Q1

`mkdir` is for making a new directory. The passed argument is `~/11712005`, which means the path is in the folder of the current user, e.g. `/home/current_username/1172005/` or `/root/11712005/` (if the current user is `root`). In my case, it is `/home/ubuntu/11712005/`.

```
(base) ubuntu@VM-0-15-ubuntu:~$ mkdir ~/11712005
(base) ubuntu@VM-0-15-ubuntu:~$ ls
11712005  frps  libtin  miniconda3  note.txt  random.ipynb  RustCode  simple_web_server  SQLBackup  temp
```

Q2

`ls` is for listing directory contents. The option `-la` is short for `-l -a`, of which meanings are using a long list format and all(not ignoring entries starting with `.`), respectively. The passed argument is `~`, which is the path of `/home/current_user/` or or `/root/` (if the current user is `root`). In my case, it is `/home/ubuntu/`.

```
(base) ubuntu@VM-0-15-ubuntu:~$ ls -la ~
total 164
drwxr-xr-x 23 ubuntu ubuntu  4096 Feb 21 15:54 .
drwxr-xr-x  4 root   root    4096 Apr  9  2019 ..
drwxrwxr-x  2 ubuntu ubuntu  4096 Feb 21 15:54 11712005
-rw-------  1 ubuntu ubuntu 12773 Feb 21 15:54 .bash_history
-rw-r--r--  1 ubuntu ubuntu   220 Aug  8  2018 .bash_logout
-rw-r--r--  1 ubuntu ubuntu  4288 Dec 19 00:13 .bashrc
drwx------  5 ubuntu ubuntu  4096 Jan  3 19:53 .cache
drwxrwxr-x  3 ubuntu ubuntu  4096 Aug 10  2019 .cargo
drwxrwxr-x  3 ubuntu ubuntu  4096 Dec 18 11:47 .cmake
drwxrwxr-x  2 ubuntu ubuntu  4096 Dec 19 00:13 .conda
-rw-rw-r--  1 ubuntu ubuntu   376 Dec 19 00:28 .condarc
drwxr-xr-x  3 ubuntu ubuntu  4096 Jan 14 16:19 frps
drwx------  3 ubuntu ubuntu  4096 Aug  9  2018 .gnupg
drwxrwxr-x  2 ubuntu ubuntu  4096 Jan  3 19:47 .ipynb_checkpoints
drwxr-xr-x  5 ubuntu ubuntu  4096 Dec 19 00:42 .ipython
drwx------  3 ubuntu ubuntu  4096 Feb  9 21:33 .jupyter
drwxrwxr-x  3 ubuntu ubuntu  4096 Dec 18 11:42 libtin
drwx------  4 ubuntu ubuntu  4096 Dec 19 00:37 .local
drwxrwxr-x 25 ubuntu ubuntu  4096 Feb  9 21:27 miniconda3
-rw-------  1 ubuntu ubuntu   142 Aug 10  2019 .mysql_history
-rw-rw-r--  1 ubuntu ubuntu   166 Aug 11  2019 note.txt
drwxr-xr-x  2 root   root    4096 Apr  9  2019 .pip
-rw-r--r--  1 ubuntu ubuntu   845 Aug 10  2019 .profile
-rw-r--r--  1 root   root      73 Apr  9  2019 .pydistutils.cfg
-rw-rw-r--  1 ubuntu ubuntu  1925 Feb  9 21:50 random.ipynb
drwxrwxr-x  3 ubuntu ubuntu  4096 Aug 10  2019 RustCode
drwxrwxr-x  6 ubuntu ubuntu  4096 Aug 10  2019 .rustup
drwxrwxr-x  6 ubuntu ubuntu  4096 Aug 10  2019 simple_web_server
drwxrwxr-x  2 ubuntu ubuntu  4096 Apr 10  2019 SQLBackup
drwx------  2 ubuntu ubuntu  4096 Sep 11  2018 .ssh
-rw-r--r--  1 ubuntu ubuntu     0 Aug  9  2018 .sudo_as_admin_successful
drwxr-xr-x  3 ubuntu ubuntu  4096 Feb 21 15:42 temp
-rw-------  1 root   root   11722 Aug 11  2019 .viminfo
drwxrwxr-x  5 ubuntu ubuntu  4096 Dec 18 11:10 .vscode-server
-rw-rw-r--  1 ubuntu ubuntu   183 Dec 18 11:10 .wget-hsts
-rw-------  1 ubuntu ubuntu    62 Mar 18  2019 .Xauthority
```

Q3

```
(base) ubuntu@VM-0-15-ubuntu:~$ cd ~/11712005
(base) ubuntu@VM-0-15-ubuntu:~/11712005$
```

`cd` is for changing directories. The passed argument is `~/11712005`, which is the path of the folder of the current user, e.g. `/home/current_username/1172005/` or `/root/11712005/` (if the current user is `root`), which, in my case, is `/home/ubuntu/11712005/`.

