4. Develop a C program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.

ALGORITHM:

 It is an extension to the traditional pipe concept on Unix. A traditional pipe is "unnamed" and lasts only as long as the process.

 A named pipe, however, can last as long as the system is up, beyond the life of the process. It can be deleted if no longer used.

 Usually a named pipe appears as a file and generally processes attach to it for inter-process communication. A FIFO file is a special kind of file on the local storage which allows two or more processes to communicate with each other by reading/writing to/from this file.

 A FIFO special file is entered into the filesystem by calling *mkfifo()* in C. Once we have created a FIFO special file in this way, any process can open it for reading or writing, in the same way as an ordinary file. However, it has to be open at both ends simultaneously before you can proceed to do any input or output operations on it.

CODE:

```
/*writer process */
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
    int fd;
    char buf[1024];
    char * myfifo = "/tmp/myfifo";

    mkfifo(myfifo, 0666);
    printf("Run Reader process to read the FIFO File\n");
    fd = open(myfifo, O_WRONLY);
    write(fd,"Hi", sizeof("Hi"));                /* write "Hi" to the FIFO */
```

```
    close(fd);

    unlink(myfifo); /* remove the FIFO */

    return 0;

}
```

OUTPUT:

```
Run Reader process to read the FIFO File
```

/* Reader Process */

```
#include <fcntl.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <unistd.h>

#include <stdio.h>

#define MAX_BUF 1024


int main()
{
    int fd;
    /* A temp FIFO file is not created in reader */
    char *myfifo = "/tmp/myfifo";
    char buf[MAX_BUF];
    /* open, read, and display the message from the FIFO */
    fd = open(myfifo, O_RDONLY);
    read(fd, buf, MAX_BUF);
    printf("Writer: %s\n", buf);
    close(fd);
    return 0;
}
```

OUTPUT:

Writer: Hi