**Program Requirements**

Your program shall be named mfs.c and shall be implemented in C or C++. You shall not use the

system calls system() or any of the exec family of system calls.

Your program shall print out a prompt of mfs> when it is ready to accept input.

**The following commands shall be supported**:

open <filename>

This command shall open a fat32 image. Filenames of fat32 images shall not contain spaces and

shall be limited to 100 characters.

If the file is not found your program shall output: “Error: File system image not found.”. If a file

system is already opened then your program shall output: “Error: File system image already

open.”.

close

This command shall close the fat32 image. If the file system is not currently open your program

shall output: “Error: File system not open.” Any command issued after a close, except for

open, shall result in “Error: File system image must be opened first.”

info

This command shall print out information about the file system in both hexadecimal and base 10:

• BPB\_BytesPerSec

• BPB\_SecPerClus

• BPB\_RsvdSecCnt

• BPB\_NumFATS

• BPB\_FATSz32

stat <filename> or <directory name>

This command shall print the attributes and starting cluster number of the file or directory name.

If the parameter is a directory name then the size shall be 0. If the file or directory does not exist

then your program shall output “Error: File not found”.

get <filename>

This command shall retrieve the file from the FAT 32 image and place it in your current working

directory. If the file or directory does not exist then your program shall output “Error: File not

found”.

put <filename>

This command shall retrieve the file from the current working directory and place it in your FAT

32 image. If the file or directory does not exist then your program shall output “Error: File not

found”.

cd <directory>

This command shall change the current working directory to the given directory. Your program

shall support relative paths, e.g cd ../name and absolute paths.

ls

Lists the directory contents. Your program shall support listing “.” and “..” . Your program shall

not list deleted files or system volume names.

read <filename> <position> <number of bytes>

Reads from the given file at the position, in bytes, specified by the position parameter and output

the number of bytes specified.