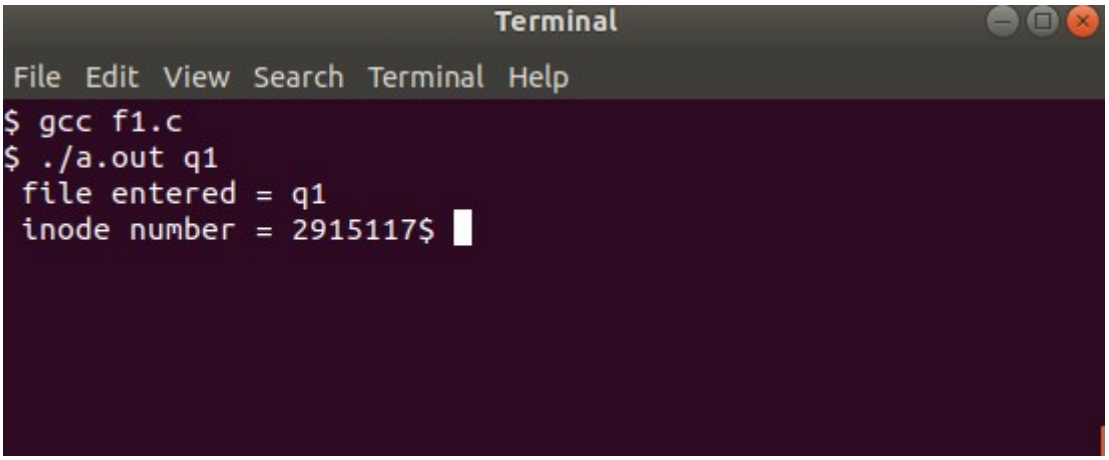
OS Lab 4

Aniruddha Amit Dutta
roll - 58
180905488

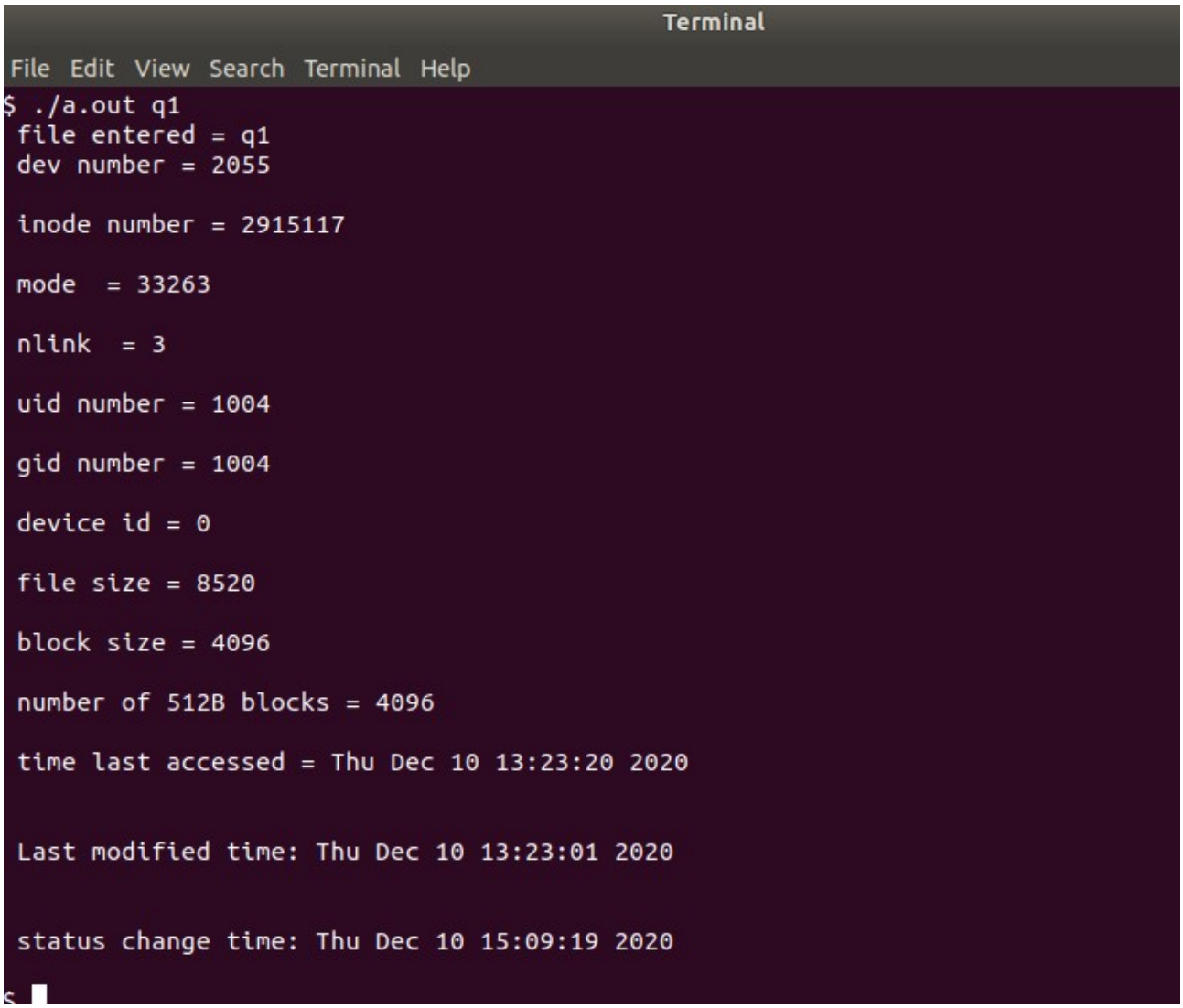
Q1.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc , char*argv[])
{
	struct stat sb;
	int ret;
	if(argc < 2){
		printf(" enter file name \n");
		exit(0);
	}

