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? ””
3. How do you address the named registers? ”**a-**”**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? ”ˆ
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*? **In C language different compilers or platforms to define their own behavior. For example, the size of different data types can be implementation defined.**
2. What is the definition of *unspecified*? **Empty struct used as placeholder return type.**
3. What is the definition of *undefined*? **Behavior that does not act in a predictable way in C, like de-referencing a null pointer.**
4. What is the definition of *a constraint*? **Syntactic or semantic restriction by which elements are interpreted.**

5. What is the definition of *strictly-conforming*? **Code that is portable and that strictly follows C standards.**

6. What is the definition of *conforming*? **Code that is not strictly-conforming but the C standard claims no juridstiction over.**

Question 3

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

1. **In file scope, restricts visibility only to file where it was declared.**

2. **In block scope, restricts visiblity only to block where it was declared.**

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

1. **Tells compiler variable is in another file so that two files can share same global variable.**

2. **Tells compiler function is defined in another file.**

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

1. **This function returns no value.**

2. **This function takes no arguments.**

3. **This pointer has an unspecified type.**

What are the three meanings of the `*` symbol?

1. **Dereference- this represents the value stored at the address.**

2. **Pointer declaration.**

3. **Pointer to a 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? **100644 78012e10478055695de0b3748d90773f7d38a40a 0 linux**

2. What are the permissions of `/home/mgwhite/homescratch/linux`? **TODO**

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? **Four bytes.**

Question 6

You have been assigned a subtree in the Linux kernel source tree to analyze. Summarize the purpose of your subsystem. **Middleman between audio software and audio hardware, for sound plyaback, recording, mixing, etc.**

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? **TODO**

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? **TODO**

3. How many lines in .c files (in your subsystem) use the `auto` keyword? And what is the pipeline you used? **TODO**

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`.

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.

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.

Question 10

1. What is gcc? **GNU compiler collection**

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