Name: TODO
Course: CSCI 312 Principles of Programming Languages
Assignment Deadline: March 19, 2025

Question 1

The delete, yank, and put commands all interact with one of Vim's registers. We can specify which register we want to use by prefixing the command with `''{register}`.

1. How do you address the unnamed register? **""**
2. How do you address the yank register? **"0**
3. How do you address the named registers? **"a, "b, ..."y, "z**
4. How do you address the black hole register? **"_**
5. How do you address the system clipboard register? **"+**
6. How do you address the selection register? **"***
7. How do you address the expression register? **"=**
8. How do you address the register holding the name of the current file? **"%**
9. How do you address the register holding the last inserted text? **".**
10. How do you address the register holding the last Ex command? **":**
11. How do you address the register holding the last search pattern? **"/**

Question 2

A knowledge of the following terms, describing characteristics of an implementation, will aid in understanding what is and isn't acceptable in C. The first two are concerned with unportable code; the next two deal with bad code; and the last two are about portable code.

1. What is the definition of *implementation-defined*? **The compiler is required by C standard to determine what happens in implementation-defined cases.**
2. What is the definition of *unspecified*? **The compiler is not responsible for setting precedent nor rules for what happens in unspecified cases.**
3. What is the definition of *undefined*? **The response to code without definition, which is completely unpredictable because anything is allowed to happen.**
4. What is the definition of *a constraint*? **A rule for what leads to undefined behavior. The compiler catches anything that violates constraints based on the C standard.**

5. What is the definition of *strictly-conforming*? **Code that only uses C standard-defined features and works the same on any device.**
6. What is the definition of *conforming*? **Meets C standard requirements but varies between compilers due to implementation-defined choices.**

Question 3

What are the two meanings of the `static` keyword?

1. **Inside function: independent of class instance, or retains value between calls**
2. **Function-level: only readable inside the same file**

What are the two meanings of the `extern` keyword?

1. **Function Definition Application: global scope**
2. **Variable Application: acknowledges that definition is elsewhere (a different file)**

What are the three meanings of the `void` keyword?

1. **Return Type: returns nothing**
2. **Pointer Type: generic declaration**
3. **Parameter List: no parameters**

What are the three meanings of the * symbol?

1. **Multiplication operator**
2. **Indirect variable access via pointer**
3. **Pointer declaration**

Question 4

Make a new directory in your ppl repo called `Assignment1`. Create a symbolic link in your `Assignment1` called `linux` to `/home/mgwhite/homescratch/linux`.

1. What are the permissions of the symbolic link? **The owner, group, and public all have read, write, and execute permissions enabled (lrwxrwxrwx)**
2. What are the permissions of `/home/mgwhite/homescratch/linux`? **(drwxr-xr-x) – The owner may read, write, and execute while the group and public may only read and execute (where execute means entering access to the directory)**

Question 5

Use the command for estimating file space usage to estimate the file space usage of `/home/mgwhite/homescratch/linux`. What is the total in human readable format?
4.6G (gigabytes)

Question 6

You have been assigned a subtree in the Linux kernel source tree to analyze. Summarize the purpose of your subsystem. **My subsystem (linux/sound/) handles audio driver hardware and processes both digital and analog audio. Sound also offers API for users to interact with sound systems.**

Question 7

1. What is the *disk usage* size (in human readable format) of the largest .c file (in your subsystem)? And what is the pipeline you used?
   Disk Usage: 157K
   Command: `find -type f -name "*.c" -exec du -h {} + | sort -h | tail -n 1`

2. What is the *disk usage* size (in human readable format) of the smallest .c file (in your subsystem)? And what is the pipeline you used?
   Disk Usage: 512
   Command: `find -type f -name "*.c" -exec du -h {} + | sort -h | head -n 1`

3. How many lines in .c files (in your subsystem) use the `auto` keyword? And what is the pipeline you used?
   Number of Lines: 1472
   Command: `grep -r –include="*.c" "auto" . | wc -l`

Question 8

Produce a sorted list of .c files (in your subsystem) that use the `typedef` keyword. Store this list in a file called `Assignment1/typedef.txt`.
Complete

Question 9

Lexically analyze the .c files (in your subsystem) and produce a frequency distribution of lexical elements.

1. Store this list in a file called `Assignment1/frequency.txt`. Use the `lex.l` that I provided as a starting point. The infrastructure is in place. You simply need to refine the rules so the scanner accurately analyzes the .c files in your subsystem.
   Complete

2. You must also submit your "test suite" in a file called `Assignment1/tests.sh` that comprises all the test cases you ran on your output to validate your rules.
   Complete

Question 10

1. What is gcc? **GCC stands for GNU Compiler Collection. GCC is the compiler for both C and C++ programs, meaning is takes these programs through pre-processing, compilation, assembly, and lastly linking to create an executable file.**

2. Where is gcc located? **gcc: /usr/bin/gcc**