October 4, 1991 Name: ______________________

TA: ______________________

CS/CE 218 — Unix and C programming

Test

Before you begin, please take a moment to look over the entire test so that you can budget your time.

All questions refer to the version of HP-UX running on Zippy and Zaphod, which is essentially Unix System V.

English answers will be judged on their completeness and clarity. However, you do *not* need to fill up all the empty space on the page. Conversely, do not feel limited by the space on a page, since you may continue an answer on the back of a page if you wish.

problem	subject	points	score
1	Permissions, short answer	10	
2	I/O redirection	12	
3	value of pipelines	10	
4	processes and pipelines	15	
5	regular expressions	12	
6	pipeline for file names	10	
7	pipeline for a TA's grades	31	
Total		100	

This exam (only) counts for 10% of your final grade.

1. (10 points) Assume for purposes of this problem that each student in CS/CE 218 has an account (e.g., "cs218a00") that is a member of the group "cs218". Another group on the Zippy and Zaphod is "grad", whose members are computer science graduate students. The mode of the directory `/tmp` is 777.

   (a) Suppose that the file `/tmp/a` is owned by you and the group owner is cs218. Write a single command that sets the permissions of `/tmp/a` so that other members of group "cs218" are permitted to read `/tmp/a`, but members of the group "grad" are not allowed to read the file.

   (b) Write a single command that sets the permissions of `/tmp/b` so that only you can read `/tmp/b`, accounts in group "cs218" cannot write `/tmp/b`, and any account not in your group `can` write `/tmp/b`.

2. (12 points) Complete the following table. That is, for each file descriptor, show the file to which I/O is directed by the given commands. The first two lines are worked out as an example.

   Assume that the shell's standard input is coming from a file I, that the standard output is going to a file O, and that the diagnostic (error) output is going to a file E. (That is, the shell that is executing the commands in the table has as its standard input I, etc.)

command	standard input	standard output	diagnostic output
`cmd arg`	I	O	E
`cmd arg <f`	f	O	E
`cmd arg >f`			
`cmd arg 2>f`			
`cmd arg >&2`			
`cmd arg 2>&1`			

3. (10 points) Suppose that pipelines were not allowed in Unix. What problems would this cause?

4. Suppose that a process running on your terminal is in a state where nothing you type makes the system give any response. So suppose you log in to another terminal.

   (a) (3 points) What command(s) would you use to find out the name of the terminal to which you were logged in previously?

   (b) (12 points) Suppose that you were previously logged in to the terminal named "ptyp9". Write a pipeline that shows the process ids of *only* the processes you want to kill (i.e., those associated with terminal ptyp9).

5. Suppose that the contents of the file `numbers` are the following four lines (with spaces indicated by ␣):

```
20088001␣111␣21199␣20725
20008200␣111␣21200␣21199
20408201␣111␣23113␣21200
20000001␣111␣23269␣23113
```

   For each of the following, fill in the underlined space (_) so that the output of the command is just the third line above (`20408201␣111␣23113␣21200`), and no other lines.

   (a) (3 points)
   `grep '^..__' numbers`

   (b) (3 points)
   `grep '[^␣]*␣23113␣____$' numbers`

   (c) (6 points)
   `grep '[^␣]*␣11*␣____[1-3][1-3].'  numbers`

6. (10 points) Write a pipeline that prints, on standard output, just the names of the files that are directly under the current working directory. The format of the file names should be `file-name` not `./file-name`. Since only files directly under the current working directory should appear in the output of the pipeline, no file names such as `./dir/file` should appear. (Hint: in the output of the command `ls -l`, there are no tabs, and the names start in column 55.)

7. A teaching assistant (TA) you know has two grade files: one for labs and the other for quizzes.

   Suppose that the file `lab.grades` contains the following 4 lines:

```
1␣␣Jack,␣J␣␣8713␣10
2␣␣Kent,␣C␣␣2318␣␣5
3␣␣Beth,␣M␣␣3982␣13
```

   Each line has a number, a name, an abbreviated ID number, and a lab score. Notice that the file contains no tabs.

   The TA would like to generate two files: one (`forme`) for the TA's records, and one (`topost`) for students that does not contain student names. The TA executes the following two commands:

```
cut -c1-12 lab.grades > temp
cut -c13- lab.grades | tr -s "␣" | tr "␣" "^I" |
    paste temp - | sort +1b -2b | tee forme |
    cut -f2- | sort +0n >topost
```

   (In the above, ␣ represents a space character, and `^I` represents a tab character.)

   The following questions refer to the state of files after executing the above commands.

   (a) (3 points) List the contents of the file `temp`, showing tab characters as `^I` and blanks as ␣.

(The questions on this page continue the last question on the previous page.)

(b) (5 points) List the contents of the file `forme`, showing tab characters as `^I` and blanks as ␣.

(c) (5 points) List the output found in `topost`, showing tab characters as `^I` and blanks as ␣.

(d) (10 points) Using `sed` show this TA a simpler way of producing the files `forme` and `topost`. Your solution should do away with the file `temp` and should have only one pipeline that accomplishes the effect of the TA's two pipelines.

(e) (8 points) Write a pipeline that transforms `lab.grades` into a sorted list of lab scores (one per line), with the lowest score first. This pipeline should work on any file `lab.grades`, as long as the contents are in the format shown above. (That is, `echo '5\n10\n13'` is an unacceptable answer.)