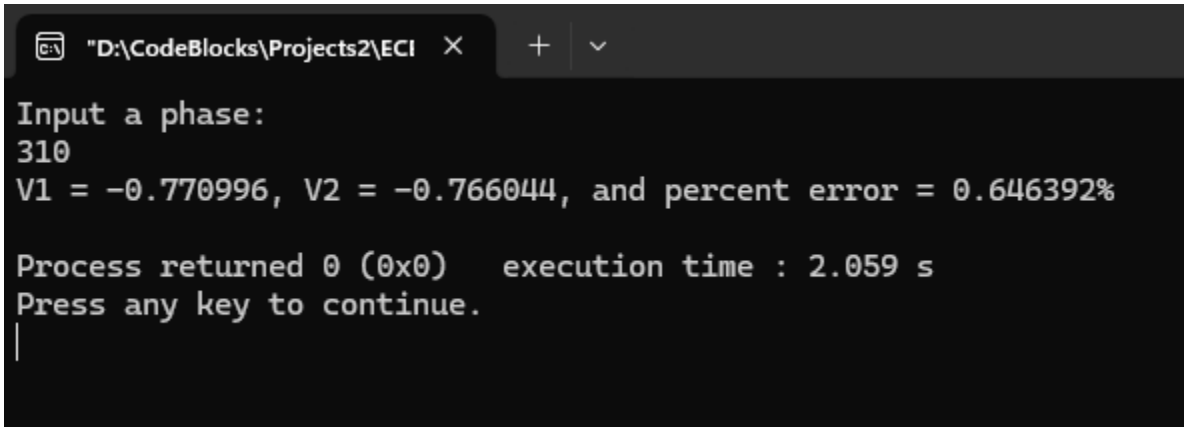
0.705 0.707 -0.28%

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "sindat.h"
int main()
{
   int i;
   int j;
   double v1;
   double v2;
   double err;
   printf("Input a phase: \n");
   scanf("%d", &i);
   if(i < 0 || i > 359)
   {
      printf("MM MM!! NOT BETWEEN 0-359 stoopid idot");
      return 0;
   }
   if(i==0)
   {
      v1=0.0;
      v2=0.0;
      err=0.0;
   }
   else
```

```
    {
      j = i/3.6;
      v1 = sin_data[j]/2047.5-1;
      v2 = sin(M_PI/180 * (double)i);
      err = (v1-v2)/v2 * 100;
    }
    printf("V1 = %lf, V2 = %lf, and percent error = %lf%% \n", v1, v2, err);
    return 0;
}
```

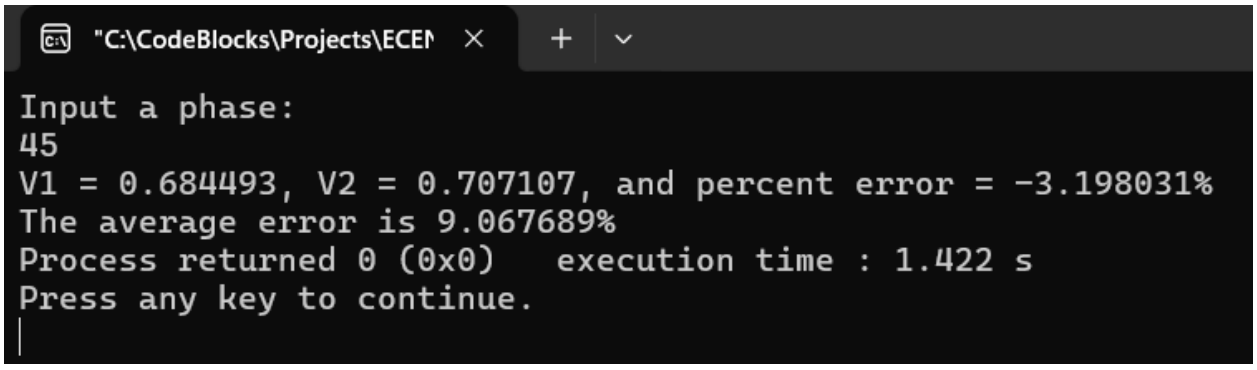



Mine was not as good for the 45 degree example.




## EXTRA STUFF BEYOND Q2 REQUIREMENTS:

I did some extra coding to compile my average percent error (using magnitude of the error) and it was even worse. It ended up being around 9% (pretty bad)




However if I average percent error without accounting for magnitude (meaning something like the avg err of -1% and 1% becomes 0 percent error instead of 1%), my avg error gets

significantly better

