Submitting Solutions

The USACO Training Program features an automatic grading system for your homework problems. You submit your programs from the problem page itself; they are compiled and graded; the results are conveyed back to you -- all within a few seconds.

C/C++/C++11, PASCAL, and Java are available. This system uses the GNU GCC compilation suite for C/C++/C++11 programs and the Free Pascal system for Pascal programs. The Java compiler is gjc as this is written and is soon to be upgraded.

The grading system compilers are those often used at the IOI.

These newer compilers uses 32 bit int's; the Borland compilers use 16 bit int's. DO NOT GET IN TROUBLE BECAUSE OF THIS!

Submit solutions via the web by typing the name of the file containing the source code into the 'Submit a file:' box at the bottom of problem description pages..

Program submissions require simple **Header comments**: your ID (i.e., your USACO login name), the name of the program (which will be given in each programming assignment, and the language used. See the examples below to get the idea.

Every training page problem has input and output. Currently, the input appears in a file named 'probname.in' (e.g., if the problem name is 'ride', then the input filename is 'ride.in'). Output must be written to a file named 'probname.out' (i.e., 'ride.out' for the 'ride' problem).

The First Challenge

The simplest programming challenge is named 'test' and requires you to read a pair of small integers from a single input line in the file 'test.in' and print their sum to the file 'test.out'.

Below is a simple solution in the 'C' programming language. Note the use of 'exit (0);', which is usually required to exit properly.

```
/*
ID: your_id_here
LANG: C
TASK: test
*/
#include <stdio.h>
main () {
    FILE *fin  = fopen ("test.in", "r");
    FILE *fout = fopen ("test.out", "w");
    int a, b;
    fscanf (fin, "%d %d", &a, &b);      /* the two input integers */
    fprintf (fout, "%d\n", a+b);
    exit (0);
}
```

Below is a simple solution in the C++ programming language. Note the use of 'return (0);', which is usually required to exit properly.

```
/*
ID: your_id_here
PROG: test
LANG: C++                    (<-- or C++11 if you prefer)
*/
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main() {
    ofstream fout ("test.out");
    ifstream fin ("test.in");
    int a, b;
    fin >> a >> b;
    fout << a+b << endl;
    return 0;
}
```

Below is a simple solution in the PASCAL programming language:

```
{
ID: your_id_here
PROG: test
LANG: PASCAL
}
Program Test;
Var fin, fout: text;
    a, b: word;
Begin
    Assign(fin, 'test.in'); Reset(fin);
    Assign(fout, 'test.out'); Rewrite(fout);
    Readln(fin, a, b);
    Writeln(fout, a+b);
    Close(fout);
End.
```

And here is the same program, this time in JAVA. Note that the program presumes the file opens will succeed, which they will:

```
/*
ID: your_id_here
LANG: JAVA
TASK: test
*/
import java.io.*;
import java.util.*;

class test {
  public static void main (String [] args) throws IOException {
    // Use BufferedReader rather than RandomAccessFile; it's much faster
    BufferedReader f = new BufferedReader(new FileReader("test.in"));
                                                  // input file name goes above
    PrintWriter out = new PrintWriter(new BufferedWriter(new FileWriter("test.out")));
    // Use StringTokenizer vs. readLine/split -- lots faster
    StringTokenizer st = new StringTokenizer(f.readLine());
                                                  // Get line, break into tokens
    int i1 = Integer.parseInt(st.nextToken());    // first integer
    int i2 = Integer.parseInt(st.nextToken());    // second integer
    out.println(i1+i2);                           // output result
    out.close();                                  // close the output file
  }
}
```

Important: BufferedReader and StringTokenizer are far more efficient than many other schemes for reading input. They can make a huge difference in the efficiency of your program! Use them!

You can try as many different things (subject to the caveats below) as you like to see how the grading system works. Theoretically, you can't break it or crash it. If you see a problem, please let me know.

The restrictions are few:

- One second runtime limit unless other specified (700 MHz Pentium III)
- About 16MB datasize limit
- About 1MB stacksize limit
- Be sure your program exits with status 0
- Be sure you print complete lines (with terminating newline), not just a few words or numbers
- Don't use files other than the specified input, output, and auxiliary files
- Other common sense rules that need not be listed

The rules are simple:

- Don't try to cheat.
- **Don't just print the answers**, you must calculate them in your program. If you just print answers, your login ID might be removed.
- Don't try to look at other files on the system or use other schemes to break security
- Don't try to break common sense rules of privacy
- Please report anomalous behavior to me right away (<kolstad@delos.com>)
- Have as much fun as possible
- Earn a trip to the IOI and other exotic contests!

Some hints:

- Both stderr and stdout are returned to you when errors occur
- Feel free to ask questions and send in comments
- Your reported output has `_'s substituted for spaces
- Include this comment if you use try/catch/throw in C++: `/*pragma handle-exceptions*/`

Compiler comments (please send in new compiler comments as you find them):

- We're using g++ (a.k.a. djgpp on PCs), Free Pascal, and gjc
- In C/C++, ints are 32 bits (char is 8; short is 16; long is 32; long long is 64)
- some libraries have new names; some have different or missing functions
- stricmp doesn't exist; use strcmp for string compares
- strrev does not exist
- neither itoa nor ltoa exists (use sprintf instead)
- No need for *huge* declarations - pointers already go everywhere
- Pascal users: be sure to "close" your output file or the output might not appear

Give it a try! Submit one of the programs above, using your own ID. Mouse it off into a file then type in the name of the text file that contains the source:

Submit a file: 选取文件 未选择文件 Send it in!