Name:

FREAD(3) BSD Library Functions Manual FREAD(3)

**NAME**

 **fread**, **fwrite** -- binary stream input/output

**LIBRARY**

 Standard C Library (libc, -lc)

**SYNOPSIS**

 **#include** **<stdio.h>**

 size\_t

 **fread**(void \*restrict ptr, size\_t size, size\_t nitems,

 FILE \*restrict stream);

 size\_t

 **fwrite**(const void \*restrict ptr, size\_t size, size\_t nitems,

 FILE \*restrict stream);

**DESCRIPTION**

 The function **fread**() reads nitems objects, each size bytes long, from the

 stream pointed to by stream, storing them at the location given by ptr.

 The function **fwrite**() writes nitems objects, each size bytes long, to the

 stream pointed to by stream, obtaining them from the location given by

 ptr.

**RETURN** **VALUES**

 The functions **fread**() and **fwrite**() advance the file position indicator

 for the stream by the number of bytes read or written. They return the

 number of objects read or written. If an error occurs, or the end-of-

 file is reached, the return value is a short object count (or zero).

 The function **fread**() does not distinguish between end-of-file and error;

 callers must use feof(3) and ferror(3) to determine which occurred. The

 function **fwrite**() returns a value less than nitems only if a write error

 has occurred.

**SEE** **ALSO**

 read(2), write(2)

Name:

A binary file, x.raw, contains the following data (in this order): 32-bit float, 8-bit byte, 16-bit short int, 64-bit integer. Below is code to open and close the file. Add code to read the above data from the file and store the correct value in a variable. The man page for fread is shown on the previous page. Assume you are using a 64-bit processor and your compiler has been built with 64-bit support ☺

int main(int argc, char\*\* argv) {

FILE\* fp=fopen(“x.raw”, “r”);

fclose(fp);

}

Code that needs to be added: