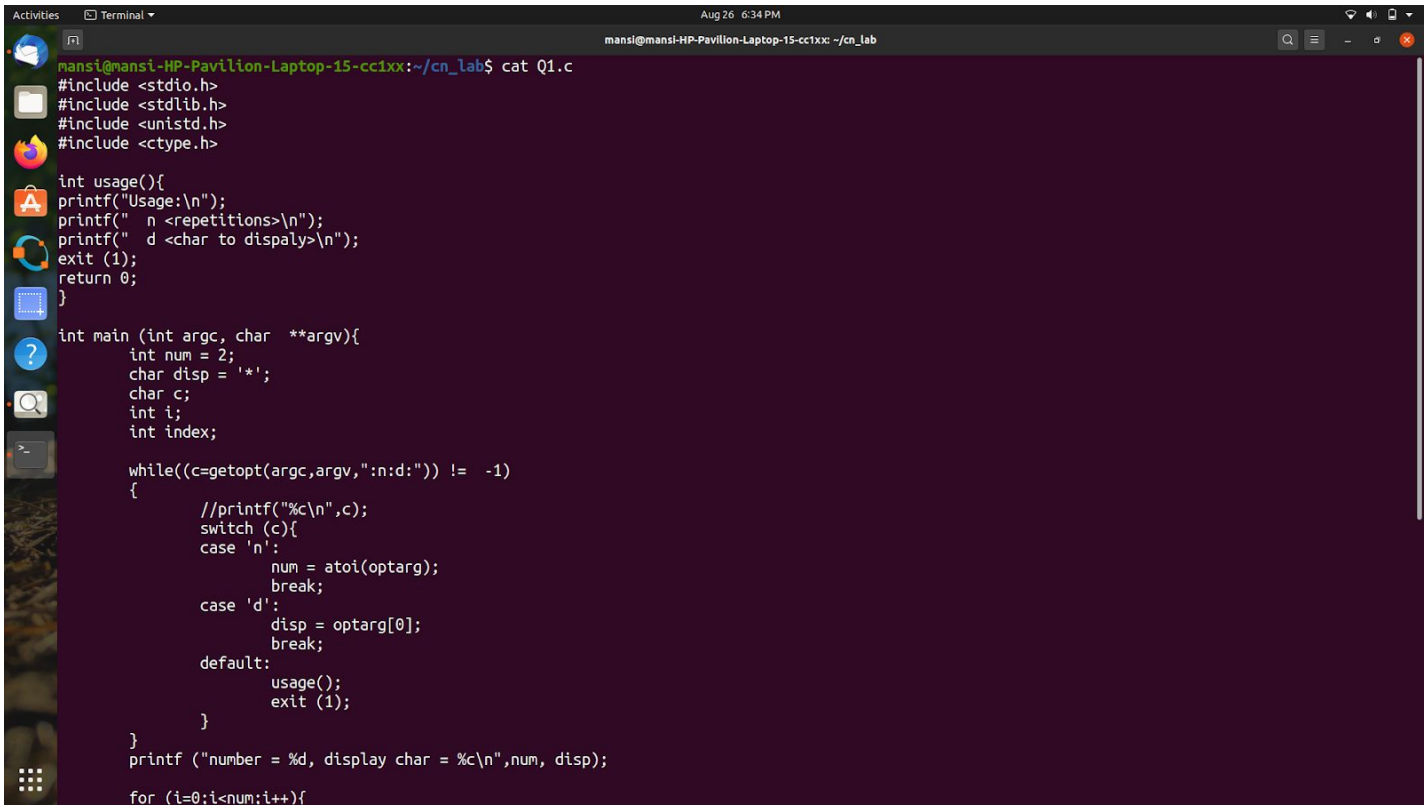
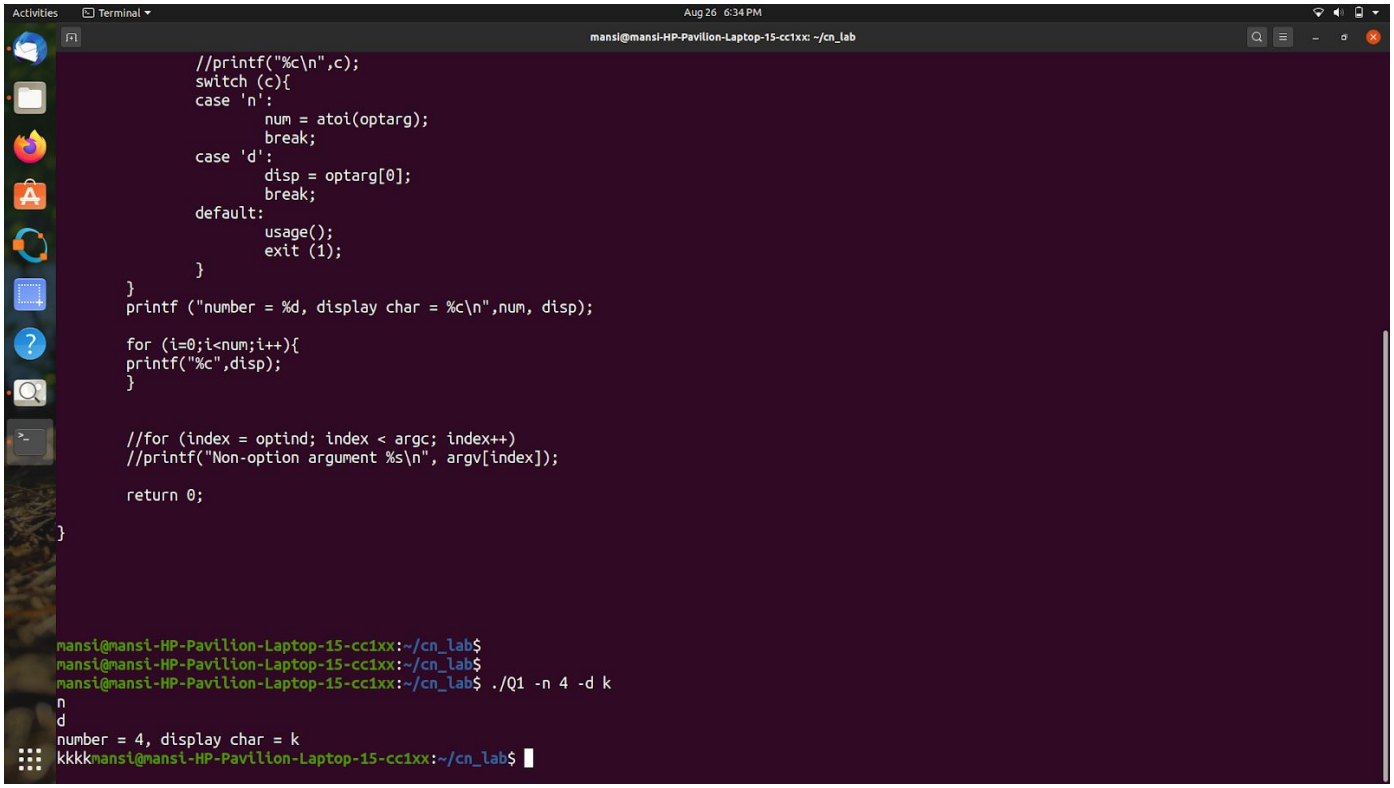
AU1841131 Mansi Dobariya
CN lab ass 1
Section 1