	printf(" file entered = %s",argv[1]);
	ret = stat(argv[1],&sb);

	if(ret){
		perror("stat");
		exit(0);
	}

	printf("\n inode number = %ld",sb.st_ino);

	return 0;
}
```



Q2.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(int argc , char*argv[])
{
	struct stat sb;
	int ret;
	if(argc < 2){
		printf(" enter file name \n");
		exit(0);
	}

	printf(" file entered = %s",argv[1]);
	ret = stat(argv[1],&sb);

	if(ret){
		perror("stat");
		exit(0);
	}

	printf("\n dev number = %ld\n",sb.st_dev);
	printf("\n inode number = %ld\n",sb.st_ino);
	printf("\n mode  = %d\n",sb.st_mode);
	printf("\n nlink  = %ld\n",sb.st_nlink);
	printf("\n uid number = %d\n",sb.st_uid);
	printf("\n gid number = %d\n",sb.st_gid);
	printf("\n device id = %ld\n",sb.st_rdev);
	printf("\n file size = %ld\n",sb.st_size);
	printf("\n block size = %ld\n",sb.st_blksize);
	printf("\n number of 512B blocks = %ld\n",sb.st_blksize);
	printf("\n time last accessed = %s\n",ctime(&sb.st_atim));
	printf("\n Last modified time: %s\n", ctime(&sb.st_mtim));
	printf("\n status change time: %s\n", ctime(&sb.st_ctim));
	return 0;
}
```

