Instructions. There are no special instructions for these practice problems.

1. Consider the following class declaration.

```cpp
class T {
private:
    int x, y;
public:
    void set (int a, int b) {x = a; y = b;}
    void foo () {int temp = x; x = y; y = temp;}
    int bar (T t) {return x * t.y - t.x * y;}
    void display () {cout << x << '"' << y << endl;}
};
```

What does the following program print?

```cpp
void main () {
    T t1, t2;
    t1.set(2,3);
    t2.set(10,4);
    cout << t1.bar(t2) << endl;
    t2.foo();
    t2.display();
}
```

2. Consider the following array declarations.

```cpp
float a[100];
int b[200], c[300];
```

(a) Write a code fragment that finds the average of all the entries in the array `a`. Store the average in the (float) variable `average`.

(b) Write a code fragment that sets all the entries of the array `b` to 0.
(c) Write a code fragment that prints out the entries of the array c as follows: \(c[0], c[2], c[4], c[6], \ldots, c[296], c[298]\). In other words, all the entries with even index are printed. They should be printed one per line.

3. Write a function called frequency that returns the number of time an integer \(n\) appears in an array \(a\) of length \(size\).

   ```
   int frequency (int a[], int size, int n) {
   }
   ```

4. Write a function called equal that returns 1 if two arrays \(a\) and \(b\) each of length \(size\) are equal. (Two arrays are called equal if their entries are equal for every index.) Otherwise, the function returns 0.

   ```
   int equal (int a[], int b[], int size) {
   }
   ```

5. Text, p. 412, #2.

6. Perform three passes of selection sort on the array below.

   ```
   57 -17 13 0 60 41 67 5 70 -11 71 83 97 101 123
   ```

7. Perform three passes of insertion sort on the array in the previous problem.

8. Perform function split on the array below.

   ```
   0 11 9 3 2 6 -7 13 23 19 17 -1 -8 -5 -3
   ```

9. Write a function called getLetter that returns the \(n\)th small letter of the alphabet. For example getLetter(11) = ’k’.

   ```
   char getLetter (int n) {
   }
   ```

10. Write a function called copy that copies \(str1\) into \(str2\).

    ```
    void copy (char str1[], char str2[]) {
    }
    ```

11. Write a function called concatenate that appends the string \(str2\) to the string \(str1\).

    ```
    void concatenate (char str1[], char str2[]) {
    }
    ```