

WEEK 2

READABILITY

```
#include <cs50.h>
```

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

```
#include <ctype.h>
```

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

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

```
int main(void)
```

```
{
```

```
    int count_letter = 0;
```

```
    int count_word = 1;
```

```
    int count_sentence = 0;
```

```
    string text = get_string("Text: ");
```

```
    int text_length = strlen(text);
```

```
    for (int i = 0; i < text_length; i++)
```

```
    {
```

```
        if (isalpha(text[i]))
```

```
        {
```

```
            count_letter++;
```

```
        }
```

```
    }
```

```
    //printf("%i\n", count_letter);
```

```
    for (int i = 0; i < text_length; i++)
```

```
    {
```

```
        if (isspace(text[i]))
```

```
        {
```

```
            count_word++;
```

```
        }
```

```
    }
```

```
    for (int i = 0; i < text_length; i++)
```

```
    {
```

```
        if (text[i] == '.' || text[i] == '?' || text[i] == '!')
```

```
        {
```

```
            count_sentence++;
```

```
        }
```

```
    }
```

```
    float calculation = (0.0588 * count_letter / count_word * 100) - (0.296 * count_sentence /  
count_word * 100) - 15.8; // to calculate Index
```

```
    int index = round(calculation);
```

```
    if (index < 1) //
```

```
    {
```

```
        printf("Before Grade 1\n");
```

```
    }
```

```
    else if (index > 16)
```

```
    {
```

```
        printf("Grade 16+\n");
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("Grade %i\n", index);
```

```
    }
```

```
}
```

```

~/pset1/ $ cd pset1
~/pset1/ $ ./readability
Text:
Before Grade 1
~/pset1/ $ ./readability
Text: Congratulations! Today is your day. You're off to Great Place! You're off and away!
Grade 3
~/pset1/ $ ./readability
Text: Harry Potter was a highly unusual boy in many ways. For one thing, he hated the summer holidays more than any other time of year. For another, he really wanted to do his homework, but was forced to do it in secret, in the dead of the night. And he also happened to be a wizard.
bash: Harry: command not found
~/pset1/ $ ./readability
Text: Harry Potter was a highly unusual boy in many ways. For one thing, he hated the summer holidays more than any other time of year. For another, he really wanted to do his homework, but was forced to do it in secret, in the dead of the night. And he also happened to be a wizard.
Grade 5
~/pset1/ $ ./readability
Text: As the average number of letters and words per sentence increases, the Coleman-Liau index gives the text a higher reading level. If you were to take this paragraph, for instance, which has longer words and sentences than either of the prior two examples, the formula would give the text an eleventh grade reading level.
Grade 11
~/pset1/ $

```

```

1 #include <cs50.h>
2 #include <stdio.h>
3 #include <ctype.h>
4 #include <string.h>
5 #include <math.h>
6
7 int main(void)
8 {
9     int count_letter = 0;
10    int count_word = 1;
11    int count_sentence = 0;
12    string text = get_string("Text: ");
13
14    int text_length = strlen(text);
15    for (int i = 0; i < text_length; i++)
16    {
17        if (isalpha(text[i]))
18        {
19            count_letter++;
20        }
21    }
22    //printf("%i\n", count_letter);
23    for (int i = 0; i < text_length; i++)
24    {
25        if (isspace(text[i]))
26        {
27            count_word++;
28        }
29    }
30
31    for (int i = 0; i < text_length; i++)
32    {
33        if (text[i] == '.' || text[i] == '?' || text[i] == '!')
34        {
35            count_sentence++;
36        }
37    }
38
39    float calculation = (0.0588 * count_letter / count_word * 100) - (0.296 * count_sentence

```

CAESER)

```
#include <cs50.h>
```

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

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

```
#include <ctype.h>
```

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

```
int main(int argc, string argv[])
```

```
{
```

```
    if (argc != 2)
```

```
    {
```

```
        printf("Usage: ./caesar key\n");
```

```
        return 1;
```

```
    }
```

```
    int arg_length = strlen(argv[1]);
```

```
    for (int i = 0; i < arg_length; i++)
```

```
    {
```

```
        if (!isdigit(argv[1][i]))
```

```
        {
```

```
            printf("Usage: ./caesar key");
```

```
            return 1;
```

```
        }
```

```
    }
```

```
    int key = atoi(argv[1]);
```

```
    string plaintext = get_string("plaintext: ");
```

```
    printf("ciphertext: ");
```

```
    int plaintext_length = strlen(plaintext);
```

```
    for (int i = 0; i < plaintext_length; i++)
```

```
    {
```

```
        if (isupper(plaintext[i]))
```

```
        {
```

```
            printf("%c", (((plaintext[i] - 65) + key) % 26) + 65);
```

```
        }
```

```
        else if (islower(plaintext[i]))
```

```
        {
```

```
            printf("%c", (((plaintext[i] - 97) + key) % 26) + 97);
```

```
        }
```

```
        else
```

```
        {
```

```
            printf("%c", plaintext[i]);
```

```
        }
```

```

    }
    printf("\n");
}

```

```

~/pset1/ $ make caesar
~/pset1/ $ make caesar
clang -ggdb3 -O0 -std=c11 -Wall -Werror -Wextra -Wno-sign-compare -Wno-unused-parameter -Wno-unused-variable -Wshadow caesar.c -lcrypt -lc550 -lm -o caesar
~/pset1/ $ ./caesar 20
bash: ./caesar: No such file or directory
~/pset1/ $ ./caesar
Usage: ./caesar key
~/pset1/ $ ./caesar 20
bash: ./caesar: No such file or directory
~/pset1/ $ ./caesar 20
plaintext: Success
ciphertext: Mowwym
~/pset1/ $ ./caesar 1
bash: ./caesar: No such file or directory
~/pset1/ $ ./caesar 1
plaintext: hello
ciphertext: ifmp
~/pset1/ $ ./caesar HELLO
~/pset1/ $

```

```

1 #include <cs50.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <ctype.h>
5 #include <stdlib.h>
6
7 int main(int argc, string argv[])
8 {
9     if (argc != 2)
10     {
11         printf("Usage: ./caesar key\n");
12         return 1;
13     }
14     int arg_length = strlen(argv[1]);
15     for (int i = 0; i < arg_length; i++)
16     {
17         if (!isdigit(argv[i][1]))
18         {
19             printf("Usage: ./caesar key\n");
20             return 1;
21         }
22     }
23
24     int key = atoi(argv[1]);
25     string plaintext = get_string("plaintext: ");
26     printf("ciphertext: ");
27     int plaintext_length = strlen(plaintext);
28     for (int i = 0; i < plaintext_length; i++)
29     {
30         if (isupper(plaintext[i]))
31         {
32             printf("%c", (((plaintext[i] - 65) + key) % 26) + 65);
33         }
34         else if (islower(plaintext[i]))
35         {
36             printf("%c", (((plaintext[i] - 97) + key) % 26) + 97);
37         }
38         else
39         {

```