Write C programs that simulate the following UNIX commands:

A. mv

B. cp

(Use system calls)

```
[KrishnaSai@H exp11]$ cat mymv3.c
#include<fcntl.h>
#include<stdio.h>
#include<unistd.h>
#include<sys/stat.h>
int main(int argc, char **argv)
{
if(argc>3 || argc<3)
{
            printf("Please Provide two arugments \n");
}
else{
  int fd1,fd2;
  int n,count=0;
if(access(argv[1],F_OK)<0)
      {
            printf("%s not found \n ",argv[1]);
      }
if(rename(argv[1],argv[2])==0)
printf(" %s is movied or renamed to %s \n  successfully \n",argv[1],argv[2]);
return (0);
}
}
```

Output:

```
 [KrishnaSai@H exp11]$ cc mymv3.c

[KrishnaSai@H exp11]$ ./a.out jkk fd
 jkk is movied or renamed to fd
  successfully
```

Before execution:

```
[KrishnaSai@H exp11]$ ls
a.out  jkk  mymv2.c  mymv3.c  mymv.c  t  u
```

After execution:

```
[KrishnaSai@H exp11]$ ls
a.out  fd  mymv2.c  mymv3.c  mymv.c  t  u
```

Program:

```
[CSESTAFF@localhost exp11]$ cat mycp.c
#include <stdio.h>
#include<fcntl.h>
#include<unistd.h>
#include<sys/stat.h>
#include <string.h>
#define BUF_SIZE 32
#define FILE_NAME_LEN 200
int main(int argc, char *argv[])
{
        FILE * file_to_read;
        FILE * file_to_write;
        char name_of_file_to_read[FILE_NAME_LEN+1];
        char name_of_file_to_write[FILE_NAME_LEN+1];
        char buf[BUF_SIZE];
        size_t num_rec;

if(argc>3 || argc<3)
{
                  printf("Please Provide two arugments \n");
}

else{
if(access(argv[1],F_OK)<0)
         {
                  printf("%s not found \n ",argv[1]);
         }

        /* Prepare the source file name */
        strcpy(name_of_file_to_read, argv[1]);
        /* Prepare the target file name */
        if ( argc == 3 )
                strcpy(name_of_file_to_write, argv[2]);
        else
                strcat(strcpy(name_of_file_to_write, name_of_file_to_read), ".fread");
        /* Open source file in read-only mode */
        if ( (file_to_read = fopen(name_of_file_to_read, "r")) == NULL )
        {
                fprintf(stderr, "Could not open file '%s' for reading\n",name_of_file_to_read);
                return 3;
        }
        /* Open target file in write mode */
        if ( (file_to_write = fopen(name_of_file_to_write, "w")) == NULL )
```

```
    {
        fprintf(stderr, "Could not open file '%s' for writing\n",
          name_of_file_to_write);
        fclose(file_to_read);
        return 4;
    }

while ( (num_rec = fread(buf, sizeof(char), BUF_SIZE, file_to_read) ) > 0 )
    {
   fwrite(buf, sizeof(char), num_rec, file_to_write);

     if ( ferror(file_to_write) )
        {
            fprintf(stderr, "Error while writing into file '%s'\n",
              name_of_file_to_write);
            fclose(file_to_read);
            fclose(file_to_write);
            return 5;
        }
    }
if ( ferror(file_to_read) )
    {
        fprintf(stderr, "Error while reading the file '%s'\n", name_of_file_to_read);
        fclose(file_to_read);
        fclose(file_to_write);
        return 6;
    }
    /* Close the files */
    fclose(file_to_read);
    fclose(file_to_write);
    printf("File '%s' successfully copied to file '%s'\n", name_of_file_to_read,
     name_of_file_to_write);
    return 1;
}
}
[CSESTAFF@localhost exp11]$
```

Output:

```
[CSESTAFF@localhost exp11]$ cc mycp.c
[CSESTAFF@localhost exp11]$ ls
a.out  fd  ff  mycp.c  mymv2.c  mymv3.c  mymv.c  re  t  u
[CSESTAFF@localhost exp11]$ ./a.out fd ff
File 'fd' successfully copied to file 'ff'
[CSESTAFF@localhost exp11]$
```