```
Input a phase:
45
V1 = 0.684493, V2 = 0.707107, and percent error = -3.198031%
The average error is 0.578785%
Process returned 0 (0x0)   execution time : 1.063 s
Press any key to continue.
```

Problem 3:

Write a program that prompts you to enter RGB color values where each of red, green, and blue is between 0 and 255. If you enter an incorrect number, report an input error. Otherwise, compute the equivalent HSL color value and print it out. You can find the formulae, examples, and a tester online.

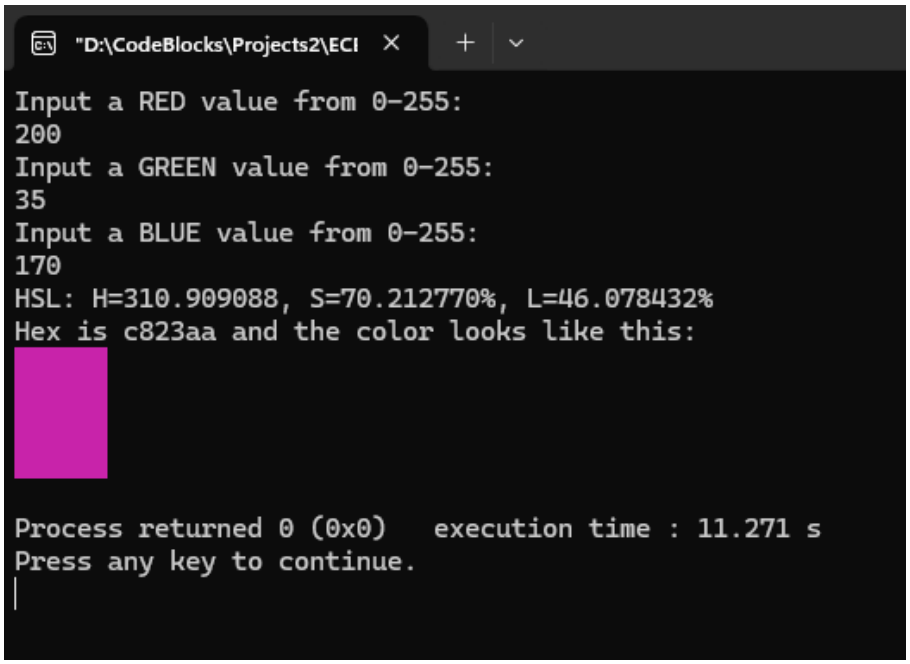
**CODE:**

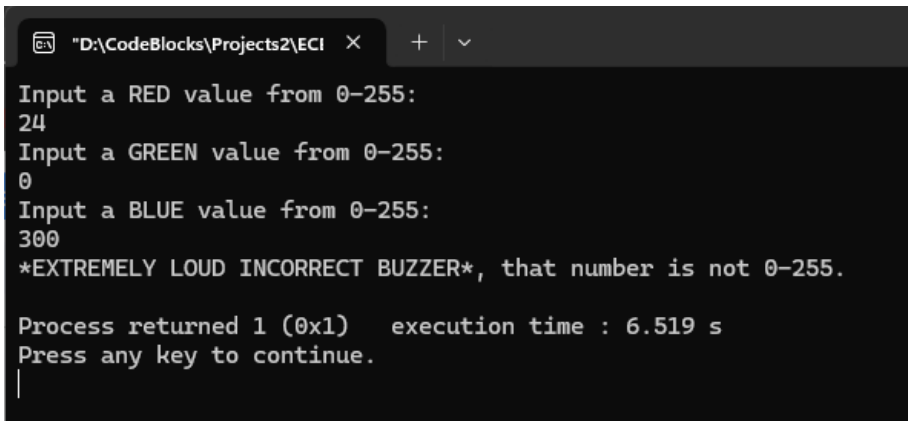
```
#include <stdio.h>
#include <stdlib.h>

