

Solutions for
CSC 352 Mid-term Exam
Monday, October 12, 2015

A note about grading mistakes

I wish the TAs and I were perfect but we're not. We do make mistakes when grading. I encourage you to double-check our work, and let us know if you find any mistakes, or disagree with any of our grading decisions, including deduction amounts.

I do regrading on a problem by problem basis; a regrade request for a single problem does not trigger an audit of the full exam.

Unless there are exceptional circumstances I require regrading requests to be in person, not via email or IM.

The TAs are not authorized to change any grades, but you're welcome to seek a TA's opinion about a mark before coming to me for an official decision.

Before asking anybody to revisit any problem involving code, first type in your code and see what it does.

There's no deadline for requesting regrading aside from the filing deadline for final grades.

I hate to have to mention it but note that **all completed exams were scanned into a PDF**, making it easy to catch post-grading alteration/addition of answers. Don't be tempted to add a couple of characters to get a point or two back. If caught, you'll fail the class, and more; see the syllabus for details.

Problem 1: (14 points)

Mean: 11.0, median: 14.0

*For this problem you are to write a **Java program** named `argGrep`. Unlike `fgrep` and your own `mgrep`, which search files or standard input for a string, `argGrep` searches its second and following command-line arguments for occurrences of its first command-line argument. It prints arguments that match, one per line.*

```
import java.io.*;
public class argGrep
{
    public static void main(String args[]) throws IOException
    {
        BufferedReader in =
            new BufferedReader(new InputStreamReader(System.in));

        for (int i = 1; i < args.length; i++)
            if (args[i].contains(args[0]))
                System.out.println(args[i]);
    }
}
```

A few students processed standard input instead of arguments. Those solutions were graded for half-credit.

Problem 2: (14 points)**Mean: 10.8, median: 12.0**

Write a **C program** `firstx` that outputs the zero-based position of the first occurrence of an 'x' (lower-case) on standard input. If there is no 'x', `firstx` prints `-1`.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int c, count = 0;

    while ((c = getchar()) != EOF)
    {
        if (c == 'x') {
            printf("%d\n", count);
            exit(0);
        }
        count++;
    }

    printf("-1\n");
}
```

Problem 3: (9 points)**Mean: 4.1, median: 4.5**

Write a **bash script** named `mvqt` that when given an argument such as `alan-questions.txt` it creates a directory named `alan`, and moves `alan-questions.txt` into `alan/questions.txt`.

```
student=$(echo $1 | cut -d- -f1)
mkdir $student
mv $1 $student/questions.txt
```

Problem 4: (3 points)**Mean: 4.1, median: 4.5**

Write a **bash for-loop** that runs `mvqt` on each of the `*-questions.txt` files in the current directory in turn. (Note that "`mvqt *-questions.txt`" would process only the first argument.)

```
for i in *-questions.txt
do
    mvqt $i
done
```

Problem 5: (14 points)**Mean: 8.3, median: 9.0**

Write a **C program** named `oxo.c` that reads integers from standard input and for each integer outputs a square of the specified size with alternating Os and Xs.

```
#include <stdio.h>

int main()
{
    int size;

    while (scanf("%d", &size) == 1)
    {
        int count = 0;
        for (int c = 1; c <= size; c++)
        {
            for (int r = 1; r <= size; r++)
                putchar(count++ % 2 == 0 ? 'O':'X');
            putchar('\n');
        }
        putchar('\n');
    }
}
```

This problem was the most difficult one to grade consistently. Here are the two key elements of the rubric we used:

Doesn't form a square: -3

Doesn't alternate Os and Xs, or doesn't start subsequent squares with an O: -3

A few students overlooked the `scanf` that was given, assumed single-digit sizes, and used `getchar()` to read the digits. We didn't deduct for that misunderstanding alone.

I came to regret writing "For one point of extra credit, don't use "if" in your solution." I was simply trying to prompt use of the conditional operator, used in my solution above, but a number of students devoted a considerable amount of effort to getting that point, nearly filling the entire space given for the solution with nested switches and more.

If you do the math, you'll find that one point on a 45-minute, 100-point exam corresponds to 27 seconds. It's important to weigh points vs. time when deciding to pursue potential extra credit on an exam.

Here's one more thing about that point of extra credit: a substantially whole solution needed to be present to get that point—blank spaces or a few scrappy lines didn't (and won't) get a point just because there was no "if".

Problem 6: (9 points)**Mean: 6.9, median: 7.5**

On assignment 3 you wrote `clp.java`, a program that analyzed command lines for arguments, redirection, piping, etc. On this problem you are to do that same analysis by hand.

Command line:

