CSC 209 Review 4

August 19, 2020

1. **K.K. 11.1:** If i is a variable and p points to i, which of the following expressions are aliases for i?

   a) `*p`

   b) `*&p`

   c) `*i`

   d) `*&i`

   e) `&p`

   f) `&*p`

   g) `&i`

   h) `&*i`

2. **K.K 11.2:** If i is an int variable and p and q are pointers to `int` which of the following assignments are legal?

   a) `p = i;`

   b) `*p = &i;`

   c) `&p =q;`

   d) `p = &q;`

   e) `p = *&q;`

   f) `p = q;`

   g) `p = *q;`

   h) `*p = q;`

   i) `*p = *q;`

3. **K.K.11.3:** The following function supposedly computes the sum and average of the numbers in the array `a`, which has length `n. avg` and sum point to variables that the function should modify. Unfortunately, the function contains several errors; find and correct them

```
    void avg_sum(double a[], int n, double *avg, double *sum)
    {
        int i;

        sum = 0.0;
        for (i = 0; i < n; i++)
            sum += a[i];
        avg = sum / n;
    }
```

4. **K.K.11.4:** Write the following function

   ```
   void swap(int *p, int *q);
   ```

   When passed the addresses of two variables, `swap` should exchange the values of the variables:

   ```
   swap(&i, &j); /* Exchanges values of i and j */
   ```

5. **K.K.11.5:** Write the following function

   ```
   void split_time(long total_sec, int *hr, int *min, int *sec);
   ```

   `total_sec` is a time represented as the number of seconds since midnight. hr, min and sec are pointers to variables in which the function will store the equivalent time in hours (0-23), minutes (0-59) and seconds (0-59), respectively.

6. **K.K.11.6:** Write the following function:

   ```
   void find_two_largest(int a[], int n, int *largest, int *second_largest);
   ```

   When passed an array `a` of length `n`, the function will search `a` for its largest and second-largest elements, storing them in the variables pointed to by `largest` and `second_largest` respectively.

7. **K.K.11.7:** Write the following function:

   ```
   void split_date (int day_of_year, int year, int *month, int *day);
   ```

   `day_of_year` is an integer between 1 and 366, specifying a particular day within the year designated by `year`. `month` and `day` point to variables in which the function will store the equivalent month (1-12) and day within that month (1-31).

8. **K.K.11.8:** Write the following function:

   ```
   int *find_largest(int a[], int n)
   ```

   When passed an array a of length n, the function will return a pointer to the array's largest element.