Part 1

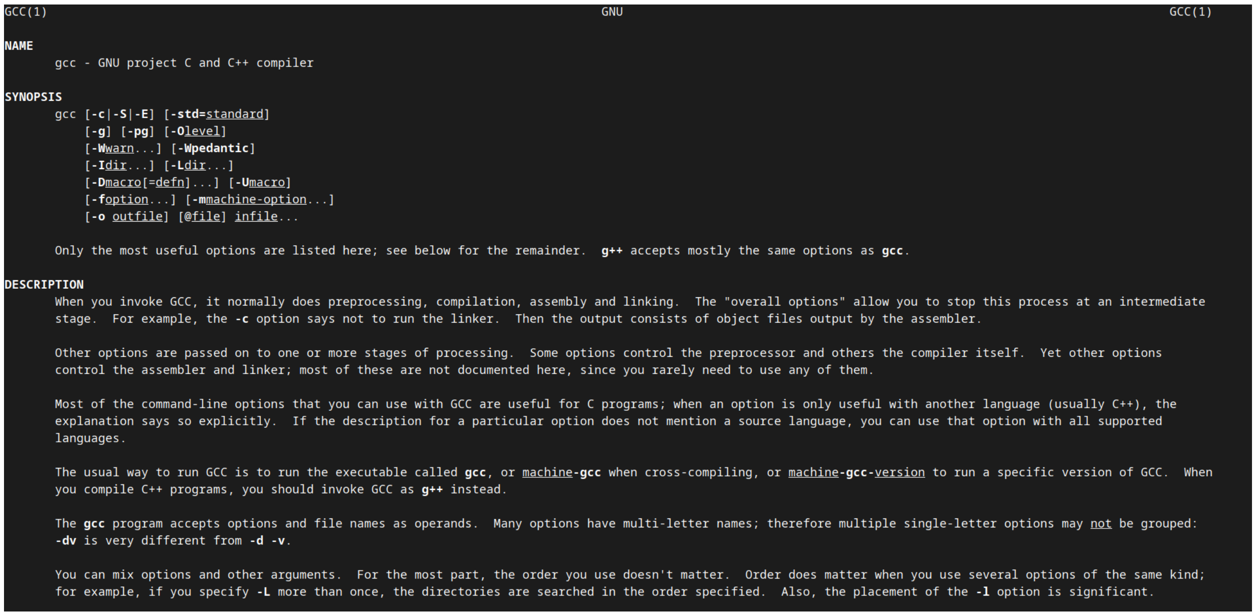
Question 1: cd /home/vpallip/ecpe170/project1/src

Question 2:

Top Level: /

Lowest Level: /home/vpallip/ecpe170/project2/src/

Question 3:



```
GCC(1)                                          GNU                                          GCC(1)

NAME
       gcc - GNU project C and C++ compiler

SYNOPSIS
       gcc [-c|-S|-E] [-std=standard]
           [-g] [-pg] [-Olevel]
           [-Wwarn...] [-Wpedantic]
           [-Idir...] [-Ldir...]
           [-Dmacro[=defn]...] [-Umacro]
           [-foption...] [-mmachine-option...]
           [-o outfile] [@file] infile...

       Only the most useful options are listed here; see below for the remainder.  g++ accepts mostly the same options as gcc.

DESCRIPTION
       When you invoke GCC, it normally does preprocessing, compilation, assembly and linking.  The "overall options" allow you to stop this process at an intermediate
       stage.  For example, the -c option says not to run the linker.  Then the output consists of object files output by the assembler.

       Other options are passed on to one or more stages of processing.  Some options control the preprocessor and others the compiler itself.  Yet other options
       control the assembler and linker; most of these are not documented here, since you rarely need to use any of them.

       Most of the command-line options that you can use with GCC are useful for C programs; when an option is only useful with another language (usually C++), the
       explanation says so explicitly.  If the description for a particular option does not mention a source language, you can use that option with all supported
       languages.

       The usual way to run GCC is to run the executable called gcc, or machine-gcc when cross-compiling, or machine-gcc-version to run a specific version of GCC.  When
       you compile C++ programs, you should invoke GCC as g++ instead.

       The gcc program accepts options and file names as operands.  Many options have multi-letter names; therefore multiple single-letter options may not be grouped:
       -dv is very different from -d -v.

       You can mix options and other arguments.  For the most part, the order you use doesn't matter.  Order does matter when you use several options of the same kind;
       for example, if you specify -L more than once, the directories are searched in the order specified.  Also, the placement of the -l option is significant.
```

Question 4a: Recursively deletes the root directory

Question 5:

General command: man ls

Next Page: PgDn or down button

Quit: q

Question 6:

wc -m < myfile.txt > myfile_char_count.txt

Question 7

ls -a ~

Question 8

mv data.txt ~/experiment1

Question 9

ls -S /etc

Question 10

wget http://www.google.com/doodles/roswells-66th-anniversary

Question 11

pwd

Question 12

ls -lh /boot

Question 13

```
df -h /
```

Question 14

The kernel is the heart of an operating system. It enables sharing hardware resources between different processes. It also allows access to the cpu, memory, disk, i/o, and networking. In short, it allows physical access to the devices plugged into the system.

Question 15

Ubuntu Linux is the kernel combined with a lot more. For example, a GUI is not part of the kernel, but Ubuntu does use the kernel to interact with your keyboard and mouse. In short, Ubuntu contains the kernel and builds on top of it to provide a usable experience to the user.

Question 16

A virtual machine is a machine that is running inside a container on an existing system. It is separated from the host and most of the actual system is emulated. This means if you brick the system in the VM, the host system will not be lost

Question 17

Dual booting is different from VM's because dual booting means you're running on actual hardware. This means that the performance of the system will be a lot better, but the risks also are more prevalent. For example, if you brick your boot drive on a dual booting environment, the system will not be able to boot at all and repairs must be made. On the other hand, if the same happens on a VM, the host system will not be affected.

Question 18

VI IS THE BEST CODE EDITOR

Question 19

1. Faster than using a UI most of the time for filesystem/syscalls
2. Allows you to do more abstract stuff with the folders and files
    a. wildcards are a thing
    b. reading and writing without using a whole text editor
3. Ability to string together multiple commands through piping and redirection

Question 20

./ = current folder

../ = current folder's parent