You are working on problem set:
Project 4 (Pause)

# leetSpeak

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

Write a function named `leetSpeak` that accepts two `string` parameters representing an input and output file respectively. Your function should convert the input file's text to "leet speak" (aka 1337 speak), an internet dialect where various letters are replaced by other letters/numbers, and output the "leet" version of the text to the given output file. Preserve the original line breaks from the input. Also wrap each word of input in parentheses. Perform the following replacements:

| Original character | 'Leet' character |
|---|---|
| o | Ø |
| l (lowercase L) | 1 |
| e | 3 |
| a | 4 |
| t | 7 |
| s (at the end of a word only) | Z |

For example, if the input file `lincoln.txt` contains the following text:

```
four score and
seven years ago our

fathers brought forth on this continent
a new nation
```

Then after a call of `leetSpeak("lincoln.txt", "leet.txt");`, the output file `leet.txt` should contain the following text:

```
(fØur) (scØr3) (4nd)
(s3v3n) (y34rZ) (4gØ) (Øur)

(f47h3rZ) (brØugh7) (fØr7h) (Øn) (7hiZ) (cØn7in3n7)
(4) (n3w) (n47iØn)
```

You may assume that each token from the input file is separated by exactly one space.

Hint: You may want to use the Stanford C++ library's `stringReplace` function, which is used as follows:

```
string str = "mississippi";
str = stringReplace(str, "s", "*");    // str = "mi**i**ippi"
```

.
.
.

```
void leetSpeak (string filename,string leet){

    ifstream input( filename );
    string line;
    string mostFreqUsedWord;
    string token;
    int highestwrdcount = 0;
    int currentwrdcount = 0;
    string currentword;
    bool firsttime = true;

    ofstream out_file{ leet };

    /*
        outfile copies full line
        istringstream copies the line to seperate by word
        subs will replace all the original letters with the leet version
        original copy of subs will be used to find index in line where it was found
        replace() will use index + temp string and replace the original word with the leet
        version and not change file
    */
    cout << endl;
    if ( input.is_open() ) {
        while ( getline(input, line) ) {
            string currentline = line; //changing line
            string temp = "";
            istringstream iss(line);

            // Iterate the istringstream
            // using do-while loop
            do {
                string subs;
                iss >> subs;
                string original = subs;
                //leet
                if(!subs.empty()){
                    if(subs[subs.length()-1] == 's'){
                        subs.pop_back();
                        subs += "z";
                    }
                    string temp = "(" + subs + ")";
                    temp = stringReplace(temp,"o" , "0");
                    temp = stringReplace(temp,"l" , "1");
                    temp = stringReplace(temp,"e" , "3");
                    temp = stringReplace(temp,"a" , "4");
                    temp = stringReplace(temp,"t" , "7");
                    //find
                    int index = 0;
                    int found = currentline.find(original);
                    //replace
                    currentline.replace(found,original.length(),temp);
                }
```

```
            } while (iss);

            out_file << currentline << "\n";


        }
        input.close();
    }
    else {
        cout << "Unable to open file." << endl;
    }
/*
    string temp = "(" + token + ")";
    temp = stringReplace(temp,"o" , "0");
    temp = stringReplace(temp,"l" , "1");
    temp = stringReplace(temp,"e" , "3");
    temp = stringReplace(temp,"a" , "4");
    temp = stringReplace(temp,"t" , "7");
    cout << temp << endl;
*/


}
```

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

Submit

**You passed 0 of 2 tests.** Try again.

**test #1:** `leetSpeak("leetSpeak-data-1.txt", "leetSpeak-output-1.txt");`

**file input:** leetSpeak-data-1.txt:

```
four score and
seven years ago our

fathers brought forth on this continent
a new nation
```

**exp. file output:**

```
output file "leetSpeak-output-1.txt":
(f0ur) (sc0r3) (4nd)
(s3v3n) (y34rZ) (4g0) (0ur)

(f47h3rZ) (br0ugh7) (f0r7h) (0n) (7hiZ) (c0n7in3n7)
(4) (n3w) (n47i0n)
```

