CS 115 Exam 2, Spring 2011

Your name: _____

Rules

- You may use one handwritten 8.5 x 11" cheat sheet (front and back). This is the only resource you may consult during this exam.
- Explain/show work if you want to receive partial credit for wrong answers.

Grade (instructor use only)

	Your Score	**Max Score**
Problem 1		35
Problem 2		25
Problem 3		40
Total		100

Problem 1: 35 points.

What will print to the screen when each of the following snippets of code is executed?

Assume that each snippet of code is inside `main()` and that all necessary libraries have been included.

Be sure that spaces and line breaks are indicated clearly in your answers.

(a)

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

(b)

```
for (int i = 1; i <= 5; i++) {
   cout << i << " ";
}
```

(c)

```
for (int i = 1; i < 5; i+=2) {
   cout << i << " ";
}
```

(d)

```
for (int i = 1; i < 5; i++) {
   cout << i+2 << " ";
}
```

(e)

```
for (int i=4; i >= 0; i--) {
   for (int j=0; j <= 2; j++) {
      cout << j << " ";
   }
   cout << endl;
}
```

(f)

```
string s[4] = { "john", "george", "paul", "ringo" };
cout << s[3];
```

(g)

```
string s[4] = { "john", "george", "paul", "ringo" };
for (int j=1; j <= 3; j++) {
     cout << s[j] << endl;
}
```

Problem 2: 25 points.

Write code to accomplish the following tasks. Each task should be just a few lines of code.

(a) Define a function that meets the following specifications:

- has 2 input parameters: an array of integers and the size of the array
- returns TRUE if all of the array elements are odd numbers and FALSE otherwise
- does NOT contain any cin or cout statements

(b) Define a function that meets the following specifications:

a. has 2 strings as input parameters
b. does not return anything
c. prints out the first string followed by the second string
d. does not have any cin statements

(c) Write a snippet of code (NOT function) that creates an array of 10000 integers and initializes them to the numbers -10000 through -1.

Problem 3: 40 points.

For this problem, you must write a **complete program**. To write *a complete program*, you must write the `#include` statements, the `int main()`, etc. in your solution to receive full credit. You may use return statements in this program.

Read the instructions carefully before you start coding!
Your program should do the following:

1. A function, *defined below the main function*, called `SetPrices`. This function has the following properties:
    - Parameters:
        - `arr`, an array of floats
        - `size`, the size of the array
        - `start`, a floating-point number
    - Return value: none
    - Description: set the first array element equal to `start`, and set each of the following elements equal to the previous element + 0.75 (so the second element is 0.75 more than `start`, etc.)

2. A function, *defined below the main function*, called `CanAfford`. This function has the following properties:
    - Parameters:
        - `arr`, an array of floats
        - `size`, the size of the array
        - `budget`, a floating-point number
    - Return value: a floating-point number
    - Description: return the most expensive (largest) array element whose value is less than or equal to `budget`. If no array elements are less than or equal to `budget`, return 0. You may assume that the array is sorted in ascending order.

3. Prototypes for `CanAfford` and `SetPrices`

4. A main function that does the following:
    - Declares an array of 50 floats
    - Asks the user for a single floating-point number
    - Exits if the user did not enter a valid positive number
    - Otherwise, calls `SetPrices` to initialize the array with values starting at the user's number
    - Does the following until the user enters an invalid or negative number:
        - Asks the user for the amount of money in their wallet
        - Calls `CanAfford` to print out the price of the most expensive item the user can afford. It should not call `CanAfford` if the user's input is invalid or negative.