int main()
{
   int red;
   int green;
   int blue;
   float r; //Note: do not use doubles. Use floats. Math works with integers so much easier.
   float g;
   float b;
   printf("Input a RED value from 0-255: \n");
   scanf("%d", &red);
   if(red < 0 || red > 255)
   {
      printf("*EXTREMELY LOUD INCORRECT BUZZER*, that number is not 0-255. \n");
      return 1;
   }
   printf("Input a GREEN value from 0-255: \n");
   scanf("%d", &green);
   if(green < 0 || green > 255)
   {
      printf("*EXTREMELY LOUD INCORRECT BUZZER*, that number is not 0-255. \n");
      return 1;
   }
   printf("Input a BLUE value from 0-255: \n");
   scanf("%d", &blue);
   if(blue < 0 || blue > 255)
   {
      printf("*EXTREMELY LOUD INCORRECT BUZZER*, that number is not 0-255. \n");
      return 1;
```

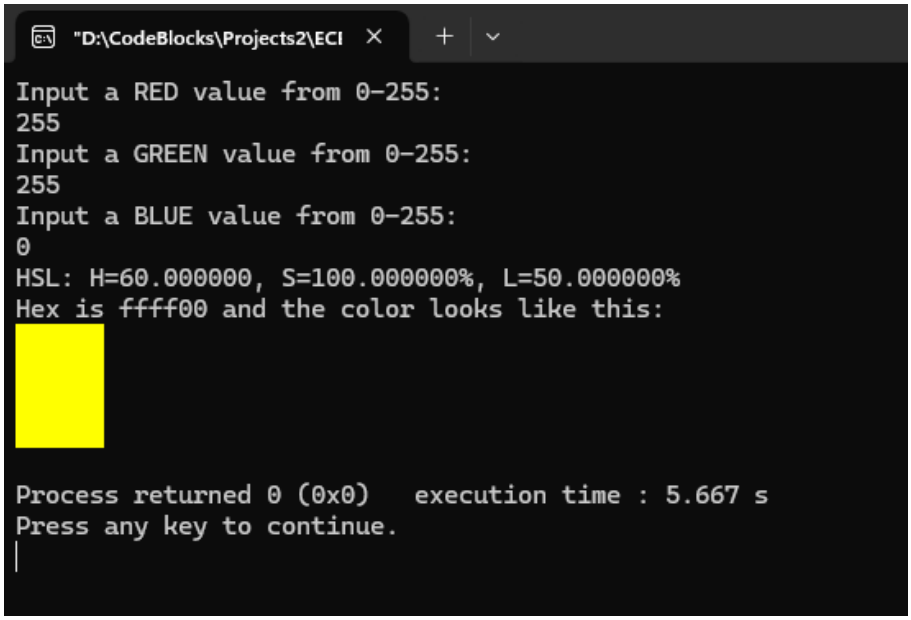
```
    }
    r = red/255.0;
    g = green/255.0;
    b = blue/255.0;
    float cmax = fmaxf(fmaxf(r, g), b); //fmaxf and fminf necessary for float values
    float cmin = fminf(fminf(r, g), b);
    float delta = cmax-cmin;
    float h = 0;
    if (delta == 0)
    {
       h = 0;
    }
    else if (cmax == r)
    {
       h = 60 * ((g - b) / delta);
    }
    else if (cmax == g)
    {
       h = 60 * ((b - r) / delta + 2);
    }
    else if (cmax == b)
    {
       h = 60 * ((r - g) / delta + 4);
    }
    if (h < 0)
    {
       h += 360;
    }
    double l = (cmax +cmin)/2;
    double s;
    if(delta==0)
    {
       s=0;
    }
    else
    {
       s=delta/(1-fabsf(2*l-1)); //same thing as fmaxf and fminf
    }
    s*=100;
    l*=100;
    printf("HSL: H=%f, S=%f%%, L=%f%%\n", h, s, l);
    printf("Hex is %.2x%.2x%.2x and the color looks like this: \n", red, green, blue);
    printf("\033[48;2;%d;%d;%dm", red, green, blue); //Found this method of printing out colors online. It just
changes text color including blank spaces so I can make this cool text box of the chosen color.
    printf("      \n");
    printf("      \n");
    printf("      \n");
    printf("      \n");
    printf("\033[0m");
```