```
% ls x y | wc -c | cat -n
```

Command 1: `ls`
Arguments: `x, y`

Stdin: `K`
Stdout: `P`

Command 2: `wc`
Arguments: `-c`

Stdin: `P`
Stdout: `P`

Command 3: `cat`
Arguments: `-n`

Stdin: `P`
Stdout: `S`

Command line:

```
% wc < a b > c d
```

Command 1: `wc`
Arguments: `b, d`

Stdin: `a`
Stdout: `c`

Command 2:
Arguments:

Stdin:
Stdout:

Command 3:
Arguments:

Stdin:
Stdout:

Command line:

```
% java java java | wc wc -pipe ls > cat
```

Command 1: `java`
Arguments: `java, java`

Stdin: `K`
Stdout: `P`

Command 2: `wc`
Arguments:
`wc, -pipe, ls`
Stdin: `P`
Stdout: `cat`

Command 3:
Arguments:

Stdin:
Stdout:

Command line:

```
% < a b | e f-g h | > x y z
```

Command 1: `b`
Arguments: `none`

Stdin: `a`
Stdout: `P`

Command 2: `e`
Arguments: `f-g, h`

Stdin: `P`
Stdout: `P`

Command 3: `y`
Arguments: `z`

Stdin: `P`
Stdout: `x`

Grading notes

There are a total of nine commands in the command lines above, and each command had four fields to complete, so each field was graded right or wrong for a quarter point.

If a command was overlooked, or a non-existing command was perceived, that was a one-point deduction per occurrence.

Problem 7: (12 points)

Mean: 6.5, median: 6.0

Each time the Tester is run it records when it was run, what was tested, and more. Here's a short sample of a simplified version of those entries:

```
% cat log
Wed Sep 23 20:11:06 MST 2015 args: patlen
Wed Sep 23 20:11:15 MST 2015 args: lpp
Wed Sep 23 20:35:58 MST 2015 args: patlen.java
Wed Sep 23 21:39:24 MST 2015 args: lpp
Wed Sep 23 22:18:34 MST 2015 args: lpp.java
Wed Sep 23 22:43:17 MST 2015 args: llp.java
Wed Sep 23 23:03:44 MST 2015 args: lpp
Wed Sep 23 23:15:38 MST 2015 args: patlen.java
Wed Sep 23 23:26:07 MST 2015 args: lpp.java
Wed Sep 23 23:58:23 MST 2015 args: lpp
```

Write a **one-pipeline bash script, rbh** (runs by hour), that reads lines like the above on standard input and outputs a count of how many runs were made in each hour of the day. For the file above, here's the output:

```
% rbh < log
3 20
1 21
2 22
4 23
```

Solution(s) and grading notes

```
cut -d" " -f4 | cut -f1 -d: | sort | uniq -c
```

The `sort` isn't need for the log sample, and wasn't required for full credit, but I include it for generality.

When writing the problem I overlooked a simpler solution, based on `cut`'s `-c` option:

