# Week 3 — September 20, 2018

## Homework

- Read chapter 5 and chapter 7 pp. 263-287 in “Beginning C”
- Read chapters 3 and 4 in “Writing Scientific Software”

## Exercises

1. Do exercises 5-1, 5-2, and 5-4 in “Beginning C”
2. Take this quiz to test your understanding of *arrays*
3. Take this quiz to test your understanding of *pointers* (questions 5 and 6 are about dynamic memory allocation which is next week’s topic, so feel free to skip these questions)
4. Analyze the following program. What does it compute? Rewrite the program to make the code more readable.

```
#include <stdio.h>

int main(void) {
    int arr[10] = {19,74,13,67,44,80,7,36,9,77}, *p = arr;
    int i = -1;
    double val = 0.0;

    while (++i < 10)
          val += *(p++);
    val /= 10;
    printf("Value: %.2f\n",val);
    return 0;
}
```

5. Write a program that creates two arrays `arr1` and `arr2` of length $N$ and prompts the user to enter $N$ double precision floating point numbers, storing these in the array `arr1`. Your program should then copy then contents of `arr1` to `arr2` using the `memcpy` routine, which is defined in `string.h` and has the following prototype:

```
void *memcpy(void *dest, const void *src, size_t n)
```

Note that the last input $n$ should be the number of bytes to copy (and not the number of elements in your array). Finally, write out the $N$ numbers in `arr2` to check that these match the input.

**Remark**: You can define $N$ using a preprocessor macro, say, `#define N 10`.

6. Implement the "one-pass" method and the "two-pass" method for computing the variance of an array (see "Writing Scientific Software", p. 24). Use the array provided in the textbook as a test case, and compute both the one-pass and two-pass variance in single precision and double precision.

7. Modify your code from the previous exercise such that all summations are performed using the Kahan summation algorithm (see lecture slides). Using the test case from the textbook, compute both the one-pass and two-pass variance in single precision and double precision, and compare your results with those obtained with sequential summation in the previous exercise.

8. The one-pass algorithm computes the variance as

$$v = \frac{1}{n-1}\left(\left(\sum_{i=1}^{n} a_i^2\right) - \frac{1}{n}\left(\sum_{i=1}^{n} a_i\right)^2\right).$$

   For the array provided in the textbook, estimate the (relative) condition number associated with each of the following sums:

   - $S_1 = \sum_{i=1}^{n} a_i$
   - $S_2 = \sum_{i=1}^{n} a_i^2$
   - $S_2 - \frac{1}{n}S_1^2$

   Which of the three summations is most critical in terms of numerical accuracy?

9. Given $n$ numbers $x_1, x_2, \ldots, x_n$ where $\sum_{i=1}^{n} |x_i| \approx 500$ and $|\sum_{i=1}^{n} x_i| \approx 0.01$, what is approximately the worst-case relative error for sequential floating-point summation of the numbers if the unit round-off is $u$ and $nu < 0.1$?

10. What is the condition number of $f(x) = \cos(x)$?