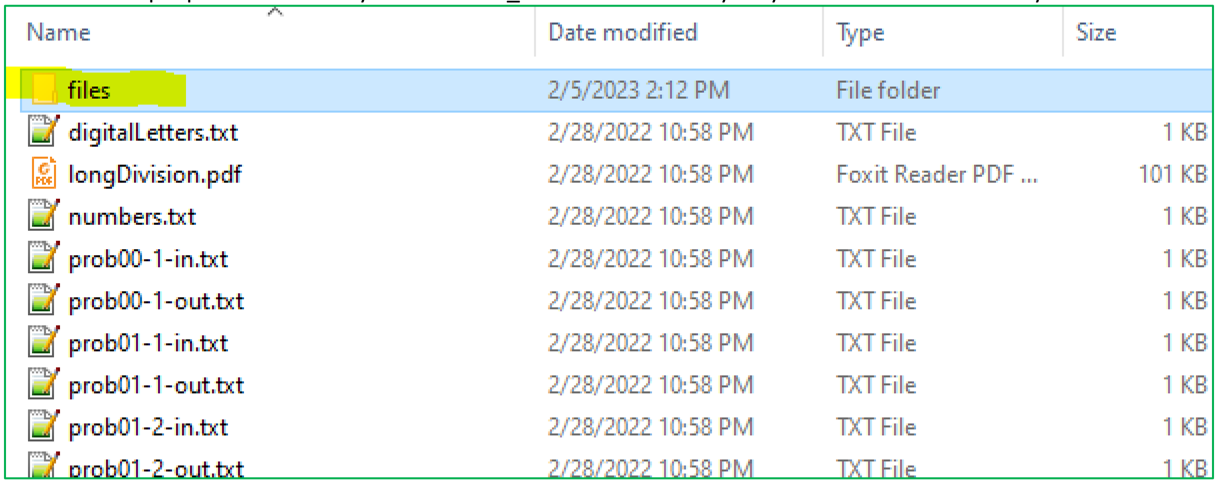
Reading & Using Directory Information

Starting in CodeWars 2023 we will be introducing problems which will require searching / using / reading / traversing directories to solve some problems. All directory problems will start in a directory inside your "student_datasets" directory named "files". That means, in addition to the expected: prob00-1-in.txt,... prob07-in.txt,...prob29-2-out.txt files you usually find in that folder, you will now also find a directory named "files". That directory will contain various sub-directories and files related to some of the problems in the packet.

To solve those problems your code will need to be able to look inside that directory and search for files and other directories relative to that path. E.G. your code already needs to be able to read input files from the same location where your solution code file runs. Now it also needs to be able to open and read the contents of a "files" directory in that same location as well.

(Note, "problem01" is given as an example sub-directory in the code samples below, that directory name is not guaranteed to be present in the actual "files" directory on contest day. Only the "files" directory is guaranteed to exist.)

Here is a sample picture of what your "student_datasets" directory may look like on contest day:



Name	Date modified	Type	Size
files	2/5/2023 2:12 PM	File folder	
digitalLetters.txt	2/28/2022 10:58 PM	TXT File	1 KB
longDivision.pdf	2/28/2022 10:58 PM	Foxit Reader PDF ...	101 KB
numbers.txt	2/28/2022 10:58 PM	TXT File	1 KB
prob00-1-in.txt	2/28/2022 10:58 PM	TXT File	1 KB
prob00-1-out.txt	2/28/2022 10:58 PM	TXT File	1 KB
prob01-1-in.txt	2/28/2022 10:58 PM	TXT File	1 KB
prob01-1-out.txt	2/28/2022 10:58 PM	TXT File	1 KB
prob01-2-in.txt	2/28/2022 10:58 PM	TXT File	1 KB
prob01-2-out.txt	2/28/2022 10:58 PM	TXT File	1 KB

Below we have written code examples showing how to list files and directories from a relative computer path using all of the programming languages currently supported for CodeWars.

PYTHON:

```python
import os
path = 'files\\problem01'
files = os.listdir(path)
for item in files:
    if os.path.isfile(os.path.join(path,item)):
        print(f'{item} is a file')
    elif os.path.isdir(os.path.join(path,item)):
        print(f'{item} is a directory')
    else:
        print(f'{item} is something else')
```

JAVA:

```java

import java.io.File;
public class DirectoryIOExample {
    public static void main(String[] args) {
        File path = new
File(System.getProperty("user.dir")+"\\files\\problem01");
        File[] files = path.listFiles();
        for (File f:files) {
            if (f.isFile()) {
                System.out.println(f.getName() + " is a file");
            }
            else if (f.isDirectory()) {
                System.out.println(f.getName() + " is a directory");
            }
            else {
                System.out.println(f.getName() + " is something else");
            }
        }
    }
}
```

C:

```c
#include <stdio.h>
#include <dirent.h>
#include <sys/stat.h>
int main(void)
{
    struct dirent *de;  // Pointer for directory entry
    struct stat s; //tells us stats about a file
    DIR *dr = opendir("files\\problem01");
    if (dr == NULL){
        printf("Could not open current directory" );
        return 0;
    }
    while ((de = readdir(dr)) != NULL){
        char path[] = "files\\problem01";
        strcat(path, de->d_name);
        if(stat(path,&s) == 0 ){
            if( s.st_mode & S_IFDIR ){
                printf("%s is a directory\n", de->d_name);
            }
            else if( s.st_mode & S_IFREG ){
                printf("%s is a file\n", de->d_name);
            }
            else{
                printf("%s is something else\n", de->d_name);
            }
        }
        else{
            printf("Error\n");
        }
    }
    closedir(dr);
    return 0;
}
```

C++:

```cpp
#include <filesystem>
#include <iostream>
#include <string>
#include <sys/stat.h>
namespace fs = std::filesystem;
int main()
{
    std::string path = "files\\problem01";
    struct stat sb;
    std::string output = "";
    for (const auto& entry : fs::directory_iterator(path)) {
        std::filesystem::path outfilename = entry.path();
        std::string outfilename_str = outfilename.string();
        const char* path = outfilename_str.c_str();
        output = "";
        if (stat(path, &sb) == 0 && !(sb.st_mode & S_IFDIR))
        {
            output.append(path).append(" is a file\n");
            std::cout << output;
        }
        else{
            output.append(path).append(" is a directory\n");
            std::cout << output;
        }
    }
}
```