

MIS :

111403003 – AKASH SARDA

111403023 – ADITYA MALU

111403019 – JASSIM ABDUL REHMAN

LAB #3

1. Using dup function redirect stdin to file1 and stdout to file2. Read a line using scanf and write the same using printf. Verify the contents of both files.

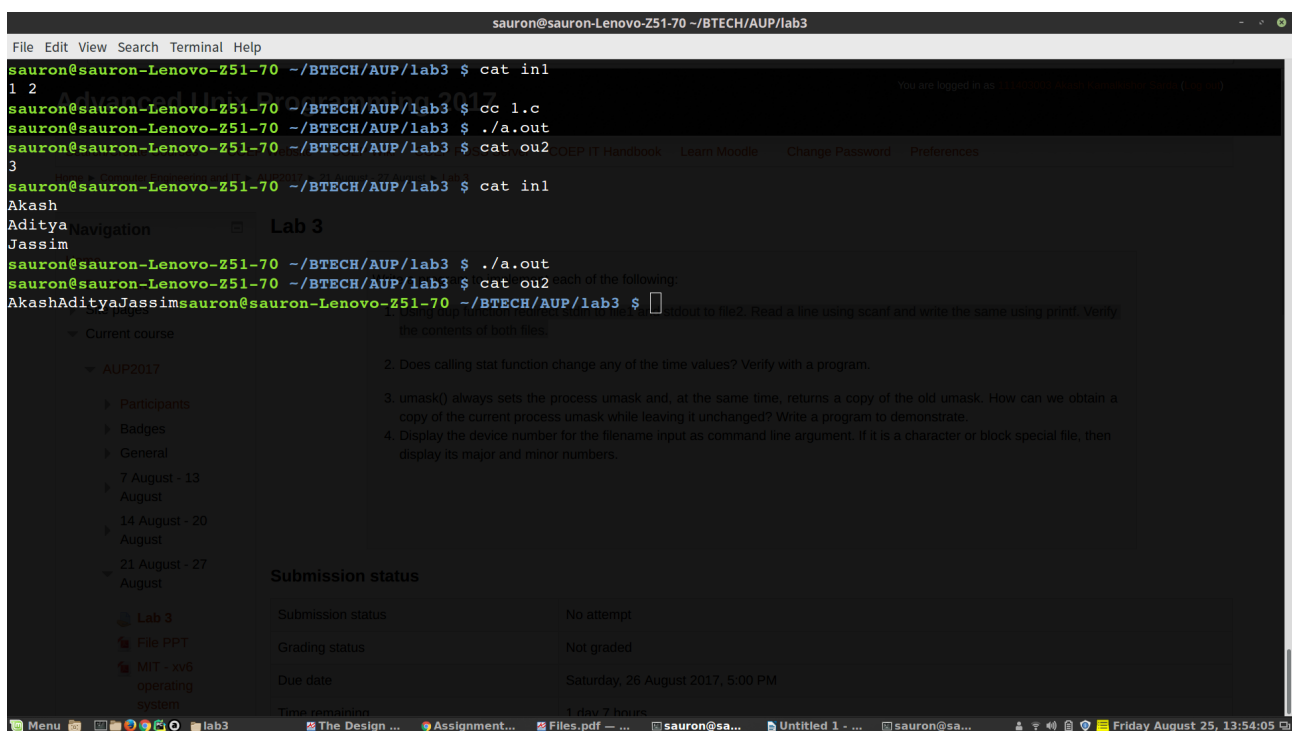
Program:

```
#include<stdio.h>
```

```
#include<unistd.h>
```

```
#include<fcntl.h>
```

