```
Script started on Fri 21 Sep 2012 10:05:54 PM CDT
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ p
wd
/home/georgia/cplusplus
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ g
++ --version
g++ (Ubuntu/Linaro 4.6.1-9ubuntu3) 4.6.1
Copyright (C) 2011 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at strextra.info
********************************************************************************

Name: Jakob Hansen
Class: CSC122
Lab: Strextra
Level: 2
    +0.5 for optional start-position-specification
    +2 for single wildcards + escape sequence
    +3 for multi-'*' capability
    Total attempted: 7.5

Program description:
    A program that makes use of a hand "find" function for c-strings not found
    in the ol' standard library. Use it to search for a single char or a
    substring. You can even tell it where to start looking...and if you're not
    sure quite WHAT you're looking for, use the '?' character to indicate a
    single "wildcard" and the '*' character to indicate a "wildcard"
    representing anywhere from 0 to a whole lot of characters! Don't worry, if
    you are looking to find an actual one of those special '*' or '?' chars,
    just escape them in your search string with '/'.

********************************************************************************
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at strextra.h
#ifndef STREXTRA_H_INC
#define STREXTRA_H_INC

#include <cstring>
#include <iostream>
#include <climits>

using namespace std;

// char version
short find(const char str[], char c, short index = 0);

// string version
short find(const char str[], char substr[], short index = 0);

#endif
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at strextra.cpp
#include "strextra.h"
#include <cstring>
#include <iostream>
#include <climits>

using namespace std;

// char
short find(const char str[], char c, short index)
{
    while (str[index] != c && str[index] != '\0')
    {
        index++;
    }
    if (str[index] == '\0')
    {
        index = -1;
    }
    return index;
}

// string
short find(const char str[], char substr[], short index)
{
    short matchpos, subindx = 0, wildcount = 0;
    bool beginmatch, endmatch;

    while (substr[subindx] != '\0' && str[index] != '\0' && matchpos != -1)
    {
        // found escape character
        if (substr[subindx] == '/' && substr[subindx-1] != '/')
        {
            subindx++;
        }
        // if the first char in the search string is * then beginmatch is true
        else if (!beginmatch && subindx == 0 && substr[subindx] == '*')
        {
            beginmatch = true;
            matchpos = 0;
        }
        // found unescaped '*' that's not the last char in search string
        else if (substr[subindx] == '*' && substr[subindx-1] != '/'
                                        && substr[subindx + 1] != '\0')
        {
            beginmatch = true;
            wildcount = index;
            // pause at '*' and look forward for next match
            while (str[wildcount] != substr[subindx+1] &&
                                        str[wildcount] != '\0')
            {
                wildcount++;
            }
            // found next match
            if (str[wildcount] != '\0')
            {
                index = wildcount;
                subindx++;
            }
            // didn't find it - abort
            else
            {
                matchpos = -1;
            }
        }
        // found an unescaped '?' that's not the last char
        else if (substr[subindx] == '?' && substr[subindx - 1] != '/'
                                        && substr[subindx + 1] != '\0')
        {
               index++;
               subindx++;
        }
        // found a match, throw begin marker and advance to check next pair
        else if (!beginmatch && substr[subindx] == str[index])
        {
            matchpos = index; //store the match
            beginmatch = true;
            index++;
            subindx++;
        }
        // in the middle of potential matching string, keep checking
        else if (beginmatch && substr[subindx] == str[index]
                            && substr[subindx + 1] != '\0')
        {
```

```
            index++;
            subindx++;
        }
        // check if the last character matches too
        else if (beginmatch && substr[subindx] == str[index]
                            && substr[subindx + 1] == '\0')
        {
            endmatch = true;
            index++;
            subindx++;
        }
        else if (substr[subindx] == '*' && substr[subindx + 1] == '\0')
        {
            endmatch = true;
            subindx++;
        }
        else if (beginmatch && substr[subindx + 1] == '\0'
                            && substr[subindx] == '?')
