```
Script started on 2022-09-24 09:40:54-05:00 [TERM="xterm" TTY="/dev/pts/8" COLUMNS=
pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ pwd
/home/students/pu06439/CSC122/Portfolio I/Labs/Now where'd I put that
pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ cat findmain.info
o
*****************************************************************************
* Upanshu Parekh                                              CSC122-001 *
*                 Lab: Now where'd I put that                               *
*                                                                           *
* This program shows the use of find functions which take in a base     *
* string to be searched, along with a char/string to look for. It will  *
* search the base for it, and then return that int index where it was   *
* found, or std::string::npos if it was not found.                      *
*                                                                           *
* Base Level: 2                                                             *
* Options Chosen:                                                           *
*   Level 1: Case-sensitivit argument.                                      *
*   Level 2: User can use wildcards to represent 0+ chars, and ?        *
*            to represent 1 char. Escape sequence char '/' added.       *
*   Level 3: User can use multiple wildcards in one input string.       *
* Total Level: 2+1+2+3 = 8                                                  *
*****************************************************************************

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ show-code findmai
n.cpp


findmain.cpp:
```

```cpp
// Driver program for strextra library
#include "strextra.h"
#include <iostream>
#include <string>
#include <limits>

using namespace std;

int main() {
    enum option{CHAR=1, STRING};
    string baseString{""},
                inputString{""};
    int selection{0};
    char inputChar{'a'};
    string::size_type findResult{0};

    cout << "\nWelcome! Please enter a base string to search."
                "\nString: ";
    getline(cin, baseString);

    cout << "\nString accepted! What would you like"
                " to search for in this string?"
                "\n [1] CHAR"
                "\n [2] STRING"
                "\nYour choice: ";
    cin >> selection;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    switch (selection)
    {
        case CHAR:
            cout << "\nYou selected: CHAR."
                        "\nPlease enter the CHAR to search for (type `"
                        " for space character): ";
                        // the ` substituting space is for
                        // testing purposes only
            cin >> inputChar;
            inputChar = (inputChar == '`') ? ' ' : inputChar;

            //default arg: caseSensitive = true
            findResult = find(baseString, inputChar);
            cout << "\nResult from |find| function was: "
                     << findResult << ".";

            findResult = find(baseString, inputChar, false);
            cout << "\nResult from case-insensitive "
                        "|find| function was: " << findResult << ".";

            break;

        case STRING:
            cout << "\nYou selected: STRING."
                        "\nPlease enter the STRING to search for: ";
            getline(cin, inputString);

            //default arg: caseSensitive = true
            findResult = find(baseString, inputString);
            cout << "\nResult from |find| function was: "
                     << findResult << ".";

            findResult = find(baseString, inputString, false);
            cout << "\nResult from case-insensitive "
                        "|find| function was: " << findResult << ".";

            break;

        default:
            cout << "\nInvalid input!";
            break;
    }
    cout << "\nEnd of program. Thank you for using.\n\n";
    return 0;
}
```

```
pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ show-code strextr
a.h


strextra.h:
```

```
 1  // Header file for find function
 2  #pragma once
 3  #include <string>
 4
 5  //This variant of the find function will work with |char| inputs
 6  std::string::size_type
 7  find(const std::string & base, const char & input,
 8       const bool & caseSensitive = true);
 9
10  //This variant of the find function will work with |string| inputs
11  std::string::size_type
12  find(const std::string & base, const std::string & input,
13       const bool & caseSensitive = true);
```

```
pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ show-code strextr
a.cpp
```

strextra.cpp:

```
 1  // implementation file for the find functions
 2  #include "strextra.h"
 3  #include <string>
 4  #include <cctype>
 5
 6  using namespace std;
 7  using sz_tp = string::size_type;
 8
 9  sz_tp find(const string & base, const char & input,
10             const bool & caseSensitive)
11  {
12      sz_tp baseLength{base.length()},
13            foundIndex{string::npos};
14      bool found{false};
15
16      for (sz_tp i = 0; i < baseLength && !found; i++)
17      {
18          if (  (caseSensitive && base[i] == input)
19             || (!caseSensitive &&
20                 tolower(base[i]) == tolower(input))  )
21          {
22              found = true;
23              foundIndex = i;
24          }
25      }
26      return foundIndex;
27      // npos is esentially the unsigned version of -1 or invalid index
28  }
29
30  sz_tp find(const string & base, const string & input,
31             const bool & caseSensitive)
32  {
33      sz_tp baseLength{base.length()},
34            inputLength{input.length()},
35            substringIndex{0}, foundIndex{string::npos};
36      bool done{false},found{false};
37      string subBase{""};
38
39      for (sz_tp i = 0; i < baseLength && !found; ++i)
40      {
41          substringIndex = i;
42          done = false;
43
44          for (sz_tp k = 0; k < inputLength && !done; ++k)
45          {
46              // escape sequence detection
47              if (input[k] != '/')
48              {
49                  if (input[k] == '*')
50                  {
51                      if (k == inputLength - 1)
52                      {
53                          done = true;
54                          found = true;
55                          foundIndex = i;
56                      }
57                      // if * at the end of input string
58                      // who cares what the rest of the base string is?