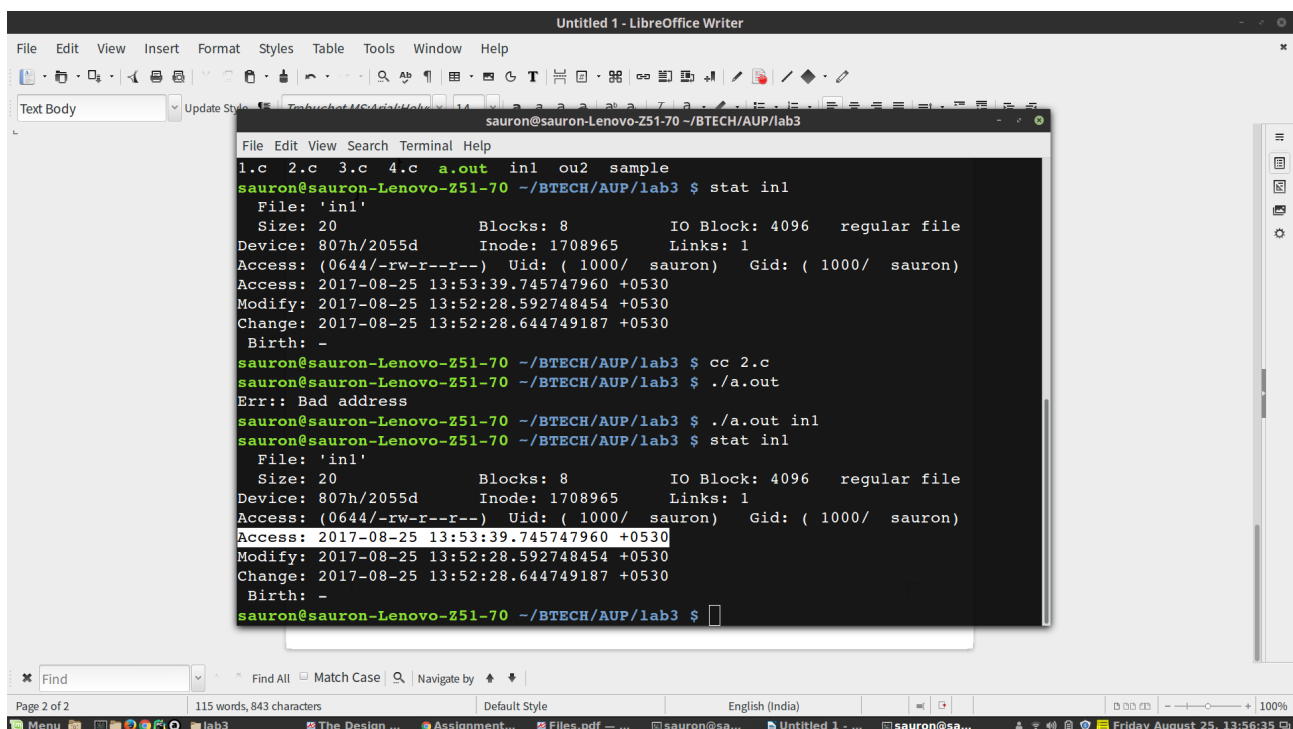
```
int main(int argc, char *argv[]){
    int fd1, fd2, a,b;
    char str[100];
    fd1 = open("./in1", O_RDONLY);
    fd2 = open("./ou2", O_WRONLY);
    dup2(fd2, 1);
    dup2(fd1, 0);
    scanf("%d%d", &a, &b);
    printf("%d", a+b);
    close(fd1);
    close(fd2);
}
```



Ques. 2:

Does calling stat function change any of the time values? Verify with a program.

