

Quiz 3

Name: _____

Question 1 (6pt)

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).

```
EXPLORE...  main.cpp  x
MyDir > mp3 >  main.cpp >  print_current_dir()
1  extern void print_current_dir();
2  int main() {
3      print_current_dir();
4  }

print.cpp 3  x
MyDir > mp3 >  print.cpp > ...
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  void print_current_dir() {
6      int num_entries;
7      dir elm info *entries = get_directory_entries(".", num_entries);
8
9      for (int i = 0; i < num_entries; i++) {
10         cout << entries[i].name << " ";
11     }
12     cout << endl;
13 }

PROBLEMS 3  DEBUG CONSOLE  TERMINAL  ...  bash - mp3
[cssuwbstudent@KelvinLinuxMachine mp3]$ rm *.o a.out
[cssuwbstudent@KelvinLinuxMachine mp3]$ rm ../KFSLib/*.o
[cssuwbstudent@KelvinLinuxMachine mp3]$
```

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.

Yes. All symbols in the file are defined.

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

Need to include KFS.h.

#include "../KFSLib/KFS.h"

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**.

g++ -c main.cpp print.cpp

g++ -o MyProg main.o print.o ../KFSLib/KFS.a

Question 2. (14pt)

- a. (4pt) Write an **iterative** function, `PrintX(int d)`, where,

$$0 \leq d \leq 9$$

The function should print out d-number of **x**, or a `_` (the underscore character) if d is 0. In all cases, the printing should end with a space.

For example :

`d = 2` → prints out: `xx` // two x followed by a space

`d = 0` → prints out: `_` // underscore followed by a space

Take note that the grey highlight is for clear indication of the space character and that your function should **not** print any ***endl***.

```
void PrintX(int d) {  
    if (d > 0) {  
        for (int i = 0; i < d; i++) {  
            cout << "x";  
        }  
    } else {  
        cout << "_";  
    }  
    cout << " ";  
}
```

You are given the following function:

```
int RemoveRight(int &n) {  
    int right_digit = n % 10;  
    n = n / 10;  
    return right_digit;  
}
```

Recall that this function returns the right-most-digit and modifies the input number, *n*, by removing the right-most-digit.

- b. (6pt) Write a **recursive** function, `PrintDigitR(int n)`, that calls `RemoveRight()` and your `PrintX()` to print x's according to the digits in the given number, *n*. For example:

`n = 103` → prints out: `xxx _ x` // 3 followed by 0, and then 1

`n = 201` → prints out: `x _ xx` // 1 followed by 0, and then 2

Make sure you pay attention to the order of printing: the x's that correspond to the right-most-digit are printed first.

```
void PrintDigitR(int n) {  
    int right_digit = RemoveRight(n);  
    PrintX(right_digit);  
    if (n > 0)  
        PrintDigitR(n);  
}
```

- c. (4pt) Now, write a new version of the **recursive** print function, `PrintDigitL(int n)`. Once again, calling `RemoveRight()` and your `PrintX()` to print digit x's in the same order as the digits are found in *n*. For example:

`n = 103` → prints out: `x _ xxx` // 1 followed by 0, and then 3

`n = 201` → prints out: `xx _ x` // 2 followed by 0, and then 1

Make sure you pay attention to the order of printing: the x's that correspond to the left-most-digit are printed first.

```
void PrintDigitL(int n) {  
    int right_digit = RemoveRight(n);  
    if (n > 0)  
        PrintDigitL(n);  
    PrintX(right_digit);  
}
```

Note the one-line difference when compare to `PrintDigitR`