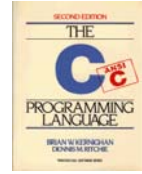




# C Programming, the UCF High School Programming Tournament, and Everything



## 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.c` 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 file operations, this document will explain below. Your solution's output should be sent to standard out (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 C++ programmer, please see the corresponding version of this document.

## Quick File I/O Tutorial:

To read in a file, you will first have to open it with the `fopen` function. Here is template code:

```
#include <stdio.h>
FILE *fp;

fp = fopen("problem.in", "r");
```

You can now read data from your file. You can read a line of text at a time using the `fgets` function:

```
char line[100];
fgets(line, sizeof(line), fp);
```

When you reach end-of-file, `fgets` will return a NULL value. Keep in mind that `fgets` will grab the entire line, including the new-line character (`\n`) at the end. Once you've read the line from the file, you can parse it using the `sscanf` function:

```
int x, y, z;
double d;

sscanf(line, "%d %d %d", &x, &y, &z); /* Read three integers */
sscanf(line, "%lf", &d); /* Read a real number */
```

You can also use any other C string functions (declared in `string.h`) to manipulate the input line. Consult your ANSI C documentation for information on how to use these functions.

When you're done reading from the file, you should close it to keep any spurious errors from happening:

```
fclose(fp);
```

## Output to the Standard Output Stream:

You should print to standard out using the `printf` function. In this contest, you should only print to standard out. Avoid writing to files or directly to the screen (don't include the `conio.h` header file).

```
int i;
double d;
string s;
...
printf("%d %0.3lf %s\n", i, d, s);
```