CS 115 Exam 2 Review Quiz

Oct. 28, 2009

Rules

- You must briefly explain your answers to receive partial credit.
- When a snippet of code is given to you, you can assume
  - that the code is enclosed within some function, even if no function definition is shown
  - that the `main` function is properly defined
  - that the `iostream`, `algorithm`, `fstream`, `iomanip`, `string`, and `cmath` libraries have been included at the beginning of the program.
- When you are asked to write *a snippet* of code, you may assume
  - that your code is enclosed within some function
  - that any necessary libraries have been included.
- When you are asked to write *a complete program*, you must write the `#include` statements, the `int main()`, etc. in your solution to receive full credit.
- A line consisting solely of "..." represents one or more unspecified C++ statements, some of which may change the values of program variables.
- You are encouraged to use the backs of these pages for scratch paper. If you want answers written there to be graded, they must be very clearly labeled and also noted on the main test, e.g. "See the back of page 1 for 3a."

Problem 1: 25 points.

(a) What does this snippet of code print?

```
for (int i = 1; i < 3; i++) {
     for (int j = 1; j <= 2; j++) {
          cout << i+j << " ";
     }
     cout << endl;
}
```

(b) What does this snippet of code print?

```
int a[6] = {5, 8, 7, 9, 13, -1};
cout << a[4] << endl;
```

(c) If the following function is defined somewhere in the program and prototyped above main….

```
int square(int& x) {
    return x*x;
}
```

…what does the following code print?

```
int x = 5;
int y = square(x);
cout << x << endl;
cout << y << endl;
```

(d) For the snippet of code…

```
float f[50];
```

…what is the datatype of `f[15]`?

(e) If the following function is defined somewhere in the program and prototyped above main….

```
int my_function (int x, float array[]) {
     array[1] = 2.5;
     return x*2;
}
```

…what does the following code print?

```
int x = 5;
float a[3] = {0.2, 0.4, 0.6};
int y = my_function(x, a);
cout << x << endl;
cout << y << endl;
cout << a[1] << endl;
cout << a[2] << endl;
```

Problem 2: 25 points.

The snippets of code in this problem do not successfully accomplish the task described in their accompanying comment. Correct the code so that it performs the task described in the comment. The code may have more than one error. **Make your corrections clear and unambiguous.**

(a)

```
/* Function that finds and returns the smallest
element of an integer array.  Inputs are the array and
its size */
int FindMin(int[] array, int size) {
     int min = 0;
     for (int i=0; i <= size; i++) {
          if (i < min) {
               min = array[i];
          }
     }

}
```

(b) Assume that the 2D array

```
int x[100][5]
```

has already been declared and defined.

```
/* Prints out all the elements of the 2D array X */
for (int i=0; i<=5; i++) {
     cout X[i][j];
}
```

(c)

```
/* Function sets every element of the array arr to
init_value. Size = size of the array. */

int InitArray(int[] arr, int init_value, int size) {
     for (int i=0; i<= size; i+1) {
          cout << init_value;
     }
}
```

Problem 3: 25 points.

Write short snippets of code to accomplish the following tasks:

(a) For an array that has been declared as

```
float floatArr[5][8];
```

write a snippet of code that computes and prints the average of all the elements in the **entire array**.

(b) Write a snippet of code that repeatedly asks for the user's input as a character. If the user's input is `'u'`, your code should call the function `Update()`, which has no inputs. If the user's input is `'p'`, your code should call the function `Print()`, which has no inputs. If the user's input is `'q'`, you should print a goodbye message and stop printing the menu. If the input is anything else, you should print an error message and reprint the menu.

(c) Write a function called `GetLargestOdd` with the following properties:

a. Parameters:
    i. `a`, an array of integers
    ii. `N`, an integer (the size of the array)
b. Return value: an int
c. Description:
    i. Goes through the array in sequence, from the first element to the last
    ii. Returns the largest *odd* element of the array, ignoring elements that are even numbers
    iii. If the array doesn't have any odd elements, returns 0

Problem 4: 25 points.

For this problem, you must write a **complete program** that contains the following:

- A function, defined below the main function, called `GetIntegers` with the following properties:
  - Parameters:
    - `inputArr`, an array of integers
    - `N`, an integer (the size of the array)
  - Return value: a bool
  - Description:
    - Prompts the user to enter 1000 integers.
    - Prompts for each integer individually
    - Returns `false` immediately if the user enters an invalid integer
    - Returns `true` if all of the integers were filled in correctly

- Prototype for `GetIntegers`

- A function, defined below the main function, called `AllEqual` with the following properties:
  - Parameters:
    - `inputArr`, an array of integers
    - `N`, an integer (the size of the array)
  - Return value: The function should return a `bool` whose value is `true` if *all* of the elements of the array are equal and `false` otherwise.

- Prototype for `AllEqual`

- A `main` function that does the following:
  - Declares **an array** of 1000 integers
  - Calls `GetIntegers` to initialize the array
  - If the user entered an invalid integer, prints an error message and exits
  - Otherwise, calls `AllEqual`. Based on the result of `AllEqual`, prints "All elements = [num]" if all of the elements are equal, where [num] is the number that they are all equal to.