```
    return 0;
}
```

**TESTS:**






Problem 4 (20 pts):
I wish to analyze the scholarly writings listed on the following pages in order to study the role of repetition in contemporary balladry. In order to assist me in this task you are to write a C program that parses the text on the next page and performs the following functions. Read the description very carefully.
Each time a word is repeated (i.e., the same word appears two or more times in a row within the same line of the song, with or without punctuation between) you will output the word, followed by the number of local repetitions. i.e. I-I-I is 3 repetitions.
You should treat all capitalized copies of the word as the same as uncapitalized copies.
Assume that hyphens, parentheses, commas, and periods are punctuation that separates words. Apostrophes are part of words.
You must figure out how to represent the text on the next 2 pages as a single constant string in your program.

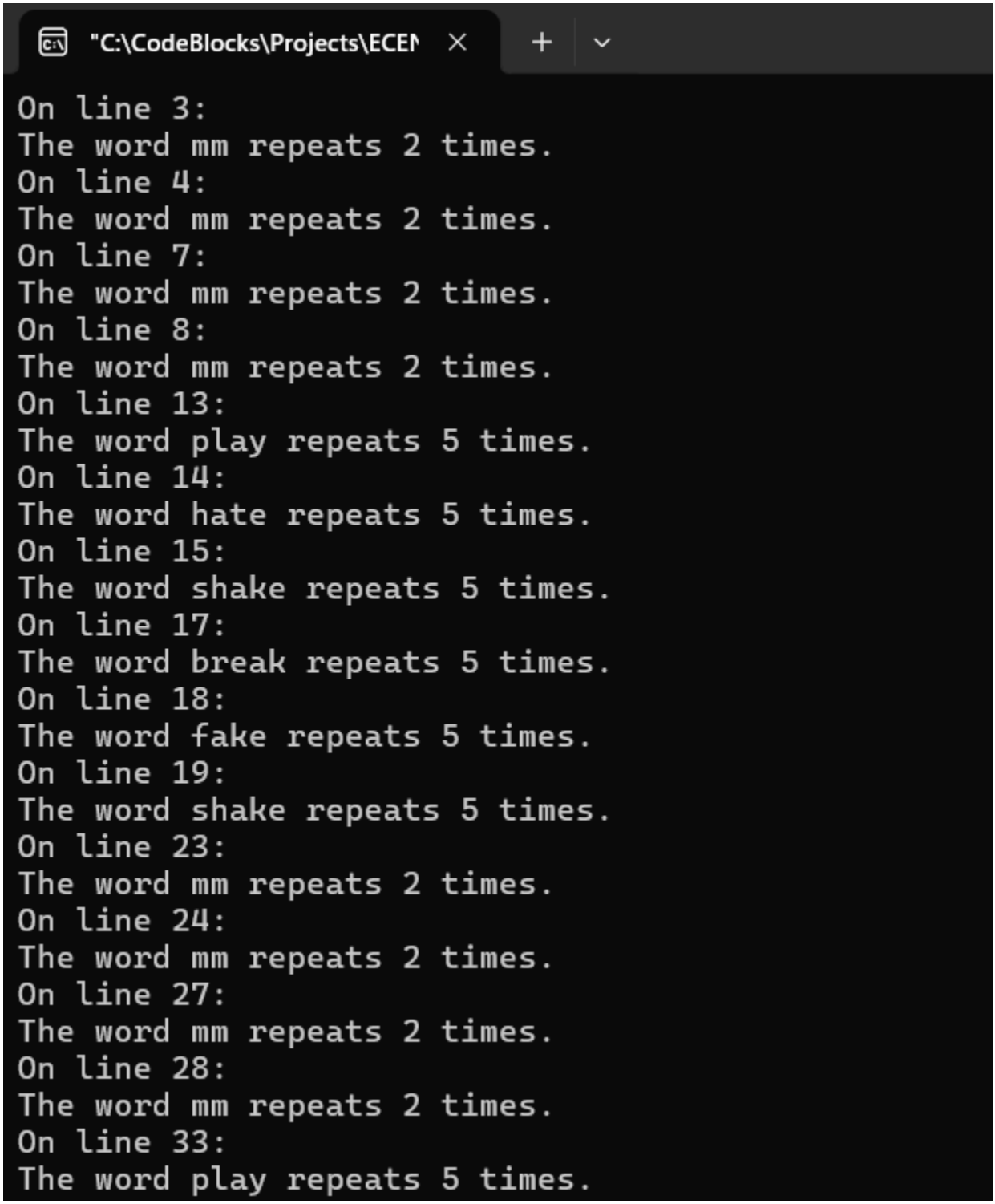
**CODE:**

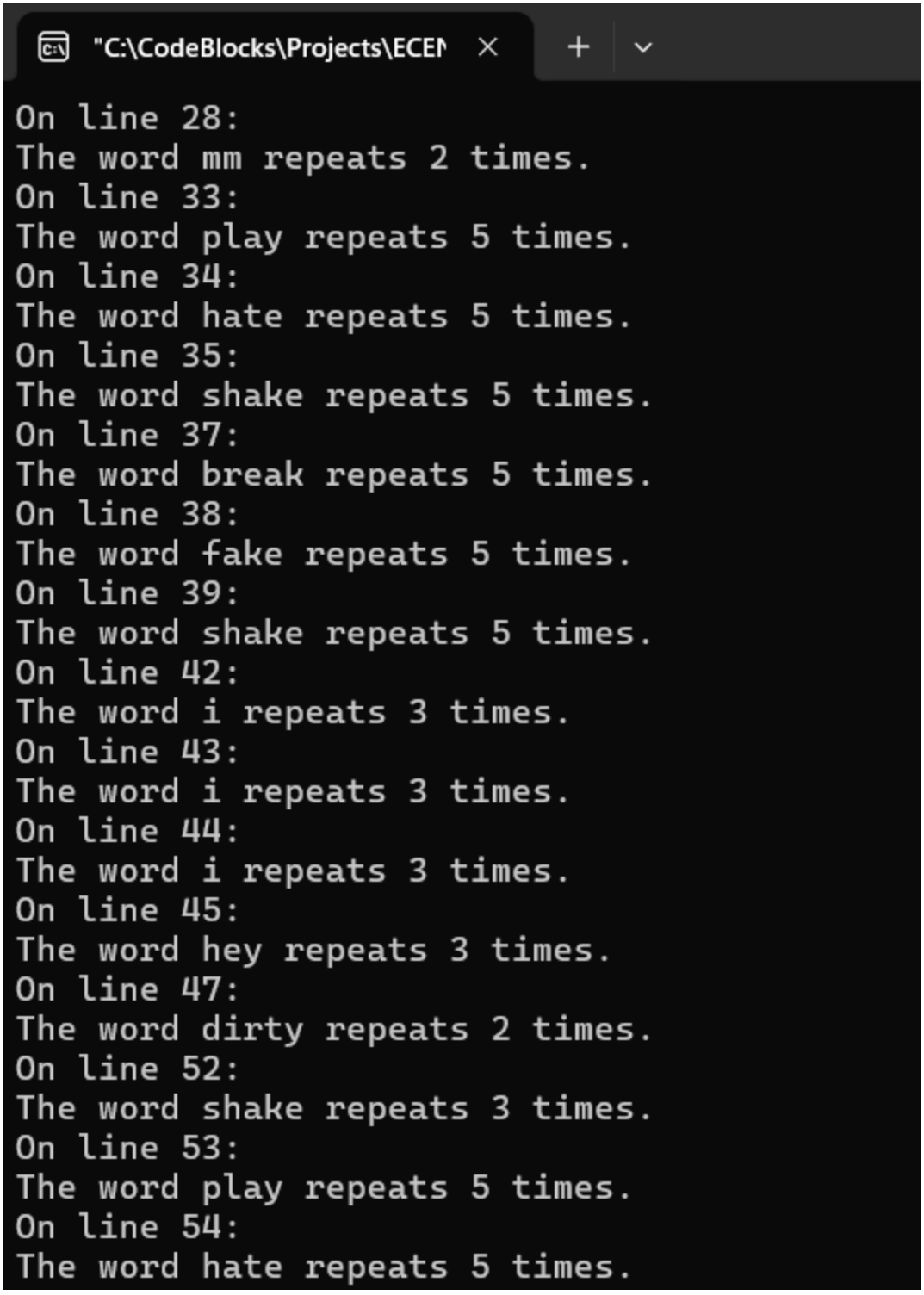
```
#include <stdio.h>
#include <stdlib.h>