Q1)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <ctype.h>

int usage(){
printf("Usage:\n");
printf(" n <repetitions>\n");
printf(" d <char to dispaly>\n");
exit (1);
return 0;
}

int main (int argc, char  **argv){
   int num = 2;
   char disp = '*';
   char c;
   int i;
   int index;

   while((c=getopt(argc,argv,":n:d:")) !=  -1)
   {
          //printf("%c\n",c);
          switch (c){
          case 'n':
                    num = atoi(optarg);
                    break;
          case 'd':
                    disp = optarg[0];
                    break;
          default:
                    usage();
                    exit (1);
          }
   }
   printf ("number = %d, display char = %c\n",num, disp);
```

```
    for (i=0;i<num;i++){
    printf("%c",disp);
    }


    //for (index = optind; index < argc; index++)
            //printf("Non-option argument %s\n", argv[index]);

             return 0;

}
```

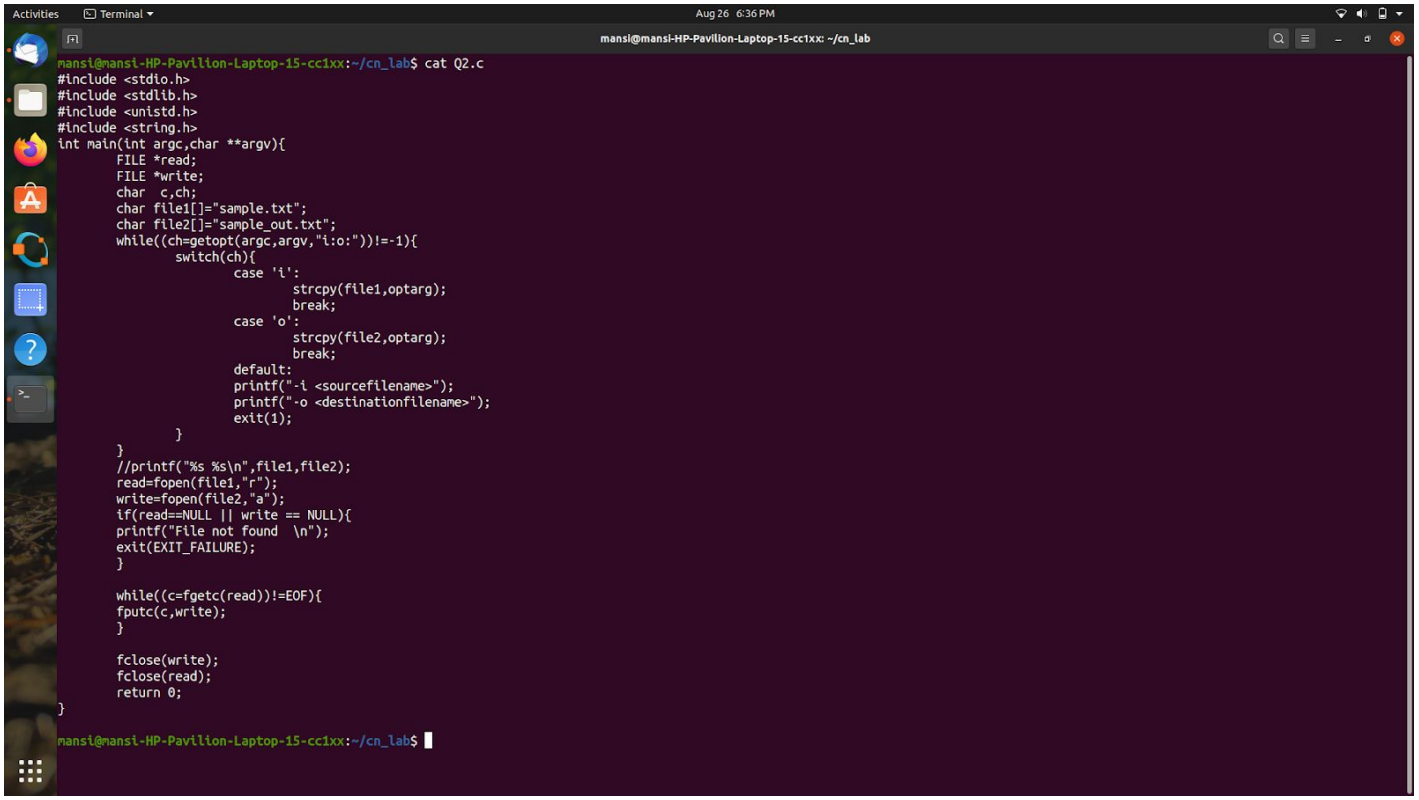


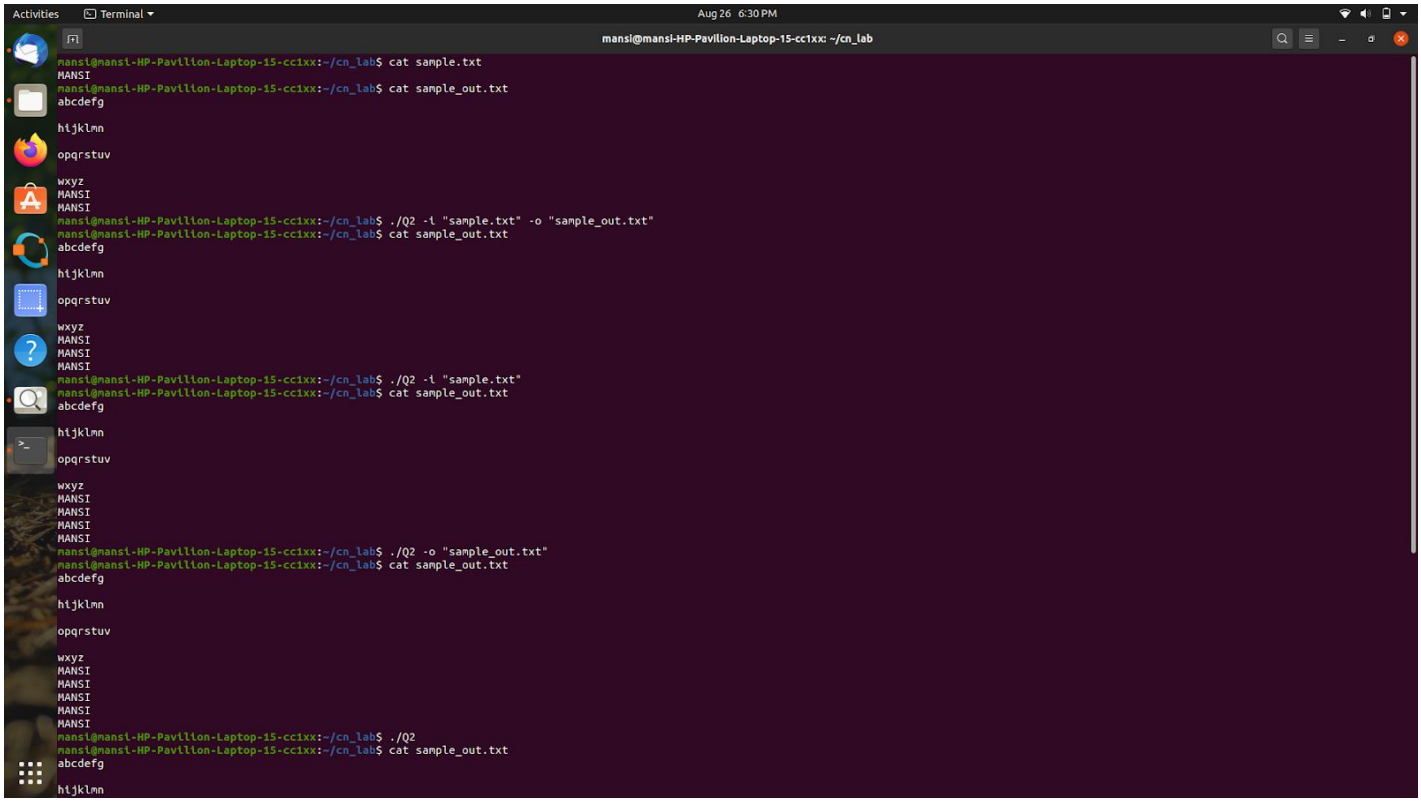


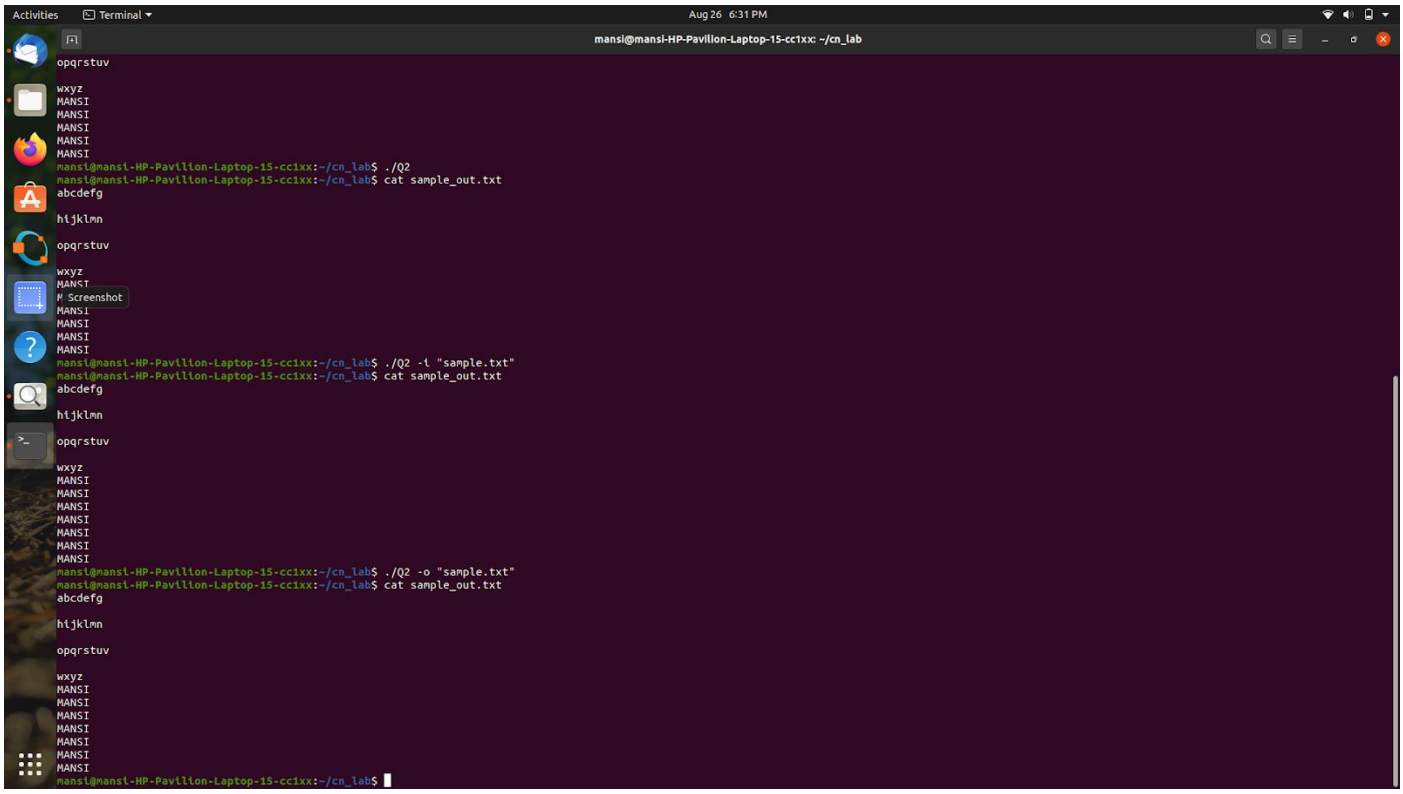
Q2)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
int main(int argc,char **argv){
    FILE *read;
    FILE *write;
    char  c,ch;
    char file1[]="sample.txt";
    char file2[]="sample_out.txt";
    while((ch=getopt(argc,argv,"i:o:"))!=-1){
            switch(ch){
                    case 'i':
                            strcpy(file1,optarg);
                            break;
                    case 'o':
                            strcpy(file2,optarg);
                            break;
                    default:
                    printf("-i <sourcefilename>");
```

```
                printf("-o <destinationfilename>");
                exit(1);
        }
    }
    //printf("%s %s\n",file1,file2);
    read=fopen(file1,"r");
    write=fopen(file2,"a");
    if(read==NULL || write == NULL){
    printf("File not found  \n");
    exit(EXIT_FAILURE);
    }

    while((c=fgetc(read))!=EOF){
    fputc(c,write);
    }

    fclose(write);
    fclose(read);
    return 0;
}
```



