**OS Lab**

**Session 1 – Lab 2 (08/12/2020)**

Parthivi Choubey CSE – B - 5th semester

180905456 Roll. no. - 60

**Question 1**

**Code**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <dirent.h>

#include <sys/stat.h>

#include <string.h>

#include <time.h>

int main()

{

char pwd[256];

if(!getcwd(pwd, sizeof(pwd)))

printf("Directory name exceeds size of buffer or is removed or permissions are changed.");

DIR \*dp;

struct dirent \*entry;

struct stat fileStat;

if((dp = opendir(pwd))==NULL)

printf("Can not open directory.");

else

{

while((entry = readdir(dp))!=NULL)

{

if(strcmp(".", entry->d\_name)==0 || strcmp("..", entry->d\_name)==0 || entry->d\_name[0]=='.') continue;

else

{

printf("%s\t", entry->d\_name);

lstat(entry->d\_name, &fileStat);

printf((S\_ISDIR(fileStat.st\_mode)) ? "d" : "-");

printf((fileStat.st\_mode & S\_IRUSR) ? "r" : "-");

printf((fileStat.st\_mode & S\_IWUSR) ? "w" : "-");

printf((fileStat.st\_mode & S\_IXUSR) ? "x" : "-");

printf((fileStat.st\_mode & S\_IRGRP) ? "r" : "-");

printf((fileStat.st\_mode & S\_IWGRP) ? "w" : "-");

printf((fileStat.st\_mode & S\_IXGRP) ? "x" : "-");

printf((fileStat.st\_mode & S\_IROTH) ? "r" : "-");

printf((fileStat.st\_mode & S\_IWOTH) ? "w" : "-");

printf((fileStat.st\_mode & S\_IXOTH) ? "x" : "-");

printf("\t%d",fileStat.st\_size);

printf("\t%s",ctime(&fileStat.st\_ctime));

}

}

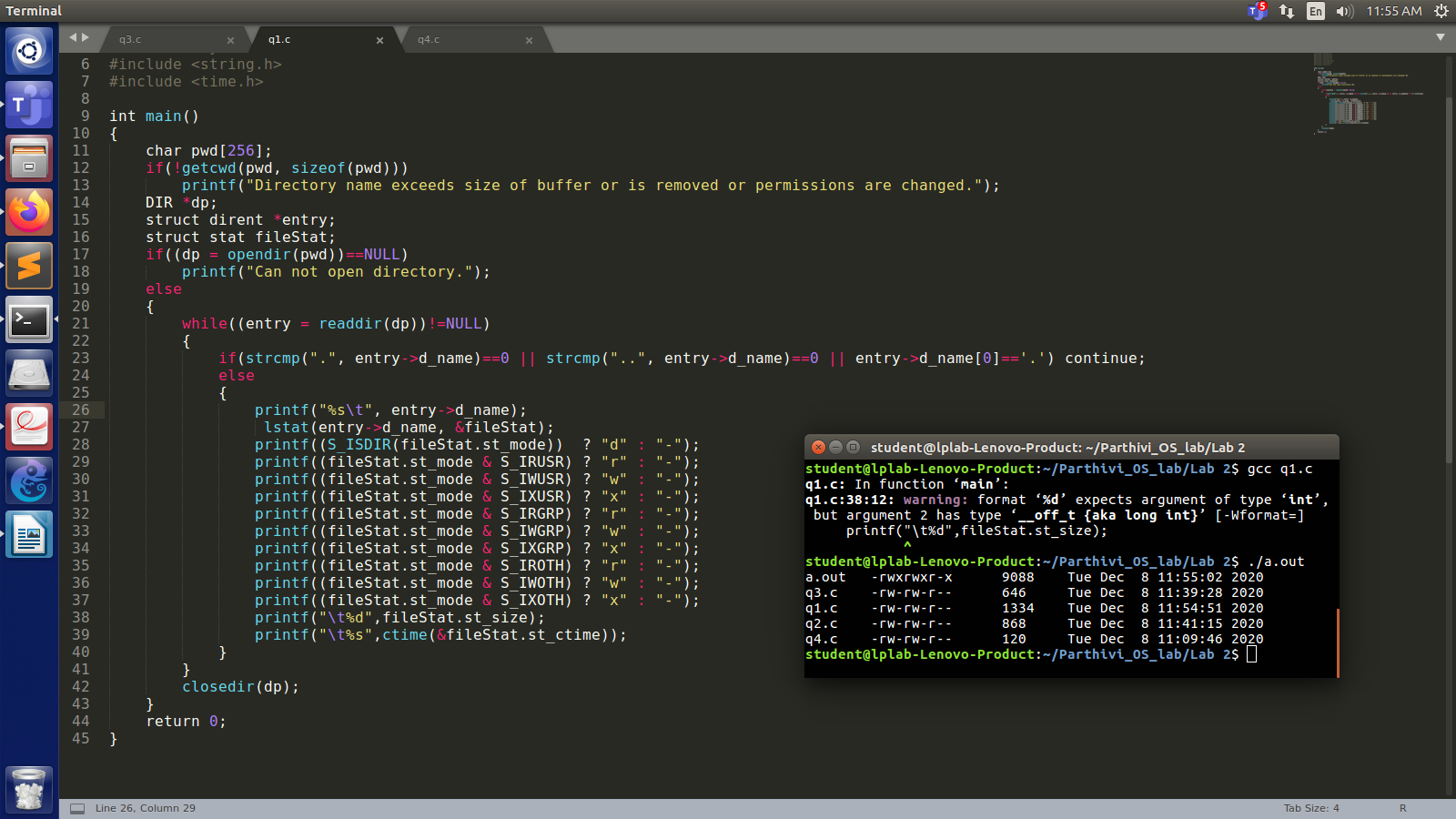
closedir(dp);

