



So Long, and Thanks For All The C++ Advice At The UCF High School Programming Tournament



Contest Conventions:

Each problem in this contest will have a file-name associated with it (printed at the top of the problem's specification, under the title). This file-name is used to name the file containing your solution's source code and the file you read input from. For example, if a problem's name is `problem`, your solution should be in a file named `problem.cpp` and read the problem's input from a file named `problem.in`. If you do not know how to open a file for input using C++ streams, this document will explain below. Your solution's output should be sent to the standard output stream (`cout` – please see below if this is unfamiliar). Any other input or output could result in your solution being judged incorrect, so stick to these guidelines. If you are a Java™ or ANSI C programmer, please see the corresponding version of this document.

Quick File I/O Tutorial:

To read in a file, you will construct an `ifstream` object. Here is template code:

```
#include <iostream>
#include <fstream>
using namespace std;

ifstream in("problem.in" );
```

To extract data from the stream, use the `>>` operator. Note that this operation uses whitespace in the input as the delimiter.

```
int i;
double d;
string s;

in >> i;
in >> d;
in >> s;
// in >> i >> d >> s; would have the same effect
```

You can read a line of text at a time from the stream using the `getline` function:

```
string line;
getline( in, line );
```

When you reach end-of-file, `getline` will return a false value. Note that `getline` is declared in `<string>`, in the `std` namespace.

Output to the Standard Output Stream:

The C++ standard output stream is accessed through the `cout` identifier. Use the `<<` operator to produce output. It is declared in `<iostream>` in the `std` namespace. For more information on controlling stream output formatting, please see your C++ reference manual.

```
int i;
double d;
string s;
...
cout << i << " " << d << " " << s << endl; // endl or "\n"
```