CSC 209 Review 4

August 19, 2020

1 Exercises

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

```
1  void avg_sum(double a[], int n, double *avg, double *sum)
2  {
3      int i;
4
5      sum = 0.0;
6      for (i = 0; i < n; i++)
7          sum += a[i];
8      avg = sum / n;
9  }
```

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.

2 Programming Projects

1. Modify Programming Project 7 from Chapter 2 so that it includes the following function:

   ```
   void pay_amount(int dollars, int *twenties, int *tens, int *fives, int *ones);
   ```

   The function determines the smallest number of $20, $10, $5 and $1 bills necessary to pay the amount represented by the `dollars` parameter. The `twenties` parameter points to a variable in which the function will store the number of $20 bills required. The `tens`, `fives` and `ones` parameters are similar.

2. Modify Programming Project 8 from Chapter 5 so that it includes the following function:

   ```
   void find_closest_flight(int desired_time, int *departure_time, int *arrival_time);
   ```

   This function wil find the flight whose departure time is closest to `desired_time` (expressed in minutes sinces midnight). It will store the departure and arrival times of this flight (also expressed in minutes since midnight) in the variables pointed to by `departure_time` and `arrival_time`, respectively.

3. Modify Programming Project 3 from Chapter 6 so that it includes the following function:

   ```
   void reduce(int numerator, int denominator, int *reduced_numerator,
   int *reduced_denominator);
   ```

   `numerator` and `denominator` are the numerator and denominator of a fraction. `reduced_numerator` and `reduced_denominator` are pointers to variables in which the function will store the numerator and denominator of the fraction once it has been reduced to lowest terms.

4. Modify the `poker.c` program of Section 10.5 by moving all external variables into main and modifying functions so that they communicate by passing arguments. The `analyze_hand` function needs to change the straight, flush, four, three and pairs variables, so it will have to be passed pointers to those variables.