59
60                      // if char after * is ?
61                      // then it's 0 chars until next valid char
62                      while ( (
63                              (caseSensitive
64                                  && input[k+1] != base[substringIndex])
65                          || (!caseSensitive
66                              && tolower(input[k+1])
67                                 != tolower(base[substringIndex])) )
68                          && input[k+1] != '?' && !done)
69                      {
70                          if (substringIndex == baseLength - 1)
71                          { // substringIndex is at end of base
72                              done = true;
73                              // found = false;
74                          }
75
76                          ++substringIndex;
77                          // keep iterating until it does find next
78                          // char after wildcard in base string
79                      }
80                  }
81                  else if ( ( (caseSensitive
82                                  && input[k] == base[substringIndex])
83                      || (!caseSensitive
84                          && tolower(input[k])
85                             == tolower(base[substringIndex])) )
86                      || input[k] == '?')
```

```
    87                  {
    88                      // if valid match and at end of string, it is found!
    89                      if (k == inputLength - 1)
    90                      {
    91                          done = true;
    92                          found = true;
    93                          foundIndex = i;
    94                      }
    95
    96                      // if any of these matches are valid, go to next base
    97                      // string char
    98                      ++substringIndex;
    99                  }
   100                  else
   101                  { // if no valid matches, stop
   102                      done = true;
   103                      // found = false;
   104                  }
   105              }
   106              else {
   107                  if (k == inputLength - 1)
   108                  {  // if escape char at end of inputString,that's invalid
   109                      done = true;
   110                      // found = false;
   111                  }
   112
   113                  ++k; // skip to char after '/' escape sequence char
   114                  if (input[k] == base[substringIndex])
   115                  { // valid match
   116                      if (k == inputLength - 1)
   117                      {
   118                          done = true;
   119                          found = true;
   120                          foundIndex = i;
   121                      }
   122                      ++substringIndex;
   123                  }
   124                  else
   125                  { // if next char does not match
   126                      done = true;
   127                      // found = false;
   128                  }
   129              }
   130          }
   131      }
   132      return foundIndex;
   133  }
pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ CPP findmain.cpp
strextra.cpp
findmain.cpp***
strextra.cpp...


pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out
```

```
Welcome! Please enter a base string to search.
String: pits

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 1

You selected: CHAR.
Please enter the CHAR to search for (type ` for space character): I

Result from |find| function was: 18446744073709551615.
Result from case-insensitive |find| function was: 1.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: dumb bunnies

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 1

You selected: CHAR.
Please enter the CHAR to search for (type ` for space character): `

Result from |find| function was: 4.
Result from case-insensitive |find| function was: 4.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: dumb bunnies

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: b bunn

Result from |find| function was: 3.
Result from case-insensitive |find| function was: 3.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: dumb bunnies
```

```
String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: u??Ie

Result from |find| function was: 18446744073709551615.
Result from case-insensitive |find| function was: 6.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: dumb bunnies

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: ??zz

Result from |find| function was: 18446744073709551615.
Result from case-insensitive |find| function was: 18446744073709551615.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: The quick brown fox

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: *T

Result from |find| function was: 0.
Result from case-insensitive |find| function was: 0.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: The quick brown fox

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: *N

Result from |find| function was: 18446744073709551615.
