SEng 265 — Introduction to Software Engineering
Fall 2002
Assignment No. 3

Note 1 This assignment is to be done, optionally, in teams of 2.

Note 2 Different implementations of the C compiler tend to vary slightly. Your assignments will be tested on Software Engineering's main server (SENG's aserver.seng.engr.uvic.ca), and it is recommended that you do your assignments either on the SENG workstations (located in ELW A321) or using or putty (see the resources web page) to connect to aserver. If you use some other Unix system, you may get different results; this will result in mark deductions.

- Due date: Friday, Nov. 6, 2001, at the beginning of the class.
- This assignment is worth 20 points.
- This assignment is worth 5% of your total course mark.
- Clearly mark student numbers on all submissions.

Objectives

After completing this assignment, you will have:

- the ability to create, compile and run small C programs;
- learned basic C constructs: simple and complex data types (including arrays, and strings), control statements, input and output, to name a few.

Introduction

Single elimination tournaments are common across sports. A single elimination tournament starts with 2^r teams (where r is the number of rounds). In each round each surviving team plays against another surviving team, and the winners of these matches pass to the next round. Matches are played until there is one team left, which is crowned champion of the tournament.

A tournament with n teams should consist of $r = log_2(n)$ rounds (hence $n = 2^r$) with a total of $n - 1$ matches.

For example, the second part of the Ice Hockey tournament during the Salt Lake City Olympics consisted of 8 teams (Sweden, Belarus, Finland, Canada, United States, Germany, Czech Republic, Russia). After the first round Belarus, Canada, the United States, and Russia survived; Canada, and the United States passed to the final match, with Canada winning the gold.

The following in an example of how that tournament is encoded:

```
8
Sweden
Belarus
Finland
Canada
United States
Germany
Czech Republic
Russia
3,4
```

```
1,2
5,0
0,1
1,7
3,2
5,2
```

The first line corresponds to the number of teams in the tournament (n), followed by the names of each of the teams. Finally, there is a list of results for each match. The first $n/2$ results correspond to first round matches, the next $n/4$ correspond to the second round, the next $n/8$, correspond to the third round, etc, until the round consists of only one match (the final).

For the first round, $team_1$ plays against $team_2$, using the first result; $team_3$ plays $team_4$ using the second result, etc. For a given match, the first number in the result corresponds to the score of the first team, and the second number to the score of the second team. Obviously, whichever teams scores more, wins. No ties are allowed. In this particular example, the results of the first round are:

- Belarus defeats Sweden by score 4 3 (Sweden, the first team, plays against Belarus, the second team; the input score is $3, 4$, hence Sweden scores 3 and Belarus scores 4, with Belarus winning the match).
- Canada defeats Finland by score 2 1.
- United States defeats Germany by score 5 0.
- Russia defeats Czech Republic by score 1 0.

For the second round, the order of the teams should be preserved from the previous round (Belarus, Canada, United States and Russia survive, in that order). Hence the results are:

- Canada defeats Belarus by score 7 1
- United States defeats Russia by score 3 2

The final match is (between Canada and United States):

- Canada defeats United States by score 5 2

Canada is therefore the champion of the tournament.

Your Task

Write a program called `single` that determines the winners of each match and the ultimate champion. `single` will print (to standard output) the following output for the previous example:

```
Round 1:
Belarus defeats Sweden by score 4 3
Canada defeats Finland by score 2 1
United States defeats Germany by score 5 0
Russia defeats Czech Republic by score 1 0
Round 2:
Canada defeats Belarus by score 7 1
United States defeats Russia by score 3 2
Round 3:
Canada defeats United States by score 5 2
Champion: Canada
```

Implementation Notes

1. You should call your C program `single.c`
2. You should follow our coding standards and comment your code.
3. Divide your code into functional units.
4. Your program should take as a parameter a filename. That filename will contain the tournament data. That is, if we have a file called `results.txt`, then we should run your program in the following way:

   ```
   ./single results.txt
   ```

   If the input file cannot be open nor read, the program should print an error to standard error and terminate with exit status 1. No data is read from standard input.
5. Your program should be able to handle tournaments of a minimum of 2 teams, to a maximum of 256 teams.
6. Your program can assume that the format of the input is perfect.
7. When successfully run, `single` should exit with exit status 0.
8. Your program should compile with the `gcc` compiler without errors or warnings when the following options are specified:

   ```
   -Wall -pedantic -ansi
   ```

What to submit

Submit your source code (`single.c`) via CVS, using the module of one of your team members.

1. You should submit your program in a directory called `assign3`
2. **Do not submit your binary (executable) program!**

Finally, at the start of the class on the deadline, **hand in a printed copy of your `cvs commit` script as well as a hard copy of your `single.c` file**.

Remember, no line in your source code should exceed 80 characters long!