CS 214: Introduction to computing systems
Winter Term, 2026

Homework 4

Due: Monday 2/2 at 11:59pm

Complete the following problem and submit your code on euclid by typing the following:

```
handin cs214 hwk4 myCode.c
```

where you replace the filename with whatever you called your code. (If you do not write this program on euclid, you'll need to transfer your code to euclid in order to submit it.)

When writing this program, you may find some functions helpful. The standard I/O library (include `stdio.h`) has a function `getchar` that takes no arguments and returns the next character of the input or `EOF` if it has ended (i.e. end of file or Control-D). Similarly, if you include `ctype.h` (for "character type"), you can use `isalpha`, which takes a char and returns whether it is an alphabet character (as opposed to a number, punctuation, white space, etc). You can also use `toupper`, which takes an alphabetic character and returns the uppercase version of it. (It converts lowercase characters into uppercase ones, but leaves uppercase characters unaffected.)

An example program using the I/O functions to read input is the following:

```
#include <stdio.h>

int main() {
  char c;
  while((c = getchar()) != EOF) {
    if(isalpha(c))
      ...
  }
}
```

When running this program on euclid (or other Linux system), you can use *input redirection* to give it input from a file. For example, the command

```
./myCode < input
```

runs the program `myCode` on the contents of `input`, just as if you'd typed them into it. You should use this to run your program on large inputs to save yourself retyping the input on each modification of the program.

Finally, remember that characters are basically a different representation of an integer. If `c` is an uppercase character, then the expression `c - 'A'` will convert it into an integer 0–25. If you have an integer `x` in this range, then `x + 'A'` will give you the corresponding character (0='A', 1='B', etc).

1. (15 points) Write a C program that reads text from standard input and generates a histogram showing the frequency of alphabetic characters it contains. Specificially, it should print all the letters A to Z (using uppercase letters). These should appear 1 per line, each followed by a space and the number of times that letter appeared in the input (in either uppercase or lowercase form). Print the numbers in a field of width 3 (use %3d in `printf` to do this). Follow this by a space and then one equal sign (`=`) for each 5 occurrences of the letter (rounded down). For example, the beginning of the output could be the following:

```
A  12 ==
B   2
C   5 =
D   0
E   0
```

Any non-alphabetic characters in the input should be ignored.

2. (1 point) At the top of your submission, put a statement about collaboration in a comment.