```
cut -c12-13 | uniq -c
```

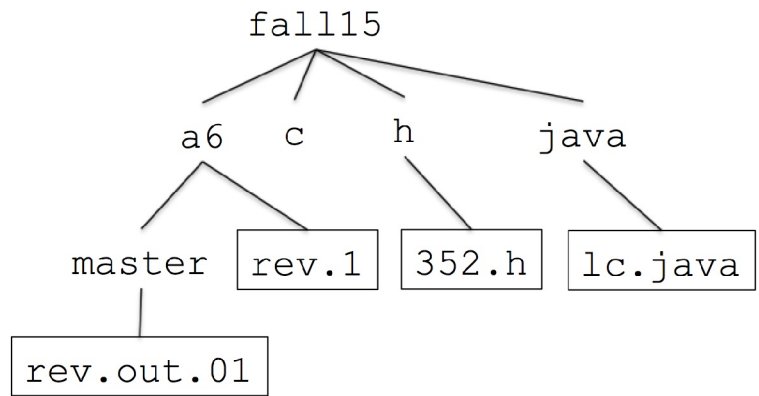
The problem was graded as having three four-point pieces: `cut`, `cut`, `uniq`, with appropriate arguments.

Problem 8: (8 points) (1 point each)

Mean: 4.0, median: 4.0

This problem is like the `tree-*` problems on assignment 3. Given the tree on the right, write a bash command line to answer each of the questions below.

Each question or problem starts with "In DIRECTORY, ..." meaning that you should assume that DIRECTORY is your current working directory.



IMPORTANT: You may use multiple commands to answer a question but **you may NOT change your working directory**. That is, an answer like "`cd ../h; wc -l *.c`" will be considered wrong.

Consider the word "file" to mean all type of entries—regular files, directories, and more. For example, "`ls | wc -l`" is a correct answer for "How many files are in the current directory?" even if the current directory contains subdirectories. Also, don't worry about hidden entries.

A few individual files, surrounded with a box, are shown in the tree above but **assume that each directory might have a number of files that are not shown.**

- (1) In `master`, how many lines are in `rev.1`?
`wc -l ../rev.1`
- (2) In `fall15`, what lines in `352.h` contain the string "define"?
`fgrep define h/352.h`
- (3) In `java`, what is the last line in `rev.out.01`?
`tail -1 ../a6/master/rev.out.01`
- (4) In `java`, how many `.pdf` files are in `fall15`?
`ls ../*.pdf | wc -l`
- (5) In `c`, how many files in `java` do NOT have a `'.'` in their name?
`ls ../java | fgrep -v . | wc -l`
- (6) In `fall15`, how many files in `c` have a name that both starts **and** ends with an `'a'`?
`ls c/a*a c/a | wc -l`

A number of students included `fgrep a*a` in their solution but (1) `fgrep` searches for a literal string, just like `a2's mgrep`, and (2) if `grep` was used, the regular expression `a*a` matches all lines with one or more `a`'s. (Yes, we haven't covered regular expressions!)

I'll be honest and admit that my initial solution didn't include `c/a`, and wouldn't have matched the single-letter name `a`.

- (7) In `master`, copy your `.bashrc` to the current directory using as few characters as possible.
`cp ~/.bashrc .`
- (8) In `master`, is `cd ../../fall15` valid? (If not, correct the path to reach `fall15`.)
`cd ../../`

Problem 9: (5 points)

Mean: 2.9, median: 3.0

For one point each, answer the following questions, all related to C.(1) In as few characters as possible, what is the value of `x++`?
x(2) Write an `if` statement that's valid in C but not in Java.
`if (1) {}`

Lots of other answers were valid, too. What I actually had in mind was an `if` that's syntactically valid in both C and Java but not semantically valid in C, but we accepted answers like `if (&a[0])`

We elected to mark answers like `if ((c = getchar()) != EOF)` wrong because they'd work if a `getchar()` method and an `EOF` variable were present, but I'm iffy on that decision now. I'll balance that with additional point of adjustment for all scores.

(3) What is the value of the following C expression? `10 > (20 > 30)`
1(4) What are the three aspects of a Java or C expression that whm says are usually the most useful to consider?
type, value, side-effects(5) On a 24-bit machine that has 12-bit bytes and uses one's complement arithmetic for computations involving chars, what is the most reasonable value for `sizeof(char)`?
`sizeof(char)` is always 1. Yes, always.**Problem 10: (12 points)**

Mean: 6.2, median: 6.0

For 1.5 points each, answer the following questions, all related to UNIX.(1) How can we quickly identify if a path is absolute or relative?
A path is absolute iff the first character is a slash.(2) Describe in English the file names that would be matched by the bash wildcard pattern `??[abc]??`?
Five character names whose middle letter is a, b, or c.(3) Same as previous question but for the pattern `T???*.?`?
Names of six or more letters that start with a T and whose next to last character is a dot.(4) What is one of the three files that bash always looks for when it is started as a login shell?
The three I mention in the slides are `~/.bash_profile`, `~/.bash_login`, and `~/.profile`, but when grading we were reminded that `/etc/profile` is read, too, and first.

Note that bash does not look for `.bashrc` when started as a login shell. That's why the suggested bash configuration in the UNIX slides has `source ~/.bashrc` in `~/.profile`.

Little known fact, `.` (just a dot) is a synonym for `source`. You could have `~/.bashrc` in your `.profile` instead.

- (5) The command `"ln -s $fall15/a6 a6"` creates an entry named `a6`. In two words or less, what is `a6`?
"symbolic link" or "symlink"
- (6) What does `bash`'s `PATH` variable specify?
The directories to search for a command that doesn't contain a slash.
- (7) Fill in the blank to make `wc` produce the output shown, regardless of the current directory contents.

