Dot Product Assignment (dot)
CS 330 - Fall 2013 - Ms. Katz

Due
5pm, Friday, September 6th (for first complete version)

Goals
- to remind you of our lab environment
- to build your C++ skills (functions, parameters, vectors, compiling)
  I realize C++ may be a new language for you; it's okay; you'll be fine
- to show off your best programming style

Overview
This is a straightforward assignment to write a simple program in C++ that computes the dot product of two vectors. Be sure to meet my requirements. You must complete this on your own, but you may ask others about the problem and our system, as well as C++. You need to understand well enough to be able to handle later assignments.

You must complete this assignment to my satisfaction to pass this course.

I'll give you feedback on this assignment. You'll make modifications until it meets my requirements. The due date is for your first complete attempt.

Input and Output Specification
All input is from standard input, cin. The first input is one integer representing the length of two vectors. Both vectors are the same length. That integer should be positive and no larger than 100. If it is not, the program should print an error message and quit. Otherwise, the program should read values of type double to fill the two vectors. If there are not enough values, fill with zero. (Hint: To fill with zero when you create the vectors, use 0 as a second parameter to the constructor.) Ignore any extra values.

After input, print a blank line. Then, print each vector on one line with a label at the front. Label each element with its index and have two spaces between elements. Print all real numbers with one digit after the decimal point. See below for a function to do that. You have to write the code to print the index; it is not built-in. An output line for a vector would look like:

v1: [0] 2.5  [1] -3.0  [2] 1.2

Then print the dot product of the two vectors. The dot product is the sum of the products of the elements with the same index. The vectors must be the same length. The summation formula above is another representation of the dot product. Label your answer.
Other Requirements
Your program should be submitted to my **katz330** account by **5pm, Friday, September 6th**. Submit it as the dot assignment. The file should have a `.cpp` ending. The compiler takes other endings, but we'll use .cpp this term.

Use the C++ compiler in our Linux lab with the -Wall option. The program should compile cleanly with no errors or warnings.

You must use functions throughout this program. Use parameters and pass them appropriately. If a parameter shouldn't change, make sure it doesn't. Each function should do one thing well (have high cohesion).

You must use the vector class from the Standard Template Library. Its `size()` member function returns an unsigned (not int) value. Do not declare bigger vectors than you need. Note that vectors do not check whether you go outside their declared bounds.

Declare storage as locally as possible to complete the task. Do not declare global variables. You should, however, use the std namespace.

Use good programming style including a summary comment with your name at the top. Each function should have a short summary comment describing what it does in terms of its parameters. That comment should be on the function prototype above the main function. The function implementation below main does not need a comment. Put your name and the month of creation, a description of the problem, and a description of the inputs and outputs in a comment at the top. Use consistent indenting that reflects the structure of the code. Take the time to do this assignment neatly and correctly.

**This assignment will be graded on a zero/one basis.** It must be completed to my satisfaction to earn a one which is then, conceptually, multiplied by your other lab grades. In other words, you have to do this, I may ask you to make changes, and I'll keep asking until you get it right.

Output Digits
The easiest way I've found for setting how many digits to use after the decimal point when printing real numbers is to use my `setOutputDigits` function. You'll need to put this function in your code and then call it once before printing to set how many digits to use throughout the rest of the execution. Note that you can copy and paste this code from the web page.

```cpp
// set pretty output format and number of digits after decimal
void setOutputDigits(int digits) {
    cout.precision(digits);
    cout.setf(ios::fixed | ios::dec | ios::showpoint);
}
```