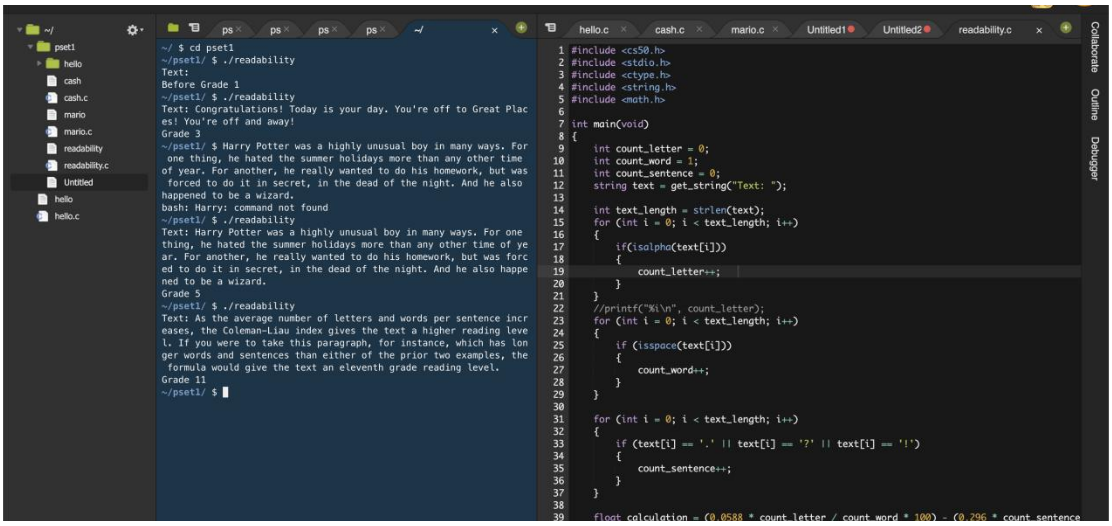
WEEK 2
READABILITY

```
#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);
    }
}
```

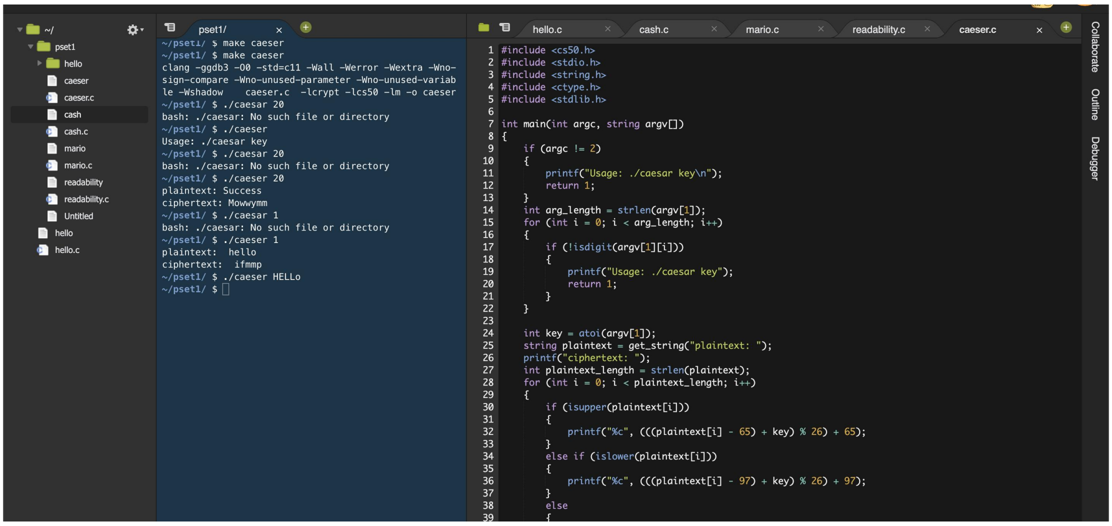


```
~/ $ cd pset1
~/pset1/ $ ./readability
Text:
Before Grade 1
~/pset1/ $ ./readability
Text: Congratulations! Today is your day. You're off to Great Plac
es! You're off and away!
Grade 3
~/pset1/ $ Harry Potter was a highly unusual boy in many ways. For
 one thing, he hated the summer holidays more 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 ye
ar. For another, he really wanted to do his homework, but was forc
ed to do it in secret, in the dead of the night. And he also happe
ned to be a wizard.
Grade 5
~/pset1/ $ ./readability
Text: As the average number of letters and words per sentence incr
eases, the Coleman-Liau index gives the text a higher reading leve
l. If you were to take this paragraph, for instance, which has lon
ger words and sentences than either of the prior two examples, the
 formula would give the text an eleventh grade reading level.
Grade 11
~/pset1/ $
```

```c
#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
```

CAESER)

```c
#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 caeser
~/pset1/ $ make caeser
clang -ggdb3 -O0 -std=c11 -Wall -Werror -Wextra -Wno-
sign-compare -Wno-unused-parameter -Wno-unused-variab
le -Wshadow    caeser.c  -lcrypt -lcs50 -lm -o caeser
~/pset1/ $ ./caesar 20
bash: ./caesar: No such file or directory
~/pset1/ $ ./caeser
Usage: ./caesar key
~/pset1/ $ ./caesar 20
bash: ./caesar: No such file or directory
~/pset1/ $ ./caeser 20
plaintext: Success
ciphertext: Mowwymm
~/pset1/ $ ./caesar 1
bash: ./caesar: No such file or directory
~/pset1/ $ ./caeser 1
plaintext:  hello
ciphertext:  ifmmp
~/pset1/ $ ./caeser HELLo
~/pset1/ $ 
```

```c
#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
        {
```