**your file output:**

```
output file "leetSpeak-output-1.txt":
(f0ur) (sc0r3) (4nd)
(s3v3n) (y34rz) (4g0) (0ur)

(f47h3rz) (br0ugh7) (f0r7h) (0n) (7hiz) (c0n7in3n7)
(4) (n3w) (n47i0n)
```

**differences:**

```
3c3
< (s3v3n) (y34rZ) (4g0) (0ur)
> (s3v3n) (y34rz) (4g0) (0ur)
5c5
< (f47h3rZ) (br0ugh7) (f0r7h) (0n) (7hiZ) (c0n7in3n7)
> (f47h3rz) (br0ugh7) (f0r7h) (0n) (7hiz) (c0n7in3n7)
```

**result:** fail

**details:** incorrect file output

**test #2:** leetSpeak("leetSpeak-data-2.txt", "leetSpeak-output-2.txt");

**file input:** leetSpeak-data-2.txt:

```
Java refers to a number of computer software
products and specifications from Sun Microsystems,
a subsidiary of Oracle Corporation, that together
provide a system for developing application software
and deploying it in a cross-platform environment.

Java is used in a wide variety of computing platforms
from embedded devices and mobile phones on the low end,
to enterprise servers and supercomputers on the high end.

Java is used in mobile phones, Web servers and enterprise
applications, and while less common on desktop computers,
Java applets are often used to provide improved and secure
functionalities while browsing the World Wide Web.
```

**exp. file output:**

```
output file "leetSpeak-output-2.txt":
(J4v4) (r3f3rZ) (70) (4) (numb3r) (0f) (c0mpu73r) (s0f7w4r3)
(pr0duc7Z) (4nd) (sp3cific47i0nZ) (fr0m) (Sun) (Micr0sys73ms,)
(4) (subsidi4ry) (0f) (Or4c13) (C0rp0r47i0n,) (7h47) (70g37h3r)
(pr0vid3) (4) (sys73m) (f0r) (d3v310ping) (4pp1ic47i0n) (s0f7w4r3)
(4nd) (d3p10ying) (i7) (in) (4) (cr0ss-p147f0rm) (3nvir0nm3n7.)

(J4v4) (iZ) (us3d) (in) (4) (wid3) (v4ri37y) (0f) (c0mpu7ing) (p147f0
(fr0m) (3mb3dd3d) (d3vic3Z) (4nd) (m0bi13) (ph0n3Z) (0n) (7h3) (10w)
(70) (3n73rpris3) (s3rv3rZ) (4nd) (sup3rc0mpu73rZ) (0n) (7h3) (high)

(J4v4) (iZ) (us3d) (in) (m0bi13) (ph0n3s,) (W3b) (s3rv3rZ) (4nd) (3n7
(4pp1ic47i0ns,) (4nd) (whi13) (13sZ) (c0mm0n) (0n) (d3sk70p) (c0mpu73
(J4v4) (4pp137Z) (4r3) (0f73n) (us3d) (70) (pr0vid3) (impr0v3d) (4nd)
(func7i0n41i7i3Z) (whi13) (br0wsing) (7h3) (W0r1d) (Wid3) (W3b.)
```

