printBox

Language/Type: C++ basics streams file input
Related Links: string istream

Write a function named **printBox** that accepts two parameters: a string holding a file name, and an integer for a width. Your function should open that file and reads its contents as a sequence of lines, and display the lines to the console with a 'box' border of # characters around them on all four sides. The second parameter to your function indicates the total width of the box including its border. You must also convert each line to "title case" by capitalizing the first letter of the line and lowercasing all subsequent letters. For example, suppose the file `poem.txt` contains the following text:

```
roses ARE red
VIOLETS Are blUE

All my BASE
ARE belong To YOU
```

Then the following calls would produce the following console output. If any lines in the file are too long to fit into the box, truncate them.

printBox("poem.txt", 19);	printBox("poem.txt", 30);	printBox("poem.txt", 7);
`` ################### #Roses are red # #Violets are blue # # # #All my base # #Are belong to you# ################### ```	``` ############################### #Roses are red # #Violets are blue # # # #All my base # #Are belong to you # ############################### ```	``` ####### #Roses# #Viole# # # #All m# #Are b# ####### ```

If the width value passed is less than 2, throw an `int` exception. Notice that the file might contain blank lines. If the input file does not exist or is not readable, your function should print no output. Your solution should read the file only once, not make multiple passes over the file data.

```
1  void printBox(string filename, int width)
2  {
3      if (width < 2) {
4          throw -1;
5      }
6
7      ifstream file(filename);
8      if (file) {
9
10         string line;
```

```
    vector<string> data;

    while (getline(file, line)) {

        // turncate lines
        if (width - 2 < line.length()) {
            line.erase((width - 2));
        }
        // Title Case
        if (line != "") {
            for (auto& c : line) {
                c = tolower(c);
            }
            line[0] = toupper(line[0]);
        }
        // store in data
        data.push_back(line);
    }

    // output
    const char border = '#';
    const size_t rows = data.size() + 2;
    const size_t cols = width;

    size_t r  = 0;
    size_t ln = 0;
    while (r < rows) {

        size_t c = 0;
        while (c < cols) {

            if (r == 0 || r == rows - 1 || c == 0 || c == cols - 1) {
                cout << border;
                c++;

            } else if (c == 1 && r != 0 && r != rows - 1) {

                if (data.at(ln) == "") {
                    cout << ' ';
                    c++;
                    ln++;

                } else {
                    cout << data.at(ln);
                    c += data.at(ln).length();
                    ln++;
                }

            } else {
                cout << ' ';
                c++;
            }
        }
        cout << '\n';
        r++;
```

```
66          }
67      }
68  }
```

Function: Write a C++ function as described, not a complete program.

Submit

You passed 7 of 7 tests.

test #1: `printBox("printBox-test1-data.txt", 19);`

file input: printBox-test1-data.txt:

```
roses ARE red
VIOLETS Are blUE

All my BASE
ARE belong To YOU
```

console output:

```
####################
#Roses are red     #
#Violets are blue  #
#                  #
#All my base       #
#Are belong to you#
####################
```

result: pass

test #2: `printBox("printBox-test1-data.txt", 30);`

file input: printBox-test1-data.txt:

```
roses ARE red
VIOLETS Are blUE

All my BASE
ARE belong To YOU
```

console output:

```
################################
#Roses are red                 #
#Violets are blue              #
#                              #
#All my base                   #
#Are belong to you             #
################################
```

result: pass

test #3: `printBox("printBox-test1-data.txt", 7);`

file input: printBox-test1-data.txt:

```
roses ARE red
VIOLETS Are blUE

All my BASE
ARE belong To YOU
```

console output:

```
#######
#Roses#
#Viole#
#     #
#All m#
```

```
#Are b#
#######
```

result: pass

test #4: `printBox("printBox-test2-data.txt", 38);`

file input: printBox-test2-data.txt:

```
Haiku are easy
But sometimes they don't make sense
Refrigerator
```

console output:

```
######################################
#Haiku are easy                      #
#But sometimes they don't make sense #
#Refrigerator                        #
######################################
```

result: pass

test #5: `printBox("printBox-test3-data.txt", 3);`

file input: printBox-test3-data.txt:

```
yo
```

console output:

```
###
#Y#
###
```

result: pass

test #6: `printBox("bogus-data.txt", 20);`

console output:

result: pass

test #7: `printBox("printBox-test1-data.txt", -4);`

console output:

exp. exception: `int exception`

your exception: `int exception: -1`

result: pass

Need help?

Stuck on an exercise? Contact your TA or instructor.

If something seems wrong with our site, please contact us.