}

return 0;

}

**Output**



**Question 2**

**Code**

#include <unistd.h>

#include <stdio.h>

#include <dirent.h>

#include <string.h>

#include <sys/stat.h>

#include <stdlib.h>

void printdir(char \*dir, int depth) {

DIR \*dp;

struct dirent \*entry;

struct stat statbuf;

if((dp = opendir(dir)) == NULL) {

fprintf(stderr, "Cannot open directory: %s\n", dir);

return;

}

chdir(dir);

while((entry = readdir(dp)) != NULL)

{

lstat(entry->d\_name, &statbuf);

if(S\_ISDIR(statbuf.st\_mode))

{

if(strcmp(".", entry->d\_name) == 0 || strcmp("..", entry->d\_name) == 0) continue;

printf("%\*s%s/\n", depth,"",entry->d\_name);

printdir(entry->d\_name, depth+4);

}

else

{

printf("%\*s%s\n", depth,"", entry->d\_name);

}

}

chdir("..");

closedir(dp);

}

int main(int argc, char\* argv[])

{

if(argc != 2)

{

printf("Invalid arguments. Only enter required directory.");

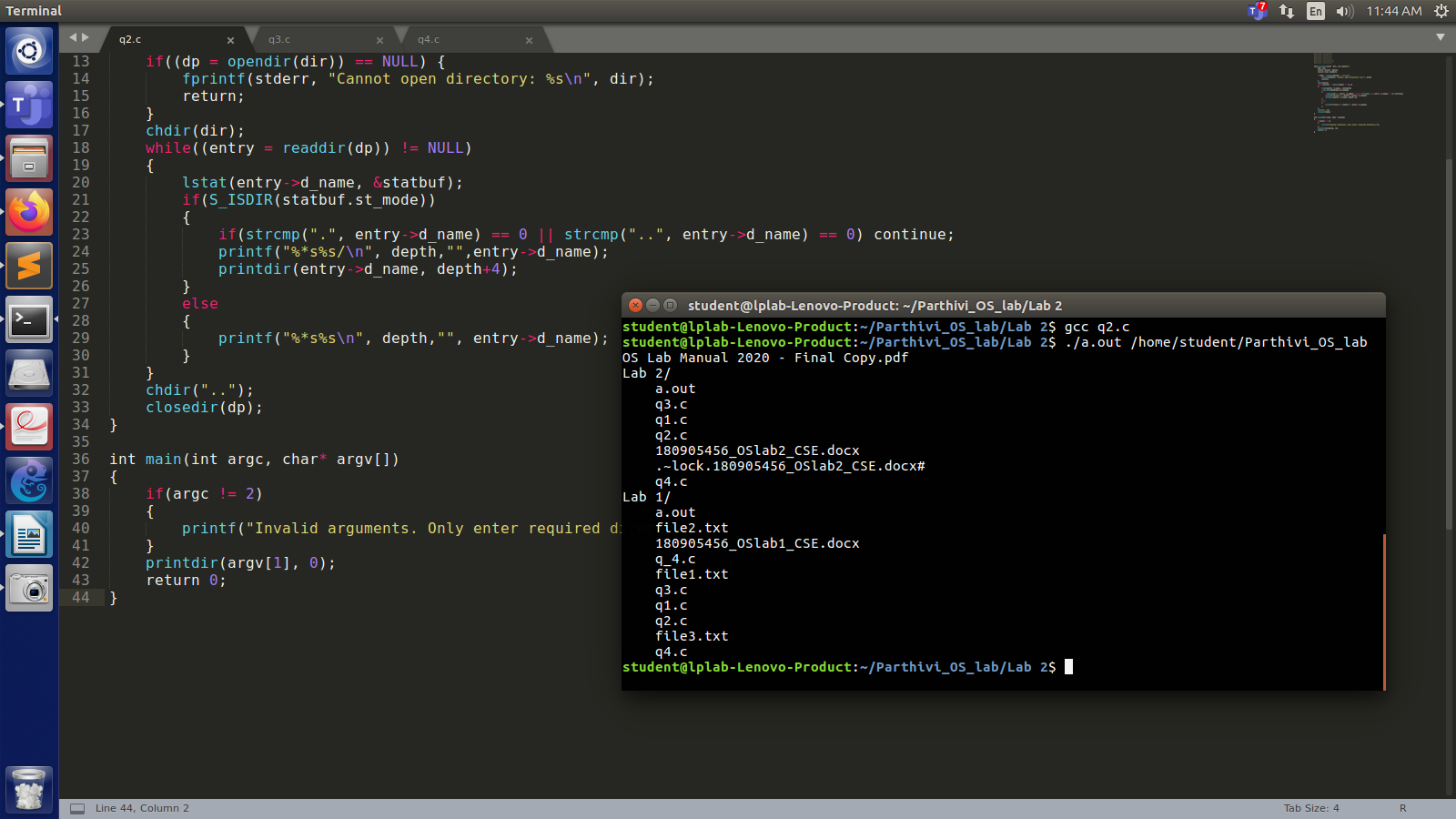
}

printdir(argv[1], 0);

return 0;

}

**Output**

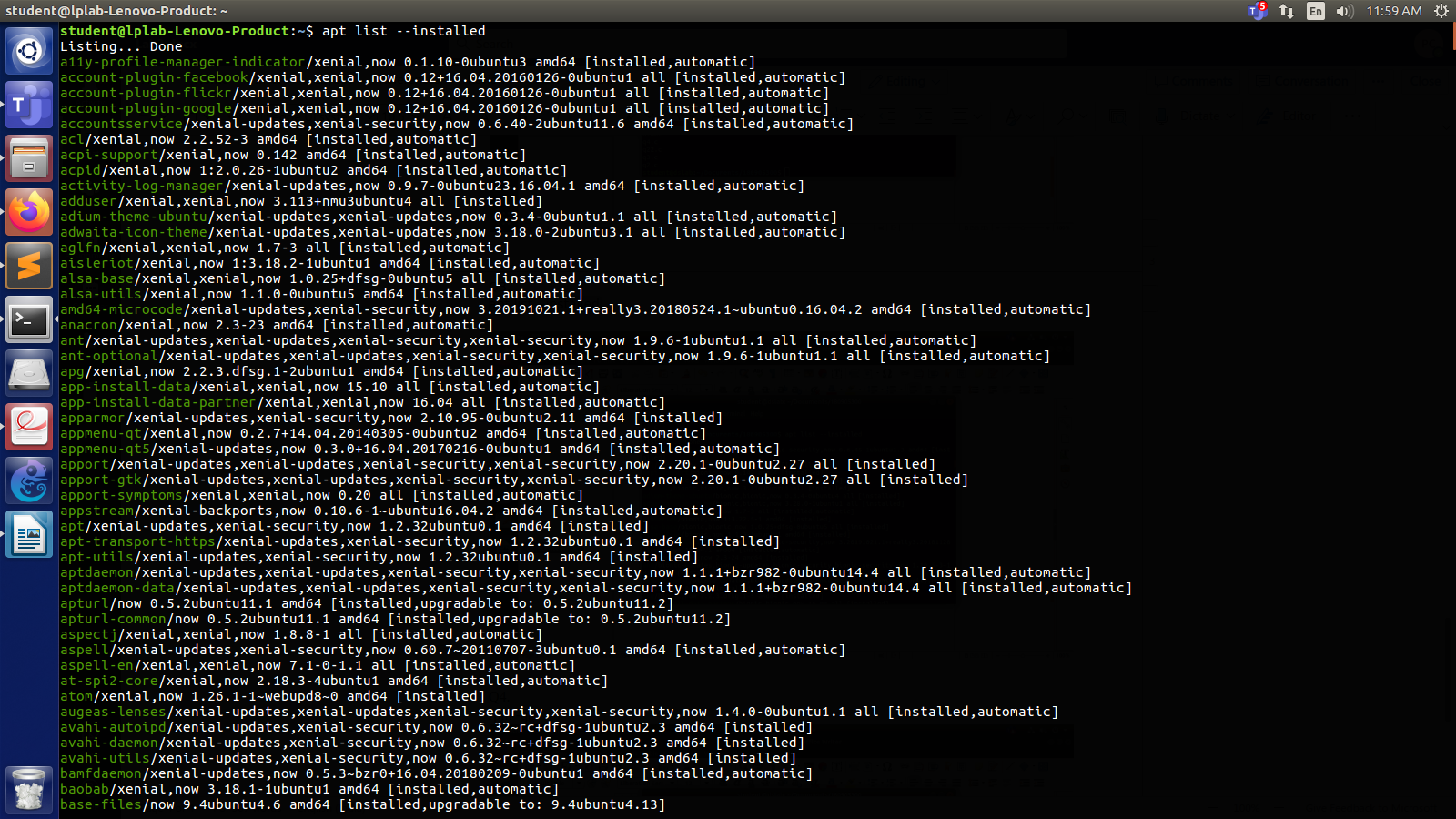
****

**Question 3 (Way 1)**

**Code**

$ apt list --installed

**Output**

****

**Question 3 (Way 2)**

**Code**

#include <string.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <dirent.h>

#include <time.h>

#include <pwd.h>

#include <grp.h>

int main(int argc, char\* argv[])

{

DIR \*dp = opendir(argv[1]);

struct dirent \*entry;

while((entry = readdir(dp)) != NULL)

{

if(strcmp(".", entry->d\_name) == 0 || strcmp("..", entry->d\_name) == 0) continue;