**your file output:**

```
output file "leetSpeak-output-2.txt":
(J4v4) (r3f3rz) (70) (4) (numb3r) (0f) (c0mpu73r) (s0f7w4r3)
(pr0duc7z) (4nd) (sp3cific47i0nz) (fr0m) (Sun) (Micr0sys73ms,)
(4) (subsidi4ry) (0f) (Or4c13) (C0rp0r47i0n,) (7h47) (70g37h3r)
(pr0vid3) (4) (sys73m) (f0r) (d3v310ping) (4pp1ic47i0n) (s0f7w4r3)
(4nd) (d3p10y(in)g) (i7) in (4) (cr0ss-p147f0rm) (3nvir0nm3n7.)

(J4v4) (iz) (us3d) (in) (4) (wid3) (v4ri37y) (0f) (c0mpu7ing) (p147f0
(fr0m) (3mb3dd3d) (d3vic3z) (4nd) (m0bi13) (ph0n3z) (0n) (7h3) (10w)
(70) (3n73rpris3) (s3rv3rz) (4nd) (sup3rc0mpu73rz) (0n) (7h3) (high)

(J4v4) (iz) (us3d) (in) (m0bi13) (ph0n3s,) (W3b) (s3rv3rz) (4nd) (3n7
(4pp1ic47i0ns,) (4nd) (whi13) (13sz) (c0mm0n) (0n) (d3sk70p) (c0mpu73
(J4v4) (4pp137z) (4r3) (0f73n) (us3d) (70) (pr0vid3) (impr0v3d) (4nd)
(func7i0n41i7i3z) (whi13) (br0wsing) (7h3) (W0r1d) (Wid3) (W3b.)
```

**differences:**

```
2,3c2,3
< (J4v4) (r3f3rZ) (70) (4) (numb3r) (0f) (c0mpu73r) (s0f7w4r3)
< (pr0duc7Z) (4nd) (sp3cific47i0nZ) (fr0m) (Sun) (Micr0sys73ms,)
> (J4v4) (r3f3rz) (70) (4) (numb3r) (0f) (c0mpu73r) (s0f7w4r3)
> (pr0duc7z) (4nd) (sp3cific47i0nz) (fr0m) (Sun) (Micr0sys73ms,)
6c6
< (4nd) (d3p10ying) (i7) (in) (4) (cr0ss-p147f0rm) (3nvir0nm3n7.)
> (4nd) (d3p10y(in)g) (i7) in (4) (cr0ss-p147f0rm) (3nvir0nm3n7.)
8,10c8,10
< (J4v4) (iZ) (us3d) (in) (4) (wid3) (v4ri37y) (0f) (c0mpu7ing) (p147
< (fr0m) (3mb3dd3d) (d3vic3Z) (4nd) (m0bi13) (ph0n3Z) (0n) (7h3) (10w
< (70) (3n73rpris3) (s3rv3rZ) (4nd) (sup3rc0mpu73rZ) (0n) (7h3) (high
> (J4v4) (iz) (us3d) (in) (4) (wid3) (v4ri37y) (0f) (c0mpu7ing) (p147
> (fr0m) (3mb3dd3d) (d3vic3z) (4nd) (m0bi13) (ph0n3z) (0n) (7h3) (10w
> (70) (3n73rpris3) (s3rv3rz) (4nd) (sup3rc0mpu73rz) (0n) (7h3) (high
12,15c12,15
< (J4v4) (iZ) (us3d) (in) (m0bi13) (ph0n3s,) (W3b) (s3rv3rZ) (4nd) (3
< (4pp1ic47i0ns,) (4nd) (whi13) (13sZ) (c0mm0n) (0n) (d3sk70p) (c0mpu
```

```
< (J4v4)·(4pp137Z)·(4r3)·(0f73n)·(us3d)·(70)·(pr0vid3)·(impr0v3d)·(4n
< (func7i0n41i7i3Z)·(whi13)·(br0wsing)·(7h3)·(W0r1d)·(Wid3)·(W3b.)
> (J4v4)·(iz)·(us3d)·(in)·(m0bi13)·(ph0n3s,)·(W3b)·(s3rv3rz)·(4nd)·(3
> (4pp1ic47i0ns,)·(4nd)·(whi13)·(13sz)·(c0mm0n)·(0n)·(d3sk70p)·(c0mpu
> (J4v4)·(4pp137z)·(4r3)·(0f73n)·(us3d)·(70)·(pr0vid3)·(impr0v3d)·(4n
> (func7i0n41i7i3z)·(whi13)·(br0wsing)·(7h3)·(W0r1d)·(Wid3)·(W3b.)
```

**result:** fail
**details:** incorrect file output

Testing began at 2023/04/23 19:15 (PDT) and ran for 1207 ms.

**Need help?**

Stuck on an exercise? Contact your TA or instructor .

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