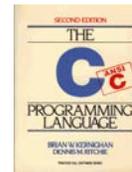




C Programming, the UCF Local Programming Contest, and Everything



Contest Conventions:

Each problem in this contest will have a filename printed at the top of the problem's specification, under the title. This filename must be used to name the file containing your solution's source code. For example, if a problem's filename is `problem`, your C solution must be in a file named `problem.c`. Your C source file may contain multiple functions but should contain only one `main` function, which should return 0 (zero). You may not submit more than one source file for a problem.

Your solution program must read the problem's input from the *standard input stream*. In C, this is the built-in file handle `stdin`, which is used automatically by `scanf` and certain other input functions. Correspondingly, your program must write its output to the *standard output stream*, for which C provides `stdout`, implicitly used by `printf` and certain other output functions. A few brief samples are provided below. 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. Do not submit C++ code in a `.c` file.

Example Use of the Standard Input Stream:

To use standard input/output, first include the header file with the necessary declarations:

```
#include <stdio.h>
```

You can now read data from standard in using the built-in `stdin` variable. For example, you can read a line of text at a time using the `fgets` function:

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

When you reach the end-of-stream, `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.

Example Use of the Standard Output Stream:

You should generate output using the `printf` function. 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);
```