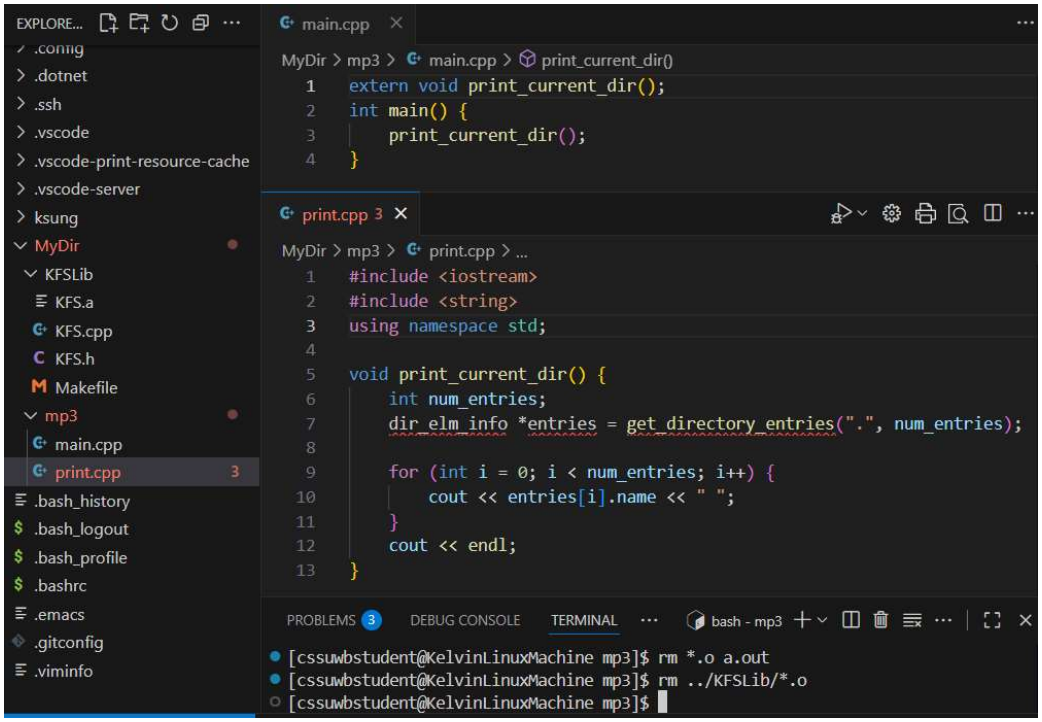
Quiz 4 Name:_____

Question 1 (10pt): *Yes, this is exactly the same as Q1 from last quiz*
Here is my initial attempt at mp3. Notice, I have decided to implement my program in two files: **main.cpp** (top window) and **print.cpp** (bottom window).



1. (2pt) Pay attention to **main.cpp**, will the 4-lines of source code compile? That is, will the **main.o** be generated if I issue the command: `g++ -c main.cpp`? Please explain your answer in one sentence. Answers without explanation will not receive any credit.

2. (2pt) Please explain how you will fix the problems with **print.cpp** such that I can successfully compile the source code file.

3. (2pt) After fixing all of the problems, please list all of the commands, one command per line, that you have to issue in order to compile and link a program named **MyProg**.

Question 2. (5pt)

Given:

```
void fFunc(int*);
void gFunc(int&);

int main() {
    int xVar = 10;
    int yVar = 20;
    int zVar = 30;

    // assume your code comes here
}
```

a. (1pt) Please show how to declare and initialize a pointer, **`ptr`**, to point to the variable **`xVar`**.

int *ptr(&aVar);

b. (1pt) Please show how you can assign the value of **`yVar`** to **`xVar`** using **`ptr`**.

*ptr = bVar;

c. (1pt) Please show how you can assign the value of **`xVar`** to **`zVar`** using **`ptr`**.

zVar = *ptr;

d. (1pt) Please show how you can call the **`fFunc()`** function with **`yVar`** as the argument.

fFunc(&yVar);

e. (1pt) Please show how you can call the **`gFunc()`** function with **`ptr`** as the argument.

gFunc(*ptr);

Question 3. (10pt)
Recall the declaration of `dir_elm_info`:

```
struct dir_elm_info {     // this is the data type
    string path;            // full path name
    string name;            // file or directory name only
    bool is_directory;      // true if directory, false otherwise
};
```

And, that the library function:

```
dir_elm_info* get_directory_entries(const string &dir_path, int &num_entries);
   // Function to get directory entries
   // dir_path: path to the directory to check (full or relative path)
   // num_entries: output parameter to return number of entries found
   // returns: pointer to an array of dir_elm_info structures
```

Please show how can implement the recursive function:

```
int num_dir_entries(const string &dir_path);
   // Function: num_dir_entries
   //     dir_path: path to a directory
   // Precondition: dir_path is a valid directory path
   // Postcondition: returns number of entries in the directory and
   //                all of its subdirectories
```

You can assume `dir_path` is a valid path to an existing directory.

```
int num_dir_entries(const string &dir_path) {
    int count = 0;
    dir_elm_info* entries = nullptr;
    int num_entries = 0;

    entries = get_directory_entries(dir_path, num_entries);

    // Count the number of entries
    for (int i = 0; i < num_entries; ++i) {
        count++;
        if (entries[i].is_directory) {
            count += num_dir_entries(entries[i].path);
        }
    }
    delete[] entries; // Clean up dynamically allocated memory
    return count;
}
```