University of Toronto
Faculty of Arts and Science
December 2012 Examinations
CSC 209H1F
Duration: 3 hours
Aids allowed: Any books and papers.
No electronic aids allowed: No calculators, cell phones, computers, ...
To pass the course you must receive at least 35% on this exam.

Make sure you have all 11 pages (including this page). (Don't panic about the page count -- there's lots of space for answers.)

Answer *all* questions. Answer questions in the space provided. Answers not in the correct space will not be graded unless a note in the correct space says "see page ..." and the answer on that page is clearly labelled with the question number.

Be careful not to get stuck on some questions to the complete exclusion of others. The amount of marks or answer-space allotted does not indicate how long it will take you to complete the question, nor does the size of the answer-space indicate the size of the correct answer.

In general, in the C programming questions you can omit the #includes, and you can omit comments unless you need to explain something unclear about the functioning of your code.

Do not open this booklet until you are instructed to.

---

1. [10 marks]
Write a shell script which expects a number as its only command-line argument (e.g. if it is called "file", someone might type "sh file 3") and in a loop, keeps squaring and outputting the number until it exceeds 1000000.

For example, if the number were 3, the output would be:

```
3
9
81
6561
43046721
```

You can assume that any argument is numeric, but you do have to check the argument count.

[Sample solution]

---

2. [12 marks]

Write a shell script (in the "sh" programming language) which takes one or more file name arguments, and outputs the name of the file which has the most lines in the opinion of "wc -l". (In the case of a tie, output any one of the applicable file names.)

[Sample solution]

---

3. [8 marks]
The file "file" has the contents:

```
echo is a good command
cat is a bad command
```

a) What is the output of the following shell script?

```
if grep good file
then
    echo yes
else
    echo no
fi
```

b) What is the output of the following (highly unusual) shell script?

```
if `grep good file`
then
    echo yes
else
    echo no
fi
```

[Solutions]

---

4. [15 marks]
Write a C program which is a simplified version of the "test" command. Your program will support only -f (test if a file exists and is a plain file), the two numeric relational operators -lt and -eq (the other four are very similar, after all), and the two string relational operators = and !=. So argc needs to be 3 for -f and 4 otherwise.

You need to check the command-line syntax but you do not need to check that the numbers are actually numeric.

[Sample solution]

---

5. [12 marks]
Write the library function "fgets" in terms of getc().

(Note: This question might be hard; don't spend all of your exam time on it!)

[Sample solution]

6. [10 marks]
Write a C program to look at all files under the current directory, recursively, and look for a file named "squid". If it finds one, it outputs either the path name relative to the current directory, or how many subdirectories deep it is, whichever you find easiest. If no such file name is found, it outputs "not found".

[Sample solution]
[Another solution]

7. [15 marks]
Write a program which creates two child processes using fork(), each connected with a pipe so they can send data to the parent. One of the children writes the number "12" to the parent over its pipe. The other child writes the number "49" to the parent over its pipe. The parent reads both of these and adds them and outputs the sum to stdout.

Note: the children must execute in parallel; start them both before reading from either.

[Sample solution]

8. [12 marks]
Write a program which listens on port 3000 (AF_INET). When someone connects to this port, it outputs "hello". Then it accepts another connection (thus waiting until someone else connects). When a second person connects, it outputs "goodbye" to the first person and drops their connection; and outputs "hello" to the second person. It continues like this indefinitely, hanging up on each person when and only when the next person connects. The outputs are lines, terminated with a network newline (CRLF).

Your program may be unix-specific.
As mentioned on the exam cover page, you can omit the #includes.

[Sample solution]

9. [6 marks]
The following code attempts to open a file named "foo" on file descriptor 8:

```
if ((fd = open("foo", ...)) < 0) {
    ...
    ... exit ...
}
dup2(fd, 8);
close(fd);
```

What happens if we have file descriptors 0 through 7 already open and the above open() call returns 8? And how do you fix this bug? (Two answers required: explain the bug, and write the fixed code.)

[Sample solution]

End of exam. Total marks: 100. Total pages: 11.