```
% echo -n | wc  
0 0 0
```

Another correct answer that was popular was `cat /dev/null`. Just `echo` was incorrect because it outputs a newline.

Any command that didn't produce any output was counted as correct.

- (8) Imagine that one of `whm`'s `C` slides indicates that an example is in `switch2.c`. Exactly what command(s) would you type on `lectura` to see the contents of `switch2.c`? Assume your current configuration on `lectura` with `.bashrc`, `symlinks`, etc.

```
cat $fall15/c/switch2.c
```

We accepted any answer that was deemed plausible, even if debatable. For example `cat fall15/switch.c` was counted as correct because, although unlikely, one might have symlinked `fall15` to `/cs/www/classes/cs352/fall15/c`.

Extra Credit Section (½ point each unless otherwise noted)

Mean: 0.5, median: 0.0

- (1) The `C11` standard mentions "translation unit" in a number of places. What is a translation unit?

A translation unit is the final result of preprocessing. `gcc -E x.c` writes the translation unit for `x.c` to standard output.

- (2) Using `C`'s `while`, write an infinite loop in as few characters as possible.

```
while(1); (Here's one character too many: while(1){})
```

- (3) Write a version of `mvqt` (page 4) that handles names like `smith-jones-questions.txt`.

```
student=$(echo $1 | rev | cut -d- -f2- | rev)  
echo mkdir $student  
echo mv $1 $student/questions.txt
```

- (4) Who is widely recognized as "The Father of ASCII"?

Bob Bemer. He came up with lots of other stuff, too!

(5) *What is the "A1B2 Challenge"?*

See the a6 solutions!

(6) *Tell me something about Lester Moore.*

See UNIX slide 62.

I imagine Lester was unrelated to Gordon E. Moore, who wrote Moore's Law, but I am not certain of that.

(7) *Instead of doing "gcc x.c && a.out", a friend is doing "gcc x.c | a.out". What's a subtle problem your friend may encounter?*

All commands in a pipeline are started immediately. a.out might run before compilation is complete, possibly running an old version.

(8) *Who was the featured speaker at the College of Science Honors Convocation, on Friday, Oct 9, 2015 or, at what time did the convocation start?*

John Pollard, from Chemistry & Biochemistry, was the keynote speaker. Convocation started at 4:30pm. **Our TA Zac Nixon got the first award!**

Credit where credit is due: In a Facebook post, I bemoaned losing one of the 43 lectures I thought I had to work with this semester, writing "You can't say The University of Arizona doesn't value ceremony over substance!" In response, my cousin Gerry Williams suggested the question above.

Statistics

Here are all scores, in order:

102.00, 95.50, 94.00, 93.16, 93.00, 92.00, 91.00, 90.50, 90.50, 89.50,
89.16, 88.00, 87.66, 87.66, 86.50, 86.16, 85.50, 85.00, 85.00, 84.50,