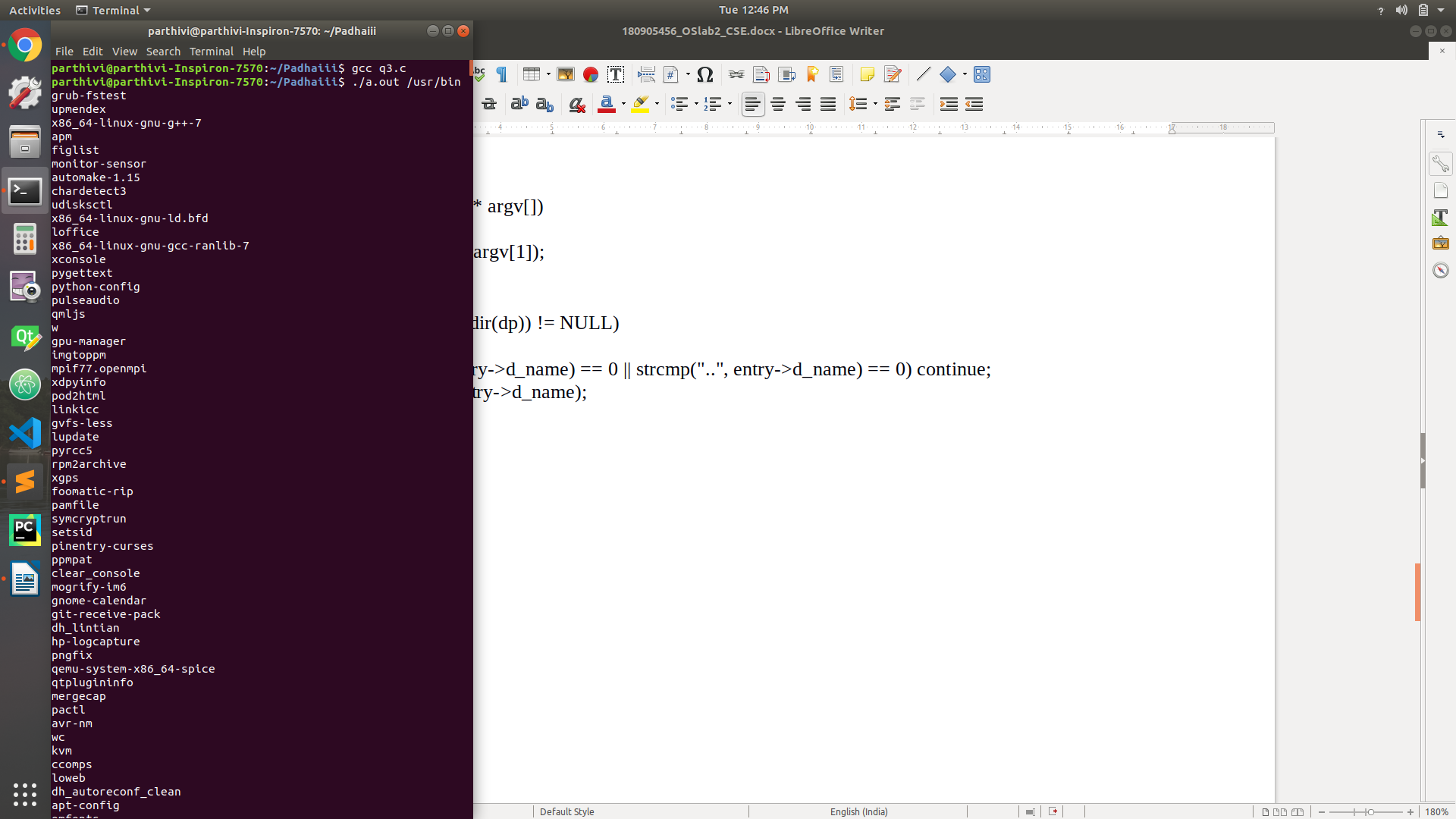
printf("%s\n", entry->d\_name);

}

closedir(dp);

