**CPP Problem Design**

|  |
| --- |
| **Subject: Copy Folder** |
| **Contributor: 王聖文** |
| **Main testing concept:** STREAMS AND FILE I/O、RECURSION   |  |  | | --- | --- | | **Basics** | **Functions** | | ■ C++ BASICS  ■ FLOW OF CONTROL  ■ FUNCTION BASICS  □ PARAMETERS AND OVERLOADING  □ ARRAYS  □ STRUCTURES AND CLASSES  □ CONSTRUCTORS AND OTHER TOOLS  □ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES  □ STRINGS  □ POINTERS AND DYNAMIC ARRAYS | □ SEPARATE COMPILATION AND NAMESPACES  ■ STREAMS AND FILE I/O  ■ RECURSION  □ INHERITANCE  □ POLYMORPHISM AND VIRTUAL FUNCTIONS  □ TEMPLATES  □ LINKED DATA STRUCTURES  □ EXCEPTION HANDLING  □ STANDARD TEMPLATE LIBRARY  □ PATTERNS AND UML | |
| **Description:**  Please implement FileController class with the following functions:   |  |  | | --- | --- | | Function | Description | | 1. Check if the file exists   bool isFileExist(string filePath); | Check if the file specified in the filePath (path + file name) exists. If it exists and is a file, return true. Otherwise return false. | | 1. Check if the folder exists   bool isFolderExist(string path); | Check if the folder specified in the filePath (path + file name) exists. If it exists and is a folder, return true. Otherwise return false. | | 1. Get the byte size of the file   long long getFileSize(string filePath); | Get the byte size of the file specified in the filePath (path + filename).  The byte size of the testing file will not exceed the long long range. And the file specified in the parameter always exists. | | 1. Display all files and folders in the folder (regardless of order)   bool showFolderContents(bool showDetail, string path); | If the folder specified in the filePath (path + file name) exists. Display all files and folders in the folder and return true. Otherwise return false.  The format of the output on the screen is as follows: (Do not need to display file size for folders.)  path:  \tfolder name or file name\tfile size\tunit  \tfolder name or file name\tfile size\tunit  \tfolder name or file name\tfile size\tunit  …  You only need to display the file’s size and unit if showDetail=true.(with simple float output, do not set decimal precision). If the file size is greater than or equal to 1MB, the unit is displayed as MB; if the file size is 1KB~1MB, the unit is displayed as KB; the rest is displayed as byte(s). | | 1. Copy file   bool copyFile(string source, string target); | If the source exists and is a file, copy the source (path + file name) file to the target (path + file name), otherwise return false. The file name cam be different.  The path of the target (path + file name) will be an existing path. You won’t need to add any additional folders. | | 1. Copy folder   bool copyFolder(string source, string target); | If the source exists and is a folder, copy all the files in the source (path + folder name) folder to the target (path) and return true, otherwise return false.  The path of the target (path) will be an existing path. |   You need to use the “data” folder provided with this problem, and place this “data” folder under the same folder as the main.cpp. Please do not change the file in the “data” folder, otherwise errors may happen.  Under the “data” folder, copy1, copy2, and copy3 are the folders for the test results. Please clear them after each test, otherwise errors may happen.  **Input:**  We will replace the main.cpp file.  **Output:**  Output files, folders or related information according to this problem.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | #include "FileController.h"  int main()  {  FileController fc;  fc.showFolderContents(false, "data\\source");  cout << (fc.isFileExist("data\\source\\test0.txt") ? "true" : "false") << endl;  cout << (fc.isFileExist("data\\source\\test1.txt") ? "true" : "false") << endl;  cout << (fc.isFileExist("data\\source\\PIC") ? "true" : "false") << endl;    cout << (fc.isFolderExist("data\\source\\test0.txt") ? "true" : "false") << endl;  cout << (fc.isFolderExist("data\\source\\test1") ? "true" : "false") << endl;  cout << (fc.isFolderExist("data\\source\\PIC") ? "true" : "false") << endl;    cout << fc.getFileSize("data\\source\\NTUST.png") << "\tbyte(s)" << endl;  cout << fc.getFileSize("data\\source\\test1.txt") << "\tbyte(s)" << endl;  cout << (fc.copyFile("data\\source\\PIC\\PNG\\Stanford.PNG", "data\\copy1\\STF.PNG") ? "Copy successful" : "Copy failed") << endl;  fc.showFolderContents(true, "data\\copy1");  cout << (fc.copyFolder("data\\source\\test", "data\\copy1") ? "Copy successful" : "Copy failed") << endl;  cout << (fc.copyFolder("data\\source\\School", "data\\copy1") ? "Copy successful" : "Copy failed") << endl;  fc.showFolderContents(true, "data\\copy1");  fc.showFolderContents(true, "data\\copy1\\NTUST");  system("pause");  return 0;  } | data\source:  123.gif  NTUST.png  NTUST\_CSIE.docx  NTUST\_CSIE.pdf  PIC  School  Stanford.PNG  test1.txt  test2.txt  false  true  false  false  false  true  5561 byte(s)  3 byte(s)  Copy successful  data\copy1:  STF.PNG 3.70888 MB  Copy failed  Copy successful  data\copy1:  NTUST  other  STF.PNG 3.70888 MB  data\copy1\NTUST:  NTUST.png 5.43066 KB  NTUST\_CSIE.docx 12.874 KB  NTUST\_CSIE.pdf 179.646 KB  Press any key to continue . . . | |
| **□ Easy,Only basic programming syntax and structure are required.**  **□ Medium,Multiple programming grammars and structures are required.**  **■ Hard,Need to use multiple program structures or complex data types.** |
| **Expected solving time:**  45 minutes |
| **Other notes:**  The “starting path” of the program execution is under the same folder as the main.cpp file, and the path will be a relative path starting from the starting “path”. There will be no path to the parent layer ("..\").  The files in this problem contain various types of files. Using binary copy is recommended.  The file name of the file will contains the file extension. |