

```
Script started on 2022-09-24 09:17:55-05:00 [TERM="xterm" TTY="/dev/pts/8" COLUMNS=
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ pwd
/home/students/pu06439/CSC122/Portfolio I/Labs/"Hi" == "hi"
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ cat ncasemain.info
*****
* Upanshu Parekh                                CSC122-001 *
*                               Lab: "Hi" == "hi"?          *
*                               *                            *
* In this lab I designed a function that is a case-insensitive version *
* of the |strcmp()| function found in the <cstring> library. It works *
* almost the exact same way as the original function, with the same *
* return value rules as defined in the standard, again just accounting *
* for the case-insensitivity. *
*                               *                            *
* Project Base Level: 1.5 *
* No options attempted. *
*****
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ show-code ncasemain.cpp
```

ncasemain.cpp:

```
1 #include "strextra.h"
2 #include <iostream>
3 #include <string>
4
5 using namespace std;
6
7 int main()
8 {
9     string s1, s2;
10    cout << "\nWelcome! Please enter the first string: ";
11    getline(cin, s1);
12
13    cout << "\nPlease enter the second string: ";
14    getline(cin, s2);
15
16    cout << "\nAnd the result from case-insensitive comparison is: "
17          << strcmp_ncase(s1, s2) << "\n\n";
18    return 0;
19 }
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ show-code strextra.j
Unknown class/file type! Please have your teacher request a group update...
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ show-code strextra.h
```

strextra.h:

```
1 #pragma once
2
3 #include <string>
4
```

```
5 int strcmp_ncase(const std::string & s1, const std::string & s2);
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ show-code strextra.cpp
```

strextra.cpp:

```
1 #include "strextra.h"
2 #include <string>
3 #include <cctype>
4
5 using namespace std;
6 using sztp = std::string::size_type;
7
8 int strcmp_ncase(const string & s1, const string & s2)
9 {
10     sztp smallerLen{min(s1.length(), s2.length())};
11     int rValue{0};
12     bool done{false};
13
14     for (sztp i{0}; i <= smallerLen && !done; ++i)
15     {
16         if (tolower(s1[i]) != tolower(s2[i]))
17         {
18             done = true;
19             rValue = static_cast<int>(tolower(s1[i]) - tolower(s2[i]));
20         }
21     }
22     return rValue;
23 }
```

```
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ CPP ncasemain.cpp strextra.
cpp
ncasemain.cpp***
strextra.cpp...
```

```
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ ./ncasemain.out
```

```
Welcome! Please enter the first string: HELLO
```

```
Please enter the second string: hello
```

```
And the result from case-insensitive comparison is: 0
```

```
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ ./ncasemain.out
```

```
Welcome! Please enter the first string: HELLO
```

```
Please enter the second string: hallo
```

```
And the result from case-insensitive comparison is: 4
```

```
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ ./ncasemain.out
```

```
Welcome! Please enter the first string: HALLO

Please enter the second string: hello

And the result from case-insensitive comparison is: -4

pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ ./ncasemain.out

Welcome! Please enter the first string: HELLOAAAA

Please enter the second string: hello

And the result from case-insensitive comparison is: 97

pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ ./ncasemain.out

Welcome! Please enter the first string: HELLO

Please enter the second string: hellocccc

And the result from case-insensitive comparison is: -99

pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ cat ncasemain.tpq
*****
1. How do you compare two characters without reference to case?
How might you do this without destroying the character
variable(s) contents?

The way I would go about comparing two chars without reference to case
and without destroying the char variable's contents, is by using
the |tolower()| function from the <cctype> library and then using the
== comparison operator. tolower() returns the lowercase character but
importantly does not change that character that is put in. This will
avoid destroying the character variable's contents.

*****
2. How can you compare two strings in a case-insensitive way without
destroying their contents? (You should not change the strings in order
to compare them!)

To compare two strings, I would just iterate through the string up to
the size of the string with fewer characters and then compare the two
string's characters using the method above as it is iterating.
If function encounters a character from s1 that is greater
than one from s2, it will automatically flag the s1 string as s1 > s2
and return a + value. Vice versa for - value. If it the characters are
equal to each other, they will keep going until the end of the string
or until one of the triggers for + or - activates.

If there is no differing character til the end of the shorter string,
then the longer string will be considered "greater".

*****
3. What kind of arguments should your string comparison function take?
```

```
(Value, reference, constant?)

The comparison function should take 2 string arguments that are constant
and passed by reference, as we are not going to be touching their data.

*****
4. How do you get that weird return value for your function?
Is it always -1, 0, 1? Or is there a reason it was defined as simply
less than 0, 0, or greater than 0?

To get that weird return value, I subtract the char from the 2nd string
from the char from the 1st string: s1[0] - s2[0], and then cast it to
int. It still follows the sign rules laid out, but it takes into account
the exact ASCII equivalent difference between the character.

This number is not usually relevant, as most of the time we're only
interested in the comparison result itself, not the exact ASCII
equivalent differences.

*****
5. How many times will you need to call your function to test it
thoroughly? How many times should you have to run the driver to do this
testing?

In my driver program file, I'll only need to compare two strings once,
but I'll need to run the driver 5 times to get a full test of the
function, ensuring to test the case-insensitivity.
Once for the -, 0, and + cases with a string of the same length but
differing cases on the characters, and then 2 more for case-insensitive
comparisons of strings with different lengths, ensuring a - and = result.
pu06439@ares:~/CSC122/Portfolio I/Labs/"Hi" == "hi"$ exit
exit

Script done on 2022-09-24 09:22:50-05:00 [COMMAND_EXIT_CODE="0"]
```