Q4

`man` is "an interface to the on-line reference manuals", which can look up a command's manual. The passed argument is `grep`, which is the name of the command.

```
GREP(1)                                                    User Commands                                                    GREP(1)

NAME
       grep, egrep, fgrep, rgrep - print lines matching a pattern

SYNOPSIS
       grep [OPTIONS] PATTERN [FILE...]
       grep [OPTIONS] -e PATTERN ... [FILE...]
       grep [OPTIONS] -f FILE ... [FILE...]

DESCRIPTION
       grep  searches  for  PATTERN in each FILE.  A FILE of "-" stands for standard input.  If no FILE is given, recursive searches examine the working directory,
       and nonrecursive searches read standard input.  By default, grep prints the matching lines.

       In addition, the variant programs egrep, fgrep and rgrep are the same as grep -E, grep -F, and grep -r, respectively.  These variants  are  deprecated,  but
       are provided for backward compatibility.

OPTIONS
   Generic Program Information
       --help Output a usage message and exit.

       -V, --version
              Output the version number of grep and exit.

   Matcher Selection
       -E, --extended-regexp
              Interpret PATTERN as an extended regular expression (ERE, see below).

       -F, --fixed-strings
              Interpret PATTERN as a list of fixed strings (instead of regular expressions), separated by newlines, any of which is to be matched.

       -G, --basic-regexp
              Interpret PATTERN as a basic regular expression (BRE, see below).  This is the default.

       -P, --perl-regexp
              Interpret the pattern as a Perl-compatible regular expression (PCRE).  This is experimental and grep -P may warn of unimplemented features.

   Matching Control
       -e PATTERN, --regexp=PATTERN
              Use  PATTERN  as the pattern.  If this option is used multiple times or is combined with the -f (--file) option, search for all patterns given.  This
              option can be used to protect a pattern beginning with "-".

       -f FILE, --file=FILE
              Obtain patterns from FILE, one per line.  If this option is used multiple times or is combined with the -e (--regexp) option, search for all patterns
              given.  The empty file contains zero patterns, and therefore matches nothing.

       -i, --ignore-case
              Ignore case distinctions, so that characters that differ only in case match each other.
```

Q5

`mv` is for moving(renaming) files. The passed arguments are `~/11712005` and `/home`, the first of which is the folder of `/home/current_user/11712005` (in my case, it is `/home/ubuntu/11712005`). The command is to move the directory of `/home/ubuntu/11712005` into `/home/`.

```
(base) ubuntu@VM-0-15-ubuntu:~/11712005$ mv ~/11712005 /home
mv: cannot move '/home/ubuntu/11712005' to '/home/11712005': Permission denied
(base) ubuntu@VM-0-15-ubuntu:~/11712005$ sudo mv ~/11712005 /home
```

```
(base) ubuntu@VM-0-15-ubuntu:~/11712005$ ls /home
11712005  maplesftp  ubuntu
```

Q6

`rm` is for removing files or directories. The option `-r` means removing recursively, which is needed when deleting all of the contents in a directory. The passed argument is the path of the directory `/home/11712005`.

```
(base) ubuntu@VM-0-15-ubuntu:~/11712005$ rm -r /home/11712005
rm: cannot remove '/home/11712005': Permission denied
(base) ubuntu@VM-0-15-ubuntu:~/11712005$ sudo rm -r /home/11712005
```

Q7

`cp` is for copying files and directories. The first argument is `/etc/apt/sources.list`, which is the source, and the second one is `/etc/apt/sources.list.bak`, which is the destination.

```
(base) ubuntu@VM-0-15-ubuntu:~$ cp /etc/apt/sources.list /etc/apt/sources.list.bak
cp: cannot create regular file '/etc/apt/sources.list.bak': Permission denied
(base) ubuntu@VM-0-15-ubuntu:~$ sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
```

Q8

`cat` is for concatenating files and print on the standard output. The argument is the path of a file `/etc/shells`. This command here actually just print out the content of the file.

```
(base) ubuntu@VM-0-15-ubuntu:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/bin/rbash
/bin/dash
/usr/bin/tmux
/usr/bin/screen
```