int main()
{
   FILE* file = fopen("Swift.txt", "r");
   if (file == NULL)
   {
      printf("Error opening file.\n");
      return 1;
```

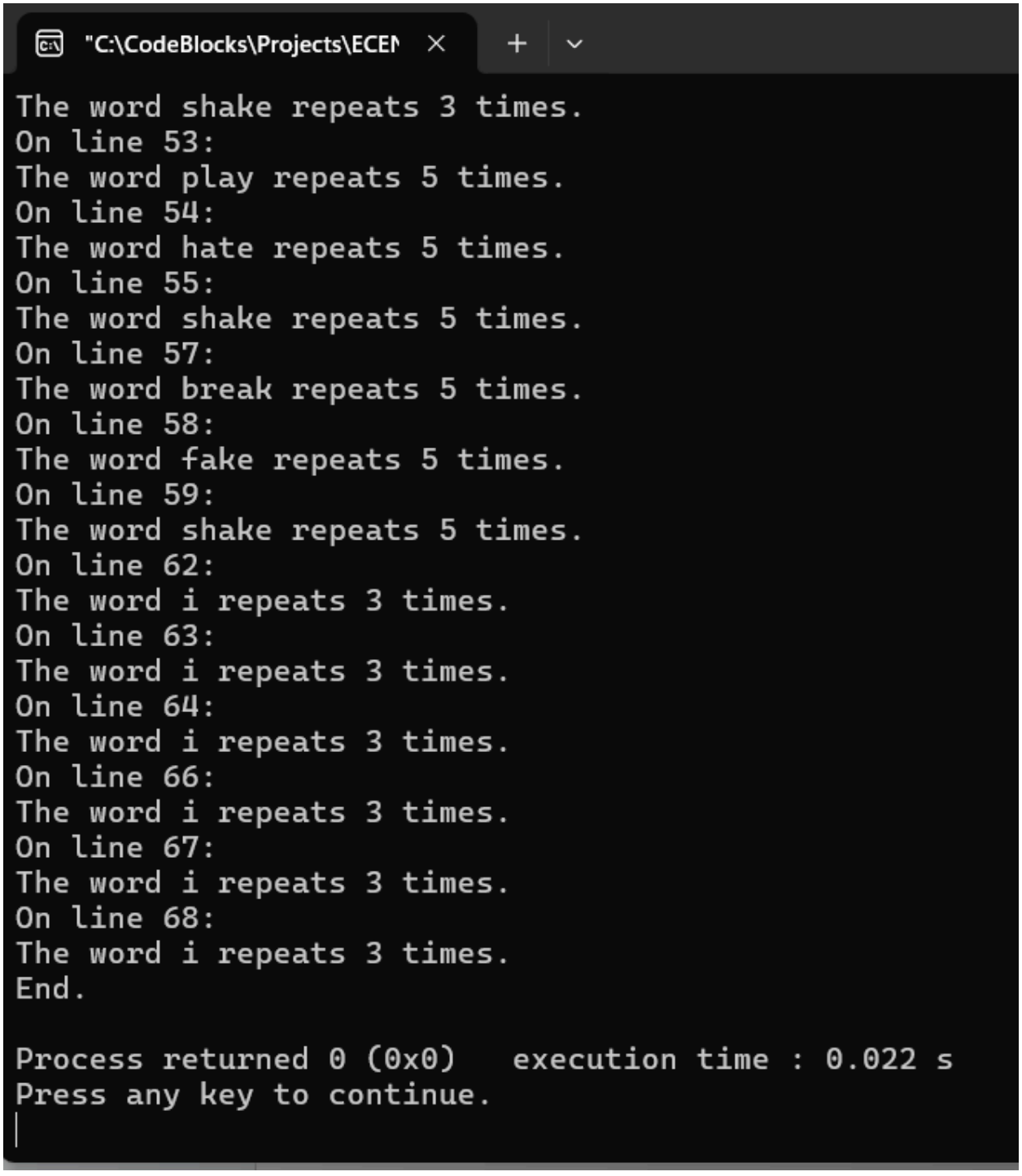
```
}
char string[68][100];
int i = 0;
int j;
int k;
int x;
char *temp;
while (i < 68 && fgets(string[i], sizeof(string[i]), file) != NULL)
{
   temp = string[i];
   j = 0;
   k = 0;