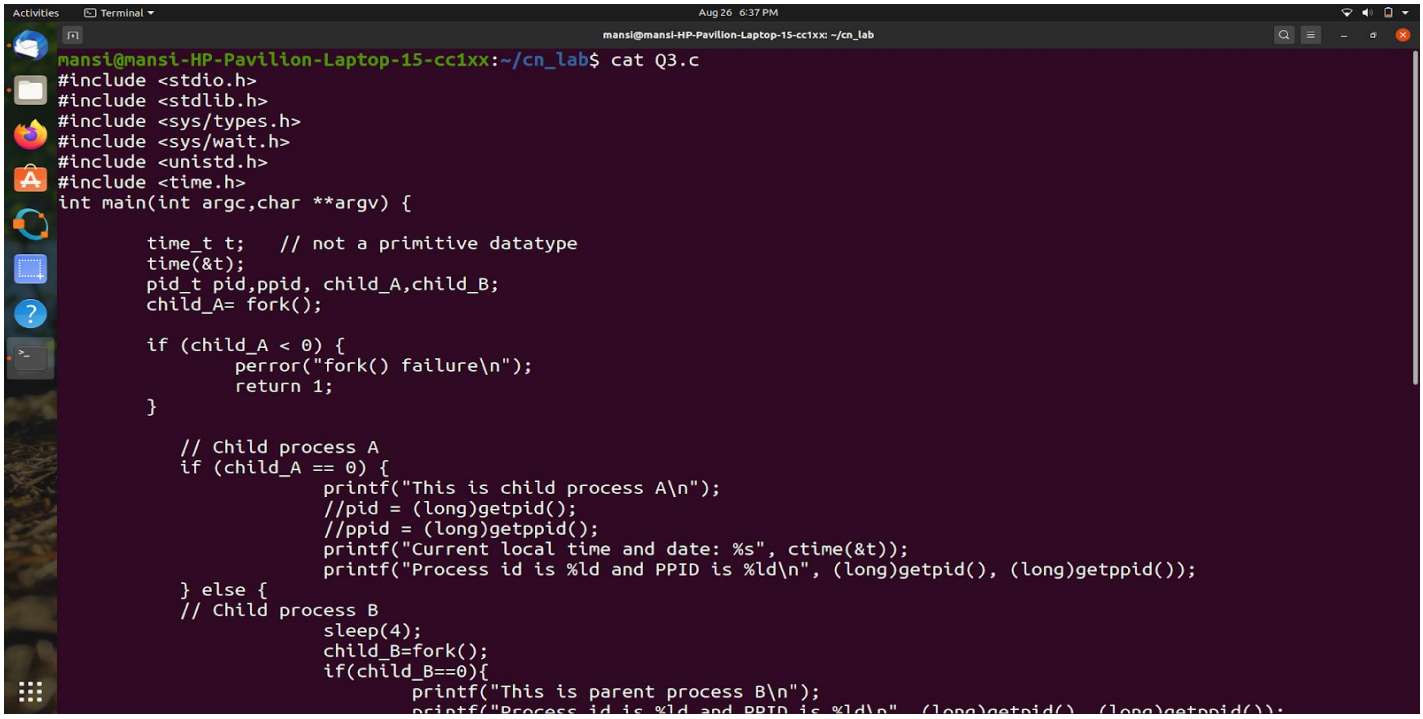

Activities
Terminal
Aug 26 6:30 PM
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx: ~/cn_lab
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat sample.txt
MANSI
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat sample_out.txt
abcdefg
hijklmn
opqrstuv
wxyz
MANSI
MANSI
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ ./Q2 -i "sample.txt" -o "sample_out.txt"
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat sample_out.txt
abcdefg
hijklmn
opqrstuv
wxyz
MANSI
MANSI
MANSI
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ ./Q2 -i "sample.txt"
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat sample_out.txt
abcdefg
hijklmn
opqrstuv
wxyz
MANSI
MANSI
MANSI
MANSI
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ ./Q2 -o "sample_out.txt"
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat sample_out.txt
abcdefg
hijklmn
opqrstuv
wxyz
MANSI
MANSI
MANSI
MANSI
MANSI
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ ./Q2
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat sample_out.txt
abcdefg
hijklmn