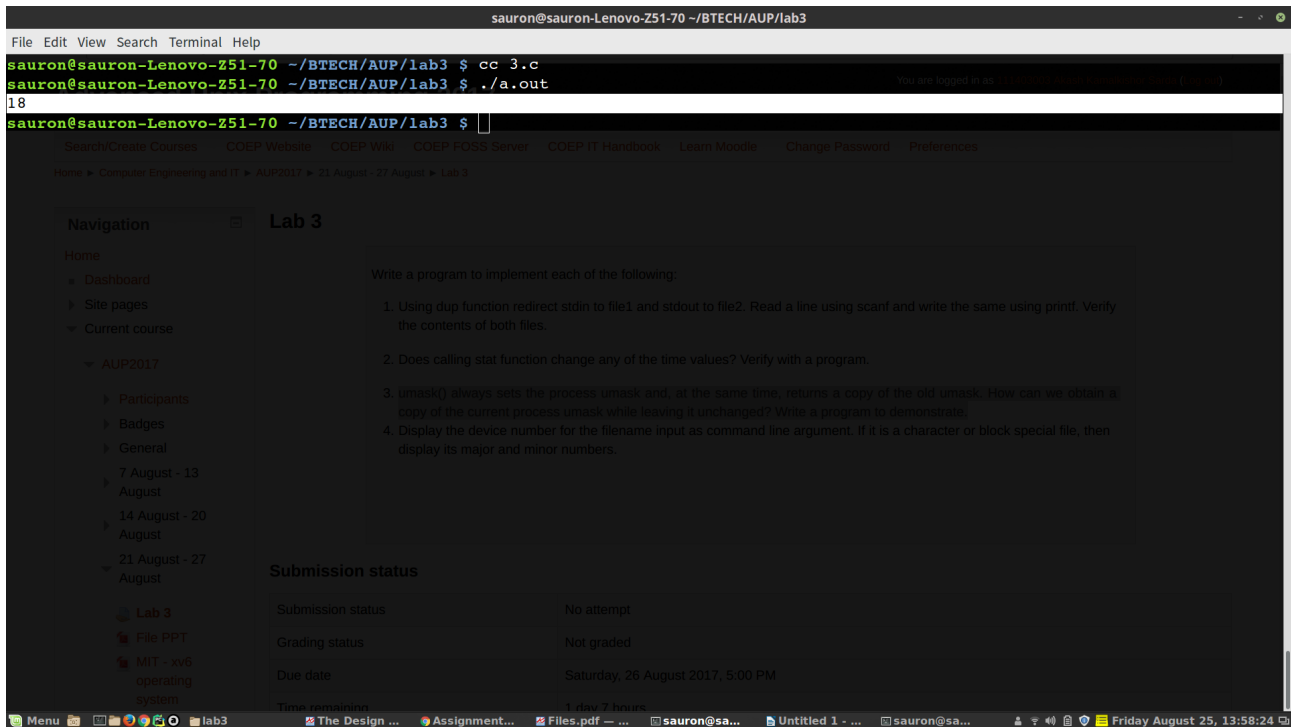
```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<fcntl.h>
#include<sys/stat.h>
#include<sys/types.h>
int main(int argc, char *argv[]){
    struct stat statbuff;
    if(stat(argv[1], &statbuff) == -1){
        perror("Err:");
    }
}
```



```
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab3
1.c 2.c 3.c 4.c a.out in1 ou2 sample
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab3 $ stat in1
  File: 'in1'
  Size: 20          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d Inode: 1708965       Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sauron)   Gid: ( 1000/   sauron)
Access: 2017-08-25 13:53:39.745747960 +0530
Modify: 2017-08-25 13:52:28.592748454 +0530
Change: 2017-08-25 13:52:28.644749187 +0530
 Birth: -
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab3 $ cc 2.c
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab3 $ ./a.out
Err: Bad address
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab3 $ ./a.out in1
  File: 'in1'
  Size: 20          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d Inode: 1708965       Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/   sauron)   Gid: ( 1000/   sauron)
Access: 2017-08-25 13:53:39.745747960 +0530
Modify: 2017-08-25 13:52:28.592748454 +0530
Change: 2017-08-25 13:52:28.644749187 +0530
 Birth: -
sauron@sauron-Lenovo-Z51-70 ~/BTECH/AUP/lab3 $
```

Ques 3:

umask() always sets the process umask and, at the same time, returns a copy of the old umask. How can we obtain a copy of the current process umask while leaving it unchanged? Write a program to demonstrate.



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

```
int main(int argc, char *argv[]){
    mode_t old;
    old = umask(0);
    printf("%d\n", old);
    umask(old);
}
```

Ques 4:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<fcntl.h>
#include<sys/stat.h>
#include<sys/types.h>
#include<linux/limits.h>
#include<linux/kdev_t.h>
#include<string.h>
int main(int argc, char *argv[]){

    struct stat statbuff;
    char filename[PATH_MAX] = "./";
    strcat(filename, argv[1]);
    if (stat(filename, &statbuff) == -1){
        perror("ERR:");
        exit(1);
    }

    printf("device number : %d\n", statbuff.st_dev);
    if((statbuff.st_mode & S_IFMT) == S_IFCHR || (statbuff.st_mode &
S_IFMT == S_IFBLK)){
        printf("Major: %d, Minor %d\n", MAJOR(statbuff.st_rdev),
MINOR(statbuff.st_rdev));
    }
}
```