Assignment 2

due 10/4 @ 9pm

Program Description

You are to write a sequential program, in C or C++, that simulates the sequence of messages that would be passed in a ***reduction operation*** that uses the binomial tree model of communication. In particular, the program would be called from the command line as follows:

```
reduce <n>
```

where `<n>` is a positive integer. The number `<n>` is the number of tasks, each of which starts out with a single value. The program produces on its standard output (the screen by default), a sequence of lines that describe the sequence of transfers that would take place in a reduction of `<n>` values by these tasks. If a task j sends a value to a task k, the line that must be output would be

```
task j sends a value to task k
```

The program must number the tasks 0,1,2,...,m-1 if the number of tasks is m, and it must assume that the task that stores the final result is task 0. For example,

```
reduce 4
```

should produce the output

```
task 3 sends a value to task 1
task 2 sends a value to task 0
task 1 sends a value to task 0
```

Because multiple tasks send values in parallel but the output is sequential, we need a rule that makes the output unique. The rule that the program must use is that if tasks j and k both send values to other tasks at the same time, then the larger of numbers j and k is written to output first. ***For example, it would be incorrect to output***

```
task 2 sends a value to task 0
task 3 sends a value to task 1
task 1 sends a value to task 0
```

because $2 < 3$.

Other Requirements

- The program should produce correct results for any number of tasks, not just powers of 2, and it must use the method described in class (which is the same as is described in the book and the lecture notes) when it is not a power of 2.
- The program is supposed to be immune to incorrect usage. This means that if the first argument is anything other than a positive number, it should detect this and exit with an error message.
- The program must run correctly on any cslab host.

Rubric

Your solution will be graded based on the following criteria.

- Correctness (70%): all requirements must be met to receive the full credit. This includes:
  - handling incorrect usage,
  - handling numbers that are not powers of 2,
  - outputing messages in the correct order
- clarity of documentation (10%)
- compliance with the Programming Rules requirements (20%).

Submitting the Homework

In these instructions, assume your program file is named `myprog.c`.

1. You will use the `submithwk_cs49365` command to submit this zip file. **To submit your file, type the command**[1]

   biocs

   ```
   $ /data/bioc/b/student.accounts/cs493.65/bin/submithwk_cs49365  -t 2  myprog.c
   ```

   The program will copy your `program` into the directory

   ```
   /data/biocs/b/student.accounts/cs493.65/hwks/hwk2/
   ```

   and if it is successful, it will name it `hwk2_username.c` and display the message, ‘‘`File hwk2_`*`username`*`.c successfully submitted.`’’ where `username` is your actual username.

   You will not be able to read this file, nor will anyone else except for me. But you can double-check that the command succeeded by typing the command

   ```
   ls -l /data/biocs/b/student.accounts/cs493.65/hwks/hwk2
   ```

   and making sure you see a non-empty file named `hwk2_`*`username`*`.c`.

2. ***You can do the preceding step as many times as you want. Newer versions of the file will overwrite older ones.***

[1] If you have modified your PATH variable to include the directory `/data/bioc/b/student.accounts/cs493.65/bin`, then you can just type `submithwk_cs49365`.