Activities
Terminal
Aug 26 6:31 PM
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx: ~/cn_lab
opqrstuv
wxyz
MANSI
MANSI
MANSI
MANSI
MANSI
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ ./Q2
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat sample_out.txt
abcdefg
hijklmn
opqrstuv
wxyz
MANSI
Screenshot
MANSI
MANSI
MANSI
MANSI
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ ./Q2 -i "sample.txt"
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat sample_out.txt
abcdefg
hijklmn
opqrstuv
wxyz
MANSI
MANSI
MANSI
MANSI
MANSI
MANSI
MANSI
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ ./Q2 -o "sample.txt"
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat sample_out.txt
abcdefg
hijklmn
opqrstuv
wxyz
MANSI
MANSI
MANSI
MANSI
MANSI
MANSI
MANSI
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$

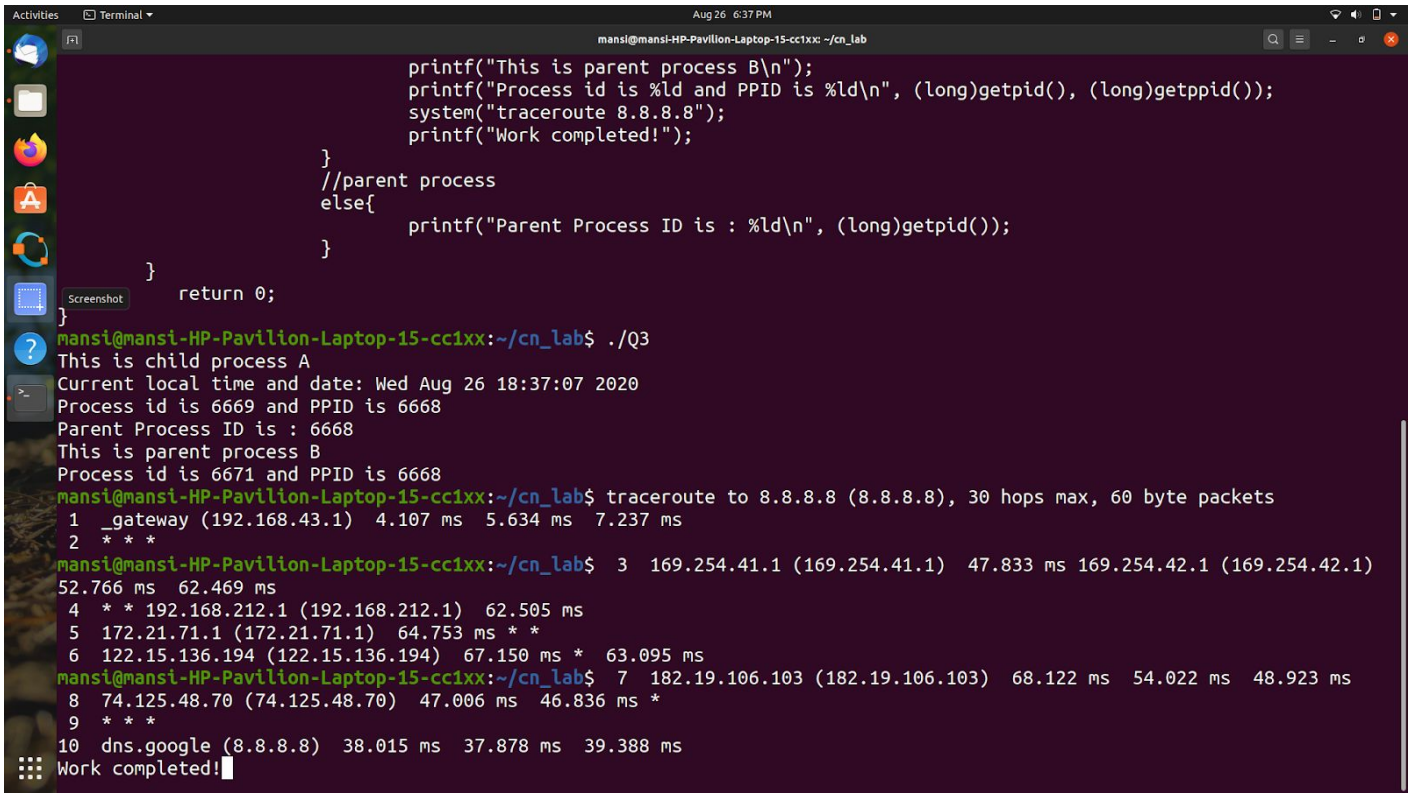
Q3)

