Christopher Wilcox
Assignment 1
CS368 Spring 2010
Due: 2010/02/03

1. Give a declaration for
  * The character variable user_input, with an initial value of the letter 'a'.

```
char user_input = 'a';
```

  * The character variable letter_k, with the constant value 'K'.

```
const char letter_k = 'K';
```

  * An array called scores of 300 double values.

```
double scores[300];
```

  * A structure with tag student with 3 structure members:
    the integer ID, the integer major, the double gpa

```
struct student{
    int ID;
    int major;
    double gpa;
};
```

  * An array of 2000 student structures called honors.

```
student honors[2000];
```

2. How many array elements result from the following array declaration?
   char errormsg[] = "Bad input.\n";

    The above char array contains the below.
    'B', 'a', 'd', ' ', 'i', 'n', 'p', 'u', 't', '.', '\n'
    There are 11 elements in the char array errormsg.

3. Write a C++ code fragment (not a declaration) that initializes the honors array, such that the ID for each student is set to 100000 + the array index value, the major for each student is set to 20, and the gpa for each student is 0.0.

```
int i;
int arrSize = sizeof(honors)/sizeof(*honors);
for(i=0; i<arrSize; i++){
    honors[i].ID = 1000 + i;
    honors[i].major = 20;
    honors[i].gpa = 0.0;
}
```

4. The wayward programmer has a program containing the following code:

```
double weights[] = {16.2, 85.0, 76.3, 13.4, 18.1, 18.1,
                    19.9, 26.55, 88.1, 51.0};

for ( i = 0; i < 15; i++) {
    weights[i] = weights[i+1] - 12.6;
}
```

* What do you expect to happen when the program containing this code is executed?

I expect that for the first 9 elements of the array, all will execute as expected. For the following 6 nonexistent/out of bounds elements, the 'wayward programmer' is in for a surprise, as the 'wayward programmer' has just left the boundaries of their declared array.

* At what point in its execution does this code do something wrong?

when i>= 9, the array of doubles named weights runs out of elements. When the line weights[9] = weights[10] - 12.6; is executed, the non existent element weights[10] will be retrieved without error, potentially giving the user bad data.

* When executed, does this program behave differently than the the closest equivalent Java code?

This code reacts very different on c++ from the equivalent java code. In java, you would receive an 'array/index out of bounds exception'. In C++, the language does no checking on your indexing. Because of this, you can address 17 elements of a 10 element array without receiving an error message or exception.