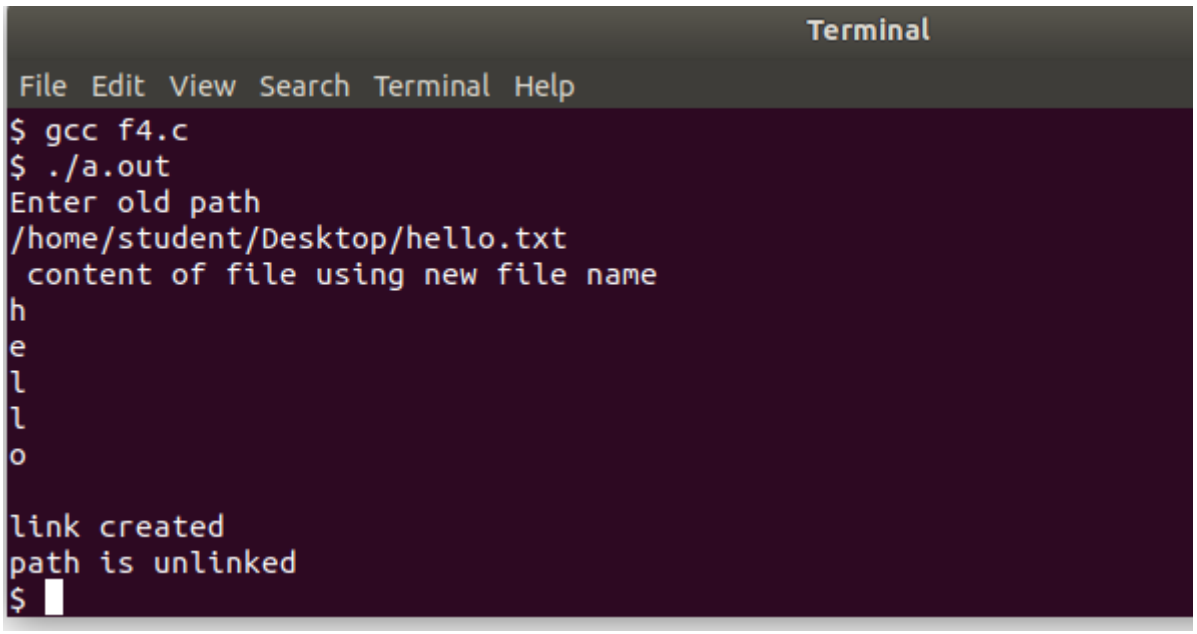


//

Q3

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>

int main(int argc , char*argv[])
{
        struct stat sb;
        int ret,ret2;
        FILE * fptr;
        // if(argc < 2){
        //      printf(" enter file name \n");
```

```
	//		exit(0);
	// }
	char oldpath[100];
	char c;
	printf("Enter old path\n");
	scanf("%s", oldpath);

	// printf(" file entered = %s",argv[1]);
	// ret = link("/home/student/Desktop/hello.txt","/home/student/Desktop/newname4");
	ret = link(oldpath,"/home/student/Desktop/newname4");
	fptr = fopen("/home/student/Desktop/newname4", "r");

	// print file using new filename
	printf(" content of file using new file name\n");
	c = fgetc(fptr);
	while(c!=EOF ){

		printf("%c\n",c );
		c = getc(fptr);
	}

	if(ret !=0){
		perror("link");
		// printf(" %m \n",errno );
		exit(0);
	}else{
		printf("link created \n");
	}

	ret2 = unlink("/home/student/Desktop/newname4");

	if(ret2!=0){
		perror("unlink");
	}
	else{
		printf("path is unlinked \n");
	}

	return 0;
}
```



//

Q4.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>

int main(int argc , char*argv[])
{
	struct stat sb;
	int ret,ret2;
	FILE * fptr;
	// if(argc < 2){
	//	printf(" enter file name \n");
	//	exit(0);
	// }
	char oldpath[100];
	char c;
	printf("Enter old path\n");
	scanf("%s", oldpath);

	// printf(" file entered = %s",argv[1]);
	// ret = link("/home/student/Desktop/hello.txt","/home/student/Desktop/newname4");
	ret = symlink(oldpath,"/home/student/Desktop/newname4");
	fptr = fopen("/home/student/Desktop/newname4", "r");

	// print file using new filename
```

```
        printf(" content of file using new file name\n");
        c = fgetc(fptr);
        while(c!=EOF ){

                printf("%c\n",c );
                c = getc(fptr);
        }

        if(ret !=0){
                perror("symlink");
                // printf(" %m \n",errno );
                exit(0);
        }else{
                printf("soft link created new path : /home/student/Desktop/newname4  \n");
        }

        // ret2 = unlink("/home/student/Desktop/newname4");

        // if(ret2!=0){
        //      perror("unlink");
        // }
        // else{
        //      printf("path is unlinked \n");
        // }

        return 0;
}
```