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <time.h>
int main(int argc,char **argv) {

   time_t t;   // not a primitive datatype
   time(&t);
         pid_t pid,ppid, child_A,child_B;
   child_A= fork();

   if (child_A < 0) {
         perror("fork() failure\n");
         return 1;
   }

        // Child process A
        if (child_A == 0) {
                printf("This is child process A\n");
                //pid = (long)getpid();
                //ppid = (long)getppid();
                printf("Current local time and date: %s", ctime(&t));
                printf("Process id is %ld and PPID is %ld\n", (long)getpid(), (long)getppid());
        } else {
        // Child process B
                sleep(4);
                child_B=fork();
                if(child_B==0){
                        printf("This is parent process B\n");
                        printf("Process id is %ld and PPID is %ld\n", (long)getpid(),
(long)getppid());
                        system("traceroute 8.8.8.8");
                                printf("Work completed!");
                }
                //parent process
                else{
                                printf("Parent Process ID is : %ld\n", (long)getpid());
                }
```

```
    }
        return 0;
}
```



```
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ cat Q3.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <time.h>
int main(int argc,char **argv) {

        time_t t;   // not a primitive datatype
        time(&t);
        pid_t pid,ppid, child_A,child_B;
        child_A= fork();

        if (child_A < 0) {
                perror("fork() failure\n");
                return 1;
        }

           // Child process A
           if (child_A == 0) {
                        printf("This is child process A\n");
                        //pid = (long)getpid();
                        //ppid = (long)getppid();
                        printf("Current local time and date: %s", ctime(&t));
                        printf("Process id is %ld and PPID is %ld\n", (long)getpid(), (long)getppid());
           } else {
           // Child process B
                        sleep(4);
                        child_B=fork();
                        if(child_B==0){
                                printf("This is parent process B\n");
```



```
                                printf("This is parent process B\n");
                                printf("Process id is %ld and PPID is %ld\n", (long)getpid(), (long)getppid());
                                system("traceroute 8.8.8.8");
                                printf("Work completed!");
                        }
                        //parent process
                        else{
                                printf("Parent Process ID is : %ld\n", (long)getpid());
                        }
        }
            return 0;
}
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ ./Q3
This is child process A
Current local time and date: Wed Aug 26 18:37:07 2020
Process id is 6669 and PPID is 6668
Parent Process ID is : 6668
This is parent process B
Process id is 6671 and PPID is 6668
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$ traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  _gateway (192.168.43.1)  4.107 ms  5.634 ms  7.237 ms
 2  * * *
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$  3  169.254.41.1 (169.254.41.1)  47.833 ms 169.254.42.1 (169.254.42.1)
52.766 ms  62.469 ms
 4  * * 192.168.212.1 (192.168.212.1)  62.505 ms
 5  172.21.71.1 (172.21.71.1)  64.753 ms * *
 6  122.15.136.194 (122.15.136.194)  67.150 ms *  63.095 ms
mansi@mansi-HP-Pavilion-Laptop-15-cc1xx:~/cn_lab$  7  182.19.106.103 (182.19.106.103)  68.122 ms  54.022 ms  48.923 ms
 8  74.125.48.70 (74.125.48.70)  47.006 ms  46.836 ms *
 9  * * *
10  dns.google (8.8.8.8)  38.015 ms  37.878 ms  39.388 ms
Work completed!
```