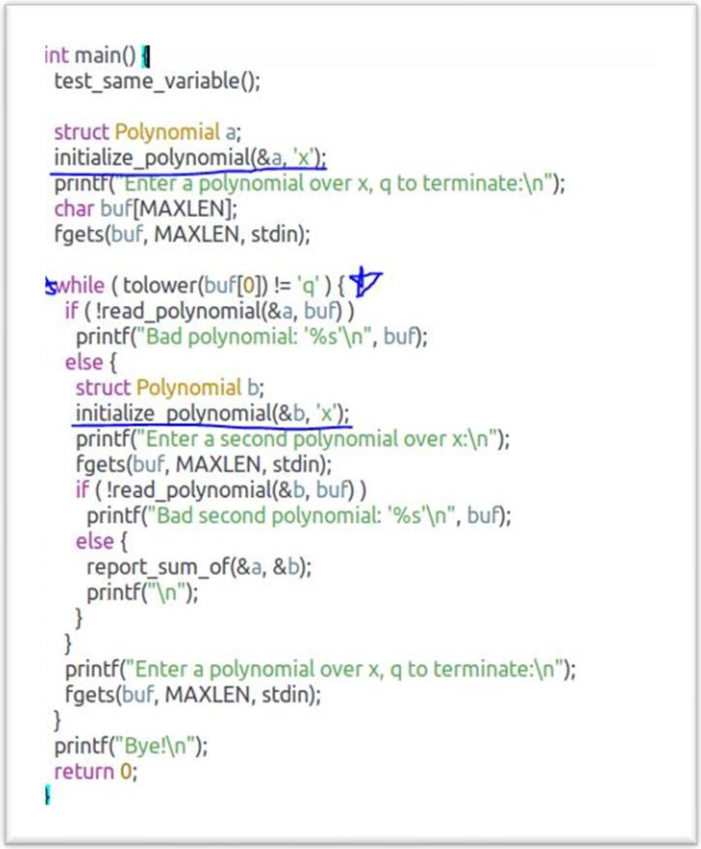
Lab 9

Author: Nigel Nelson
Course: CS2040
Date: 5/13/21

- **What did you learn from doing this lab?**
  - I learned a lot over the course of this extended lab. The first lesson that I learned is how to use input and output in C. Specifically, how when using “sscanf()” or “printf()” that it is necessary to specify the type of variable that you are attempting to read in or print out using a format specifier such as “%c”. In addition, I learned that when looping through a string with sscanf(), by using the “%n” format specifier, sccanf() will propagate a supplied variable with the number of bytes it read. This way, this value can be used to accurately move the location of the char pointer used to traverse the string. Lastly, I learned how to use a structs to model data that would likely be classes in C++ or other object-oriented programming languages.
- **What did you find challenging about the lab?**
  - The most challenging part of the lab for me was moving the pointer for the string representing a polynomial correctly as I looped. At first, I attempted to move the pointer based on the number of elements read, however, I quickly learned that numbers vary in length and this metric could not be relied up. Next, I discovered that sscanf() would gladly provide the number of bytes read which proved a much more effective way to know how much to move the pointer. However, I struggled longer than I would have liked to on the fact that a pointer moved inside the context of a helper method, will not be mirrored by the pointer in the calling function. In addition, I struggled with understanding the correct terminated conditions to use for the loops that I employed. This is because I had difficult time knowing how to differentiate between a valid polynomial that had no more terms, an invalid polynomial, and a valid +/- sign vs. an invalid +/- sign. This resulted in many long debugging sessions to understand at which point reading in a polynomial should be terminated.
- **What would you recommend changing if this lab is reused in future years?**
  - Overall, I really enjoyed this lab, prior to this lab C was very intimidating, but this lab did a great job of making me comfortable with many basic C concepts. The only way in which I believe that this lab could be improved is by updating the main.c file that is used. The reason for this is that it seemed odd that inside the main loop, only the “b” variable was re-initialized, this made it necessary to implement a cleaning method to reset all coefficients to 0 inside the read_polynomial() function, where it seemed much more natural for this to take place in the initialize function.

Lab 9

Author: Nigel Nelson
Course: CS2040
Date: 5/13/21



```
int main() {
 test_same_variable();

 struct Polynomial a;
 initialize_polynomial(&a, 'x');
 printf("Enter a polynomial over x, q to terminate:\n");
 char buf[MAXLEN];
 fgets(buf, MAXLEN, stdin);

 while ( tolower(buf[0]) != 'q' ) {
   if ( !read_polynomial(&a, buf) )
     printf("Bad polynomial: '%s'\n", buf);
   else {
     struct Polynomial b;
     initialize_polynomial(&b, 'x');
     printf("Enter a second polynomial over x:\n");
     fgets(buf, MAXLEN, stdin);
     if ( !read_polynomial(&b, buf) )
       printf("Bad second polynomial: '%s'\n", buf);
     else {
       report_sum_of(&a, &b);
       printf("\n");
     }
   }
   printf("Enter a polynomial over x, q to terminate:\n");
   fgets(buf, MAXLEN, stdin);
 }
 printf("Bye!\n");
 return 0;
}
```

-