Result from case-insensitive |find| function was: 0.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: The quick brown fox

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: u*n

Result from |find| function was: 5.
Result from case-insensitive |find| function was: 5.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: The quick brown fox

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: u*z

Result from |find| function was: 18446744073709551615.
Result from case-insensitive |find| function was: 18446744073709551615.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: The quick brown fox

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2
```

```
You selected: STRING.
Please enter the STRING to search for: x*

Result from |find| function was: 18.
Result from case-insensitive |find| function was: 18.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: The quick brown fox

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: h*i*b*n

Result from |find| function was: 1.
Result from case-insensitive |find| function was: 1.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: The quick brown fox

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: h*z*b*n

Result from |find| function was: 18446744073709551615.
Result from case-insensitive |find| function was: 18446744073709551615.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: The quick brown fox

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: h*i?k

Result from |find| function was: 1.
Result from case-insensitive |find| function was: 1.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: The quick brown fox

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: h*i??k

Result from |find| function was: 18446744073709551615.
Result from case-insensitive |find| function was: 18446744073709551615.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: Is the fox quick? Not sure* Who knows?

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: quick/? n

Result from |find| function was: 18446744073709551615.
Result from case-insensitive |find| function was: 11.
End of program. Thank you for using.

pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ ./findmain.out

Welcome! Please enter a base string to search.
String: Is the fox quick? Not sure* Who knows?

String accepted! What would you like to search for in this string?
 [1] CHAR
 [2] STRING
Your choice: 2

You selected: STRING.
Please enter the STRING to search for: sure/* W

Result from |find| function was: 22.
Result from case-insensitive |find| function was: 22.
End of program. Thank you for using.
```

```
pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ cat findmain.tpq
**************************************************************************

1. What arguments does each find function take? Are they changed?
What special care should you take with them?

Each find function must take in a base string that will be searched.
The second argument must be the phrase to find within that string.
One must take in a char, while the other must take in a string.
The special care I should take is that I need to make one search function
that works with the char variant, and another different, but related
search function that works with the string variant.

**************************************************************************

2. What value is returned by your functions? What type is it and what
does it represent?

The functions will return an |int| index position of the base string
where the char or string phrase occurred. If it failed to find an
occurence of the char/string, it will return an |int| value of -1.

**************************************************************************

3. What care does a caller of your functions have to take with
this return value?
(i.e. Can they immediately assume it is a valid index?)

The caller of my function has to know that it will not return
a valid index 100% of the time. If the char/string could not be found,
it will return -1. If that caller assumes a valid index
was going to be printed, they would run into nasty IndexOutOfBounds
exceptions, or worse.

**************************************************************************

4. How does the compiler distinguish which of your functions is
being used for a particular call?
(They have the same name, after all...)

It wil distinguish my functions from each other by checking their
signatures. In this case, the return types will be the same, as will the
type of the first paramater (the string to
be searched), but the second parameter will be either a |char| or
|string|. Whichever one is included in the function call will run either
the char-corresponding |find| function or the string-corresponding |find|
function.

**************************************************************************

5. How do you protect your library from being circularly included?

I can use definitions as such:

#ifndef STREXTRA_H_INCLUDE
#define STREXTRA_H_INCLUDE
// ... header file interface in between ...
#endif

Alternatively, I can just type:
#pragma once
This is what I will use.

**************************************************************************

6. What changes are needed in your main application
(the test application here) to get it to work with the library? What
about the compiling process?

I need to ensure that the library header file is included with the
directive:
#include "strextra.h"
That's it for what the main application needs in terms of code-wise.

When it is being compiled, I need to ensure that I add the corresponding
strextra.cpp file which contains the definitions of the functions in the
header file.

**************************************************************************

7. How many files does your library consist of? What are they?
Which one(s) do you #include?

My library consists of 2 files: the interface file |strextra.h| and the
implementation/definition file |strextra.cpp|.
In the |findmain.cpp| main file, I include the header file
via #include "strextra.h"


**************************************************************************
pu06439@ares:~/CSC122/Portfolio I/Labs/Now where'd I put that$ exit
exit

Script done on 2022-09-24 09:50:36-05:00 [COMMAND_EXIT_CODE="0"]
```