CSSE 132 – Introduction to Systems Programming
Rose-Hulman Institute of Technology

Worksheet 18 Input/Output I

Name (Print):_____ RHIT Username:_____

1. How does Linux handle input/output to and from external devices such as hard-drives, keyboards, network communications, etc?

   Linux utilizes an idea called [ ], *which treats the external devices as* [ ] *and then uses* [ ] *to communicate with them in writing/reading to and from files.*

2. What are these "files" used for?

   `stdin`: [ ] `stdout`: [ ]

3. Complete the following code such that when executing this program, it takes a filename as the command-line input. Then it opens the file, reads the data, and prints it on the screen. Check man page for usage of `fopen, fread, fclose`.

```c
#include <stdio.h>

void main(int argc, char** argv) {
    char buff[256];
    FILE* read_f = fopen(__________, ______);
    if(_________________){
        printf("Failed to open the file\n");
        return;
    }
    int b = ____________;
    while (_________) {
        b = fread(____, ____, ____, ____);
        buff[_____] = '\0';
        printf("%s", buff);
    }
    fclose(_________);
}
```

4. Assuming a file is opened already and represented as `read_f`, finish the following `fseek` function calls to move the cursor to specified places.

   Skip the first 40 bytes:
   `fseek(read_f, _____, ________)`

   To the end of the file:
   `fseek(read_f, _____, ________)`

5. Complete the following code such that when executed it will *copy/save* **the second half** *of a file to another new file*. For example, if the program is named `a.out`, when executed with the command: `./a.out src_file.txt dst_file.txt`, then it will save the second half of `src_file.txt` to another new file `dst_file.txt`
   A useful function is `fprintf`, which behaves similarly to `printf`. However, instead of printing on the screen, `fprintf` prints into a file, specified by its first input parameter. The rest of the input parameters are identical to those of `printf`.

```c
#include <stdio.h>

void main(int argc, char** argv) {
    char buff[128];
    FILE* read_f;
    FILE* write_f;

    read_f = fopen(___________, ______);
    write_f = fopen(__________, ______);

    fseek(_________, 0, ____________);
    int len = __________(read_f);
    fseek(____________, _________, SEEK_SET);

    int b = _________________;
    while (b > 0) {
        b = fread(buff, 1, __________, read_f);
        //print buff as a string into write_f
        buff[__________] = '\0';
        fprintf(___________, _________, buff);
    }

    fclose(read_f);
    fclose(write_f);
}
```