}

**Output**

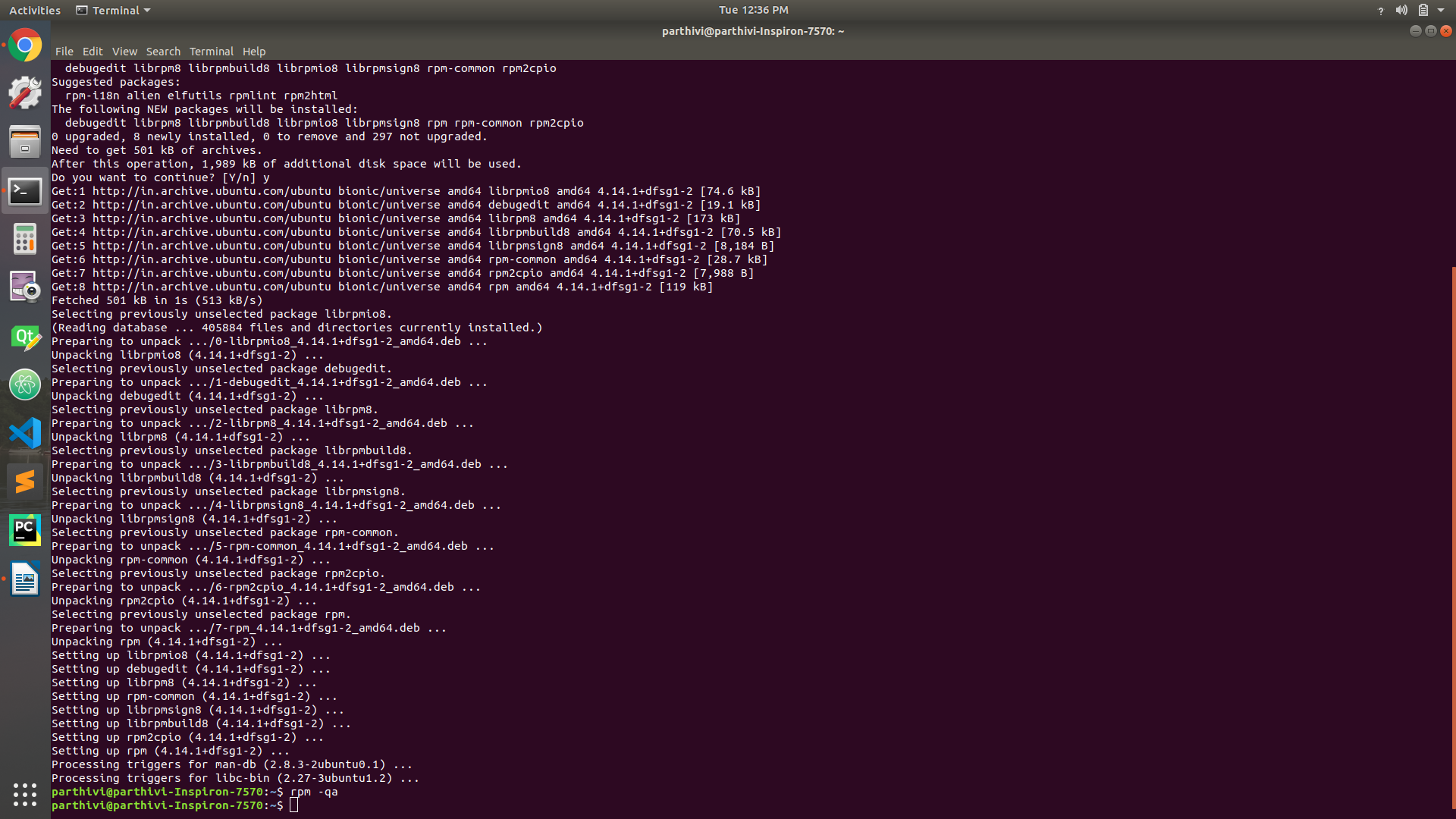
****

**Question 4 (Way 1)**

**Code**

$ rpm -qa

**Output**



**Question 4 (Way 2)**

**Code**

#include <unistd.h>

#include <stdio.h>

#include <dirent.h>

#include <string.h>

#include <sys/stat.h>

#include <stdlib.h>

void printDir(char \*dir,int depth)

{

DIR \* dp;

struct dirent \* entry;

struct stat statbuf;

if((dp = opendir(dir)) == NULL) {

printf("Cannot open directory \n");

return;

}

chdir(dir);

while((entry = readdir(dp)) != NULL) {

lstat(entry->d\_name,&statbuf);

if(S\_ISDIR(statbuf.st\_mode)){

if(strcmp(".",entry->d\_name) == 0 || strcmp(".." , entry->d\_name ) == 0 )

continue;

printf("%\*s%s/\n",depth,"",entry->d\_name );

printDir(entry->d\_name,depth+4);

}

else{

printf("%\*s%s/\n",depth,"",entry->d\_name);

}

}

chdir(".");

closedir(dp);

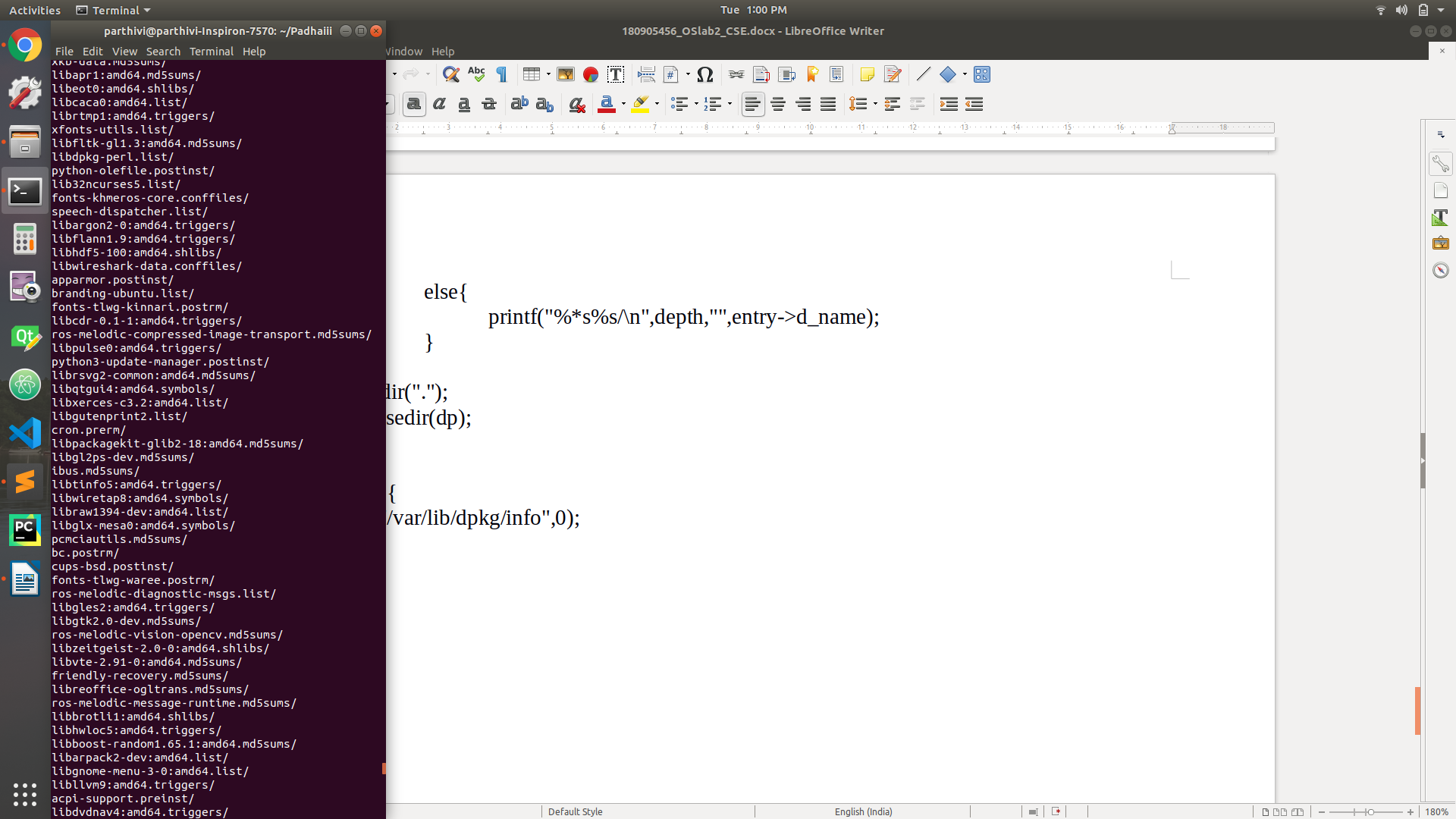
}

int main() {

printDir("/var/lib/dpkg/info",0);

}

**Output**

****