Q9

`cat` is for concatenating files and print on the standard output. The argument of `cat` is the path of a file `/etc/shells`. `grep` is for printing lines matching a pattern, which here is `bash`, meaning containing the string of “bash”. `|` is for tube or tunnel. Here it means taking the standard output of `cat` as the standard input of `grep`. This command in fact goes through the content of `/etc/shells`, searching for lines containing “bash” as substrings.

```
(base) ubuntu@VM-0-15-ubuntu:/etc$ cat /etc/shells | grep bash
/bin/bash
/bin/rbash
```

Q10

To find out the `pid`s of two terminals, we can find their associated shells’ `pid`s. The default shell I use is `bash`, so I type `ps –ef|grep bash`. As I am connecting a remote machine via `ssh`, to kill one of the terminals, I can kill its process or its parent process(which is the process of `sshd`). To kill its process, type `kill 29688` and the corresponding terminal is closed.

```
(base) ubuntu@VM-0-15-ubuntu:~$ ps -ef|grep bash
ubuntu   29688 29687  0 17:22 pts/0    00:00:00 -bash
ubuntu   29800 29797  0 17:22 pts/3    00:00:00 -bash
ubuntu   29848 29688  0 17:23 pts/0    00:00:00 grep --color=auto bash
```

Q11

Source Code:

```c
#include <stdio.h>
int main()
{
    int x = 0;
    x+=1;
    x+=1;
    x+=1;
    printf("%d\n",x);
    return 0;
}
```