        {
            endmatch = true;
            index++;
            subindx++;
        }
        /* didn't find a match, keep going in "searched" string, start over in
        substring */
        else
        {
            index++;
            subindx = 0;
            beginmatch = false;
        }
    }
    if (!beginmatch || !endmatch)
    {
        matchpos = -1;
    }

    return matchpos;
}
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at findextra.cpp
#include <cstring>
#include <iostream>
#include <climits>
#include "strextra.h"

const short ARR_MAX = 50;

void get_line(char s[], const long max);

int main(void)
{
    char searchstr[ARR_MAX], subsearch[ARR_MAX];
    char c, answer;
    short pos = 0, start;
    do {
        cout << "\n\tWelcome to the find() driver.\nPlease Enter a string: ";

        get_line(searchstr, ARR_MAX);

        cout << "\nPlease enter the char you'd like to find: ";

        cin >> c;

        cout << "\nWould you like to specify a starting position? ";
        cin >> answer;
        cin.ignore(INT_MAX, '\n');
        if (toupper(answer) == 'Y')
        {
            cout << "\nOk, enter the starting position: ";
            cin >> start;
            pos = find(searchstr, c, start);
        }
        else
        {
            pos = find(searchstr, c);
        }
        if (pos == -1)
        {
            cout << "\nCharacter not found!";
        }
        else
        {
            cout << "\nCharacter found at position: " << pos;
        }

        cout << "\n\nPlease Enter a substring you'd like to find: ";

        get_line(subsearch, ARR_MAX);

        cout << "\nWould you like to specify a starting position? ";
        cin >> answer;
        cin.ignore(INT_MAX, '\n');
        if (toupper(answer) == 'Y')
        {
            cout << "\nOk, enter the starting position: ";
            cin >> start;
            pos = find(searchstr, subsearch, start);
        }
        else
        {
            pos = find(searchstr, subsearch);
        }
        if (pos == -1)
        {
            cout << "\nSubstring not found!";
        }
        else
        {
            cout << "\nSubstring found starting at position: " << pos << "\n";
        }

        cout << "\nWould you like to try again? ";
        cin >> answer;
        cin.ignore(INT_MAX, '\n');

    } while (toupper(answer) == 'Y');

    cout << "\nGoodbye.\n";
    return 0;
}

void get_line(char s[], const long max)
{
    cout.flush();
    if (cin.peek() == '\n')
    {
        cin.ignore();
    }
    cin.getline(s, max);
    if (cin.fail())
    {
        cin.clear();
        cin.ignore(INT_MAX, '\n');
    }
    return;
}
```

```
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ C
PP findextr\033[K033[Ktra strextra
findextra.cpp***
strextra.cpp...


\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ .
/findextra.out

        Welcome to the find() driver.
Please Enter a string: Here's a strident straight tricky string!

Please enter the char you'd like to find: s

Would you like to specify a starting position? y

Ok, enter the starting position: 6

Character found at position: 9

Please Enter a substring you'd like to find: string

Would you like to specify a starting position? n

Substring found starting at position: 34

Would you like to try again? y

        Welcome to the find() driver.
Please Enter a string: Here's a *crazy* kind of string!

Please enter the char you'd like to find: *

Would you like to specify a starting position? n

Character found at position: 9

Please Enter a substring you'd like to find: /*cdy/*

Would you like to specify a starting position? n

Substring found starting at position: 9

Would you like to try again? y

        Welcome to the find() driver.
Please Enter a string: Hs thisai string? Wit with a couple of ?s in it?

Please enter the char you'd like to find: ?

Would you like to specify a starting position? n

Character found at position: 16

Please Enter a substring you'd like to find: st?ing/?

Would you like to specify a starting position? n

Substring found starting at position: 10

Would you like to try again? y

        Welcome to the find() driver.
Please Enter a string: I should show an example of finding the!

Please enter the char you'd like to find: z

Would you like to specify a starting position? n

Character not found!

Please Enter a substring you'd like to find: nothing

Would you like to specify a starting position? n

Substring not found!
Would you like to try again? y

        Welcome to the find() driver.
Please Enter a string: One last showoff off the multiple *s!

Please enter the char you'd like to find: f

Would you like to specify a starting position? n

Character found at position: 14

Please Enter a substring you'd like to find: l*t*w*t*m

Would you like to specify a starting position? n

Substring found starting at position: 4

Would you like to try again? n

Goodbye.
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at strextra.tpq
********************************************************************************


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

    The char version of the function takes two (or three) arguments: the string
    to be searched, the character we are searching for, and the position at
    which to begin the search (which has a default value of 0). The substring
    version of the function takes the same arguments, with the exception of the
    second one, being an array of chars instead a single one. The first two
    arguments to each function need not and actually should not be changed - so
    it's a good idea to make them const!


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

    The functions return short integers. I suppose that if we were dealing with
    huge, huge amounts of chars the return type might need to be a bigger
    integer, but for my purposes short works fine. The integer that's returned
    represents either the index of the found char (or first char of the
    substring) or -1, a categorically invalid index (meaning, to quote
    Spaceballs, "We ain't found shit!")

3. What care does a caller of your functions have to take with this return
value?
    The caller of the function would do well not to assume that its a valid
    index, since the whole point is to find out whether there IS a valid index.
    So the value returned by the function has to be worked into a conditional
    structure in the calling code.

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

    They have different parameters - one a char one an array of chars.

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

By using the #ifndef #define #endif preprocessor directives with some unique identifier.

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?

You've got to #include "strextra.h", and you've got to pass the name of your main app along with the name of the lib (implementation file) to the compiler come compile-time.

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

There's the header file and then the implementation file (which #includes the header), but you need only include the header file in your main file.

****

\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ e
xit
exit

Script done on Fri 21 Sep 2012 10:12:27 PM CDT