



## Assignment 5

### 1 Overview

This assignment is giving you more practice in writing regular expressions and using filters.

### 2 Tasks

This assignment consists of three tasks.

1. (40%) Write a script named `search_dna` that will expect on the command line a single argument which is the name of a file containing multiple DNA strings, each on a single line with no white space characters of any kind except for the terminating newline character at the end of the line. Each line is just a sequence of the letters `a`, `c`, `g`, and `t`. This script will print each line in which the 4<sup>th</sup> 3-letter sequence is repeated twice later in the line at starting positions that are multiples of three apart. For example, the string below, which is broken into groups of three letters for ease of reading

```
acg ttt ggg ccc agc tct ccg ccc tca cac aca ccc cgg ggt
```

satisfies this requirement, since `ccc` is the 4<sup>th</sup> three-letter sequence and it reoccurs twice at positions that are multiples of three apart, whereas this one does not:

```
acg ttt ggg ccc agc tct ccg gcc cca cac aca ccc cgg ggt
```

because, although `ccc` occurs twice later in the line, the first reoccurrence is not a multiple of three positions from the first one.

**Error checking:** This script must check that it is given a filename on the command line that can be read and issue a usage statement if there is an error of this type.

2. (30%) Write a script named `dollars` that prints all and only the lines in the standard input stream that contain monetary amounts of at least one dollar, expressed in dollars and cents. There is no upper limit on the amounts you may find in the file. Dollar amounts in excess of \$999.99 will have commas in the number, such as \$32,478,120.52 or \$906,201,777,528.10. Numbers should not have spaces or any other characters in them other than the dollar sign, comma, decimal point, and digits. Amounts should not be matched if they start with a leading 0, or are missing cents, or contain invalid characters. This program does not issue any error messages.
3. (30%) Write a script named `periods` that prints all and only the lines in the standard input stream that contain exactly 20 periods, with an arbitrary number of non-periods, possibly none, between each one. For example, the following two lines meet this criterion:

```
.. . . . . . . . . . . . . . . . . . . . .  
a.a.b.. sde 12.234.456...$$$,()..%^&(... .. <> ..z.z.
```



### 3 Rubric

This homework is graded on a 100 point scale. Each script is worth the points indicated next to it. Each script will be graded primarily on its correctness. This means that it does exactly what the assignment states it must do, in detail. Correctness is worth 70% of the grade. Then it is graded on its clarity, simplicity, and efficiency, as described above. Good comments are worth 15%; good design another 5%, and simplicity and efficiency the remaining 5%. Naming all files and directories correctly is 5%.

### 4 Submitting the Solution

This assignment is due by the end of the day (i.e. 11:59PM, EST) on Monday, October 29. (I give a grace period of six hours after that, so it is okay to submit it by 6:00 AM of the following day.)

There is a directory in the CSci Department network whose full path name is

```
/data/biocs/b/student.accounts/cs132/hwks/hwk5.
```

You must put it in that directory. To submit your project, you must follow the instructions below exactly! Do not deviate from these instructions.

To be precise:

1. Login using `ssh` to `eniac.cs.hunter.cuny.edu` with your valid username and password, and then `ssh` into any `cs132` host. Do not forget this step. You will not be able to run the `submithwk` command on `eniac`.
2. If you did not do the work on one of the computers in our network, then upload the two files into your home directory. Create a directory named `your-username-hwk5` and put the three scripts into it.
3. Run the command

```
zip -r your-username-hwk5.zip your-username-hwk5
```

This will create the file `your-username-hwk5.zip`. The `zip` command is a special command that compresses the files in the directory and creates a new file that can later be extracted by the `unzip` command. So it will create a “zip file” named `your-username-hwk5.zip` containing your `your-username-hwk5` directory and the three files it contains. For example, I would run

```
zip -r sweiss-hwk5.zip sweiss-hwk5
```

4. Run the command

```
/data/biocs/b/student.accounts/cs132/bin/submithwk 5 your-username-hwk5.zip
```

Do exactly this. Do not mistype it. The command will create a copy of the file `your-username-hwk5.zip` in the directory

```
/data/biocs/b/student.accounts/cs132/hwks/hwk5
```

It will be named `hwk5_username`, where `username` is your username on the network. You will not be able to read this file, nor will anyone else except for me. If you decide to make any changes and resubmit, just do all the steps again and it will replace the old file with the new one. I will be able to unzip the file, extracting whatever files you created. Do not try to put your file into this directory in any other way - you will be unable to do this.

5. **Do not put anything in this directory other than the three scripts. You will lose one point for each file that is in this directory that is not one of the three scripts.**

Although these instructions may seem complicated, they simplify the way you submit your work and the way I can retrieve it. If you make mistakes, just start over. If things don't seem to work out, post a question on Piazza with the details included.