Commands:

```
gcc -S -O0 opt.c
gcc -S -O1 -o opt.s1 opt.c
```

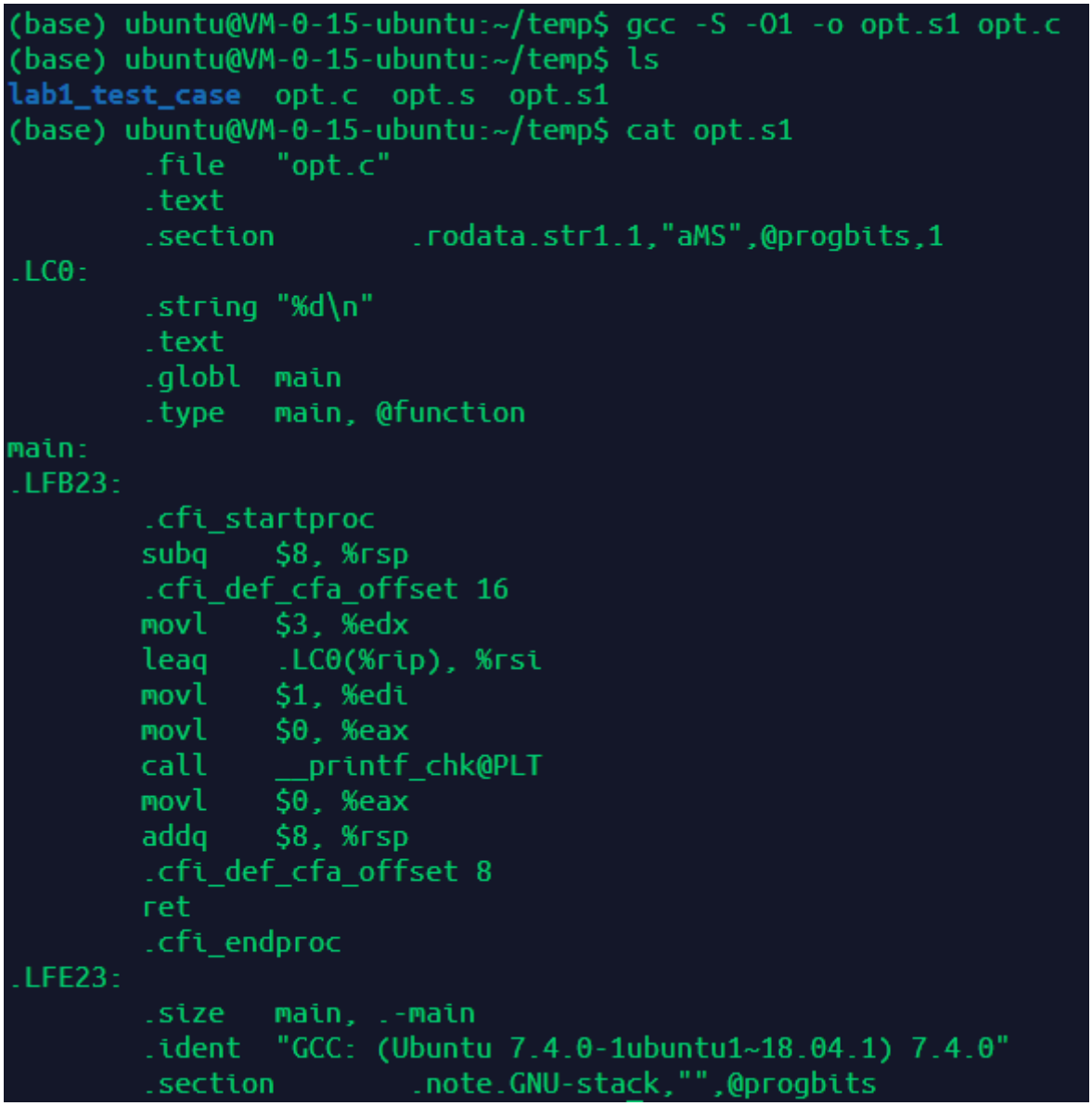
Results:

```
# in opt.s using O0
        .file   "opt.c"
        .text
        .section        .rodata
.LC0:
        .string "%d\n"
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movl    $0, -4(%rbp)
        addl    $1, -4(%rbp)
        addl    $1, -4(%rbp)
        addl    $1, -4(%rbp)
        movl    -4(%rbp), %eax
        movl    %eax, %esi
        leaq    .LC0(%rip), %rdi
        movl    $0, %eax
        call    printf@PLT
        movl    $0, %eax
        leave
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0"
        .section        .note.GNU-stack,"",@progbits

# in opt.s1 using O1
        .file   "opt.c"
        .text
        .section        .rodata.str1.1,"aMS",@progbits,1
.LC0:
        .string "%d\n"
        .text
        .globl  main
        .type   main, @function
main:
.LFB23:
        .cfi_startproc
        subq    $8, %rsp
        .cfi_def_cfa_offset 16
```

```
        movl    $3, %edx
        leaq    .LC0(%rip), %rsi
        movl    $1, %edi
        movl    $0, %eax
        call    __printf_chk@PLT
        movl    $0, %eax
        addq    $8, %rsp
        .cfi_def_cfa_offset 8
        ret
        .cfi_endproc
.LFE23:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0"
        .section        .note.GNU-stack,"",@progbits
```

```
(base) ubuntu@VM-0-15-ubuntu:~/temp$ gcc -S -O0 opt.c
(base) ubuntu@VM-0-15-ubuntu:~/temp$ ls
lab1_test_case  opt.c  opt.s
(base) ubuntu@VM-0-15-ubuntu:~/temp$ cat opt.s
        .file   "opt.c"
        .text
        .section        .rodata
.LC0:
        .string "%d\n"
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        subq    $16, %rsp
        movl    $0, -4(%rbp)
        addl    $1, -4(%rbp)
        addl    $1, -4(%rbp)
        addl    $1, -4(%rbp)
        movl    -4(%rbp), %eax
        movl    %eax, %esi
        leaq    .LC0(%rip), %rdi
        movl    $0, %eax
        call    printf@PLT
        movl    $0, %eax
        leave
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0"
        .section        .note.GNU-stack,"",@progbits
```



```
(base) ubuntu@VM-0-15-ubuntu:~/temp$ gcc -S -O1 -o opt.s1 opt.c
(base) ubuntu@VM-0-15-ubuntu:~/temp$ ls
lab1_test_case  opt.c  opt.s  opt.s1
(base) ubuntu@VM-0-15-ubuntu:~/temp$ cat opt.s1
        .file   "opt.c"
        .text
        .section        .rodata.str1.1,"aMS",@progbits,1
.LC0:
        .string "%d\n"
        .text
        .globl  main
        .type   main, @function
main:
.LFB23:
        .cfi_startproc
        subq    $8, %rsp
        .cfi_def_cfa_offset 16
        movl    $3, %edx
        leaq    .LC0(%rip), %rsi
        movl    $1, %edi
        movl    $0, %eax
        call    __printf_chk@PLT
        movl    $0, %eax
        addq    $8, %rsp
        .cfi_def_cfa_offset 8
        ret
        .cfi_endproc
.LFE23:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0"
        .section        .note.GNU-stack,"",@progbits
```

Analysis:

As can be seen in these two screenshots, the numbers of assembly codes are different. The number of instructions of `opt.s1` is less than that of `opt.s`, which means that from the perspective of the number of assembly instructions, O1 optimization will give better performance than O0 optimization. If we take a closer look at the differences, it is noticeable that `addl` instructions are removed from the 01–optimized version of code, which means O1 optimization will examine unnecessary calculations and replace them with the result of these calculations.