83.50, 83.50, 82.75, 82.50, 82.00, 81.66, 81.00, 81.00, 80.50, 80.00,
80.00, 79.50, 79.00, 78.75, 78.75, 78.58, 76.83, 75.25, 74.50, 74.50,
74.25, 74.00, 73.66, 73.50, 71.75, 71.75, 71.50, 71.50, 71.00, 71.00,
70.50, 69.75, 69.50, 68.75, 68.16, 68.08, 67.75, 66.75, 66.50, 66.25,
66.00, 65.00, 64.33, 64.25, 64.00, 64.00, 63.75, 63.00, 63.00, 62.75,
62.50, 61.91, 61.50, 61.50, 60.75, 60.66, 60.33, 58.75, 56.25, 55.75,
55.50, 55.00, 55.00, 54.75, 54.75, 54.00, 54.00, 54.00, 54.00, 53.75, 53.41,
53.25, 52.50, 51.75, 50.00, 50.00, 49.50, 49.25, 48.25, 48.00, 47.75,
47.50, 47.25, 47.00, 45.83, 45.75, 45.50, 42.75, 42.75, 42.50, 40.25,
39.00, 37.50, 35.75, 33.91, 33.91, 33.50, 33.25, 32.75, 32.50, 30.75,
30.50, 30.00, 29.00, 28.00, 23.50, 22.50, 22.50, 17.75, 11.25

n = 129
mean = 62.6788
median = 64

Here are the numbers for only the 127A/B (or 227) only students, the "AB only" group. As mentioned in Piazza post 280, I equate ECE 175 and ISTA 130 with CSC 227.

90.50, 83.50, 82.00, 78.75, 78.75, 74.50, 69.50, 67.75, 66.50, 63.00,
61.50, 61.50, 55.75, 55.00, 55.00, 53.75, 53.41, 50.00, 50.00, 49.50,

48.25, 42.75, 37.50, 33.91, 32.75, 30.75, 30.00, 29.00, 28.00, 23.50,
17.75

n = 31
mean = 53.3652
median = 53.75

Below are the numbers for what I call the "AB plus" group—students who have completed at least one CS course beyond 127B or 227.

102.00, 95.50, 94.00, 93.16, 93.00, 92.00, 91.00, 90.50, 89.50, 89.16,
88.00, 87.66, 87.66, 86.50, 86.16, 85.50, 85.00, 85.00, 84.50, 83.50,
82.75, 82.50, 81.66, 81.00, 81.00, 80.50, 80.00, 80.00, 79.50, 79.00,
78.58, 76.83, 75.25, 74.50, 74.25, 74.00, 73.66, 73.50, 71.75, 71.75,
71.50, 71.50, 71.00, 71.00, 70.50, 69.75, 68.75, 68.16, 68.08, 66.75,
66.25, 66.00, 65.00, 64.33, 64.25, 64.00, 64.00, 63.75, 63.00, 62.75,
62.50, 61.91, 60.75, 60.66, 60.33, 58.75, 56.25, 55.50, 54.75, 54.75,
54.00, 54.00, 54.00, 53.25, 52.50, 51.75, 49.25, 48.00, 47.75, 47.50,
47.25, 47.00, 45.83, 45.75, 45.50, 42.75, 42.50, 40.25, 39.00, 35.75,
33.91, 33.50, 33.25, 32.50, 30.50, 22.50, 22.50, 11.25

n = 98
mean = 65.6249
median = 67.415

The scores are lower than I like to see, so I'll be adding some number of points to all scores, but before deciding on a number I want to look at correlation between mid-term scores and the averages for assignments, and that'll be after a 5 and a 6 grades are completed. At the moment I'm contemplating a 10-15 point addition to all scores, but don't make any big decisions based on that range.

I'm particularly bothered by the 14-point gap between the ABonly and ABplus medians. I wish there were a way to treat the two groups differently, like can be done when 400- and 500-levels of a class are co-convened. I don't imagine there's any option for that in this case—everybody is enrolled in the same class—but I'm going to look into it.