   while (temp[j])
   {
      if (temp[j] == '-' || temp[j] == '.' || temp[j] == ';' || temp[j] == ',' || temp[j] == ':' || temp[j] == '(' || temp[j] == ')')
      {
         temp[j] = ' ';
      }
      j++;
   }

   // Reset k to 0 at the beginning of each line
   k = 0;

   char prev[100] = "";
   int count = 1;

   while (temp[k] != '\0')
   {
      x = 0;
      while(temp[x] != '\n')
      {
         temp[x] = tolower(temp[x]);
         x++;
      }
      temp[x] = ' ';
      temp[x+1] = '\n';
      char word[100] = {0};

      // Skip past spaces
      while (temp[k] == ' ')
      {
         k++;
      }

      j = 0;
      while (temp[k] != ' ' && temp[k] != '\0')
      {
```

```
            word[j++] = temp[k++];
        }
        word[j] = '\0'; // Null da word


        if (strcasecmp(word, prev) == 0)
        {
            count++;
        }
        else
        {

            if (count > 1)
            {
                printf("On line %d: \n", (i+1));
                printf("The word %s repeats %d times.\n", prev, count);
            }
            count = 1;
            strcpy(prev, word);
        }

        // Move to the next word
        while (temp[k] == ' ')
        {
            k++;
        }
    }
    // Case for only if the very last word is repeated
    if (count > 1)
    {
        printf("The word %s repeats %d times.\n", prev, count);
    }

    i++;
  }
  printf("End.\n");
  fclose(file);
  return 0;
}
```

**SCREENSHOTS:**




"C:\CodeBlocks\Projects\ECE
On line 28:
The word mm repeats 2 times.
On line 33:
The word play repeats 5 times.
On line 34:
The word hate repeats 5 times.
On line 35:
The word shake repeats 5 times.
On line 37:
The word break repeats 5 times.
On line 38:
The word fake repeats 5 times.
On line 39:
The word shake repeats 5 times.
On line 42:
The word i repeats 3 times.
On line 43:
The word i repeats 3 times.
On line 44:
The word i repeats 3 times.
On line 45:
The word hey repeats 3 times.
On line 47:
The word dirty repeats 2 times.
On line 52:
The word shake repeats 3 times.
On line 53:
The word play repeats 5 times.
On line 54:
The word hate repeats 5 times.




Lyrics:
I stay out too late
Got nothing in my brain
That's what people say, mm-mm
That's what people say, mm-mm

I go on too many dates
But I can't make 'em stay
At least that's what people say, mm-mm
That's what people say, mm-mm
But I keep cruising
Can't stop, won't stop moving
It's like I got this music
In my mind, saying it's gonna be alright
'Cause the players gonna play, play, play, play, play
And the haters gonna hate, hate, hate, hate, hate, baby
I'm just gonna shake, shake, shake, shake, shake
Shake it off, I shake it off
Heartbreakers gonna break, break, break, break, break
And the fakers gonna fake, fake, fake, fake, fake, baby
I'm just gonna shake, shake, shake, shake, shake
I shake it off, I shake it off
I'll never miss a beat
I'm lightning on my feet
And that's what they don't see, mm-mm
That's what they don't see, mm-mm
I'm dancing on my own (Dancin' on my own)
I'll make the moves up as I go (Moves up as I go)
And that's what they don't know, mm-mm
That's what they don't know, mm-mm
But I keep cruising
Can't stop, won't stop grooving
It's like I got this music
In my mind, saying it's gonna be alright
'Cause the players gonna play, play, play, play, play
And the haters gonna hate, hate, hate, hate, hate, baby
I'm just gonna shake, shake, shake, shake, shake
Shake it off, I shake it off
Heartbreakers gonna break, break, break, break, break
And the fakers gonna fake, fake, fake, fake, fake, baby
I'm just gonna shake, shake, shake, shake, shake
I shake it off, I shake it off
Shake it off, I shake it off
I-I-I shake it off, I shake it off
I-I-I shake it off, I shake it off
I-I-I shake it off, I shake it off

Hey, hey, hey
Just think while you've been getting down and out
About the liars, and the dirty dirty cheats of the world
You coulda been getting down to this sick beat
My ex man brought his new girlfriend
She's like "Oh, my God!" but I'm just gonna shake it
To the fella over there with the hella good hair
Won't you come on over baby we could shake, shake, shake (Yeah, oh)
'Cause the players gonna play, play, play, play, play
And the haters gonna hate, hate, hate, hate, hate, baby (Haters gonna hate)
I'm just gonna shake, shake, shake, shake, shake
I shake it off, I shake it off
Heartbreakers gonna break, break, break, break, break
And the fakers gonna fake, fake, fake, fake, fake, baby
I'm just gonna shake, shake, shake, shake, shake
Shake it off, I shake it off
Shake it off, I shake it off
I-I-I shake it off, I shake it off
I-I-I shake it off, I shake it off
I-I-I shake it off, I shake it off
Shake it off, I shake it off
I-I-I shake it off, I shake it off
(You got to) I-I-I shake it off, I shake it off
I-I-I shake it off, I shake it off