```
Script started on 2023-07-14 02:49:57-05:00 [TERM="xterm" TTY="/dev/pts/0" COLUMNS=
bj94684@ares:~$ pwd
/home/students/bj94684
bj94684@ares:~$ cat ns.info
/*****************************************************************************
 *                                                                           *
 *  NAME:  Jose Barron                          CLASS:  CSC122-002           *
 *                                                                           *
 *         Lab: Hide 'n' Go Seek                Level:   1.5                 *
 *       Option: Loop                           Level: + 1.5                 *
 *       Option: Substring                      Level: + 1.0                 *
 *       Option: Print all matches              Level: + 0.5                 *
 *       Option: Name position print nicely     Level: + 2.5                 *
 *                                         Total Level:   7.0                *
 *                                                                           *
 * This program is designed to tell the user the positon of a name in a      *
 * list provided by the user. If the name is found, it returns the           *
 * position of it, and if not found then it returns not found. If more       *
 * then one name matches what the user wants then it will print all names    *
 * that match and their location in the list.                                *
*****************************************************************************/
bj94684@ares:~$ show-code strextra.h


strextra.h:
```

```
 1  #ifndef STREXTRA_H_INC
 2  #define STREXTRA_H_INC
 3
 4  #include<iostream>
 5  #include<string>
 6  #include<vector>
 7  #include<cctype>
 8
 9  using namespace std;
10
11
12  inline std::string tolower_str(std::string s)
13  {
14      std::string t = s;
15      for(std::string::size_type pos = 0; pos != s.length(); ++pos)
16      {
17          t[pos] = static_cast<char>( tolower(t[pos]) );
18      }
19      return t;
20
21  }
22
23  inline std::vector<std::string::size_type> find(std::string s, char t,
24                                          bool case_sensitive = true)
25  {
26      if( ! case_sensitive)
27      {
28          s = tolower_str(s);
29          t = static_cast<char>( tolower(t) );
30      }
31
32      std::vector<string::size_type> num;
33      for (std::string::size_type pos = 0; pos != s.length(); ++pos)
34      {
35          if (static_cast<char>( s[pos] ) == t )
36          {
37              num.push_back(pos);
38          }
39      }
40      return num;
41  }
42
43  inline bool str_is_incl(std::string s, std::string t, std::string::size_ty
44                                                  std::string::size_type end)
45  {
46      std::string nr = s.substr(beg, end-beg+1);
47      bool tf;
48      if ( nr == t)
49      {
50          return tf = true;
51      }
52      return tf = false;
53  }
54
55  inline std::string::size_type strcom(std::vector<string::size_type> b,
56                                       std::vector<string::size_type> e,
57                                       std::string s, std::string t)
58  {
59      string::size_type num;
60      for(auto p : b)
61      {
62          for(auto p1 : e)
63          {
64              if ( p > p1)    // The starter position can never be greater
65              {               // than the end position
66                  num;
67              }
68              else
69              {
70                  bool ys = str_is_incl(s, t, p, p1);
71                  if ( ys)
72                  {
73                      return num = p;
74                  }
75                  num;
76              }
77          }
78      }
79      return num = s.length();
80  }
81
```

```
    82  std::string::size_type find(std::string s, std::string t, bool c_s = true);
    83
    84
    85
    86  #endif
bj94684@ares:~$ show-code strextra.cpp


strextra.cpp:


     1  #include<iostream>
     2  #include<string>
     3  #include<vector>
     4  #include<cctype>
     5  #include"strextra.h"
     6
     7  using namespace std;
     8
     9  string::size_type find(string s, string t, bool c_s)
    10  {
    11      if( ! c_s)
    12      {
    13          s = tolower_str(s);
    14          t = tolower_str(t);
    15      }
    16      string::size_type loc;
    17      vector<string::size_type> pos = find(s, t[0], c_s);
    18      vector<string::size_type> pos_b = find(s, t[ t.length() - 1], c_s);
    19      if ( ! pos.empty() && ! pos_b.empty())
    20      {
    21          string::size_type ys = strcom(pos, pos_b, s, t);
    22          if ( ys != s.length() )
    23          {
    24              return loc = ys;
    25          }
    26          else
    27          {
    28              return loc = s.length();
    29          }
    30      }
    31      else
    32      {
    33          return loc = s.length();
    34      }
    35  }
bj94684@ares:~$ show-code suffix.h


suffix.h:


     1  #ifndef SUFFIX_H_INC
     2  #define SUFFIX_H_INC
```

```
     3
     4  #include<iostream>
     5  #include<string>
     6  #include<ctime>
     7  #include <cstdlib>
     8
     9
    10  inline short rand_num(short min, short max)
    11  {
    12      return static_cast<short>( rand()%(max-min+1) + min );
    13  }
    14  inline int get_lastdigit(int n)
    15  {
    16      int last_digit = n % 10;
    17      return last_digit;
    18  }
    19  inline int get_sec_lastdigit(int n)
    20  {
    21      int s_last_digit = (n / 10) % 10;
    22      return s_last_digit;
    23  }
    24  std::string get_suffix (int n);
    25
    26
    27
    28  #endif
bj94684@ares:~$ show-code suffix.cpp


suffix.cpp:


     1  #include<iostream>
     2  #include<string>
     3  #include<ctime>
     4  #include <cstdlib>
     5  #include"suffix.h"
     6
     7
     8  using namespace std;
     9
    10  string get_suffix (int n)
    11  {
    12      string suffix;
    13      int ld = get_lastdigit(n);
    14      int s_ld = get_sec_lastdigit(n);
    15      if (ld == 0)
    16      {
    17          return suffix = "th";
    18      }
    19      else if ( ld == 1)
    20      {
    21          if (s_ld == 1)
    22          {
```

```
    23              return suffix = "th";
    24          }
    25          return suffix = "st";
    26      }
    27      else if ( ld == 2)
    28      {
    29          if (s_ld == 1)
    30          {
    31              return suffix = "th";
    32          }
    33          return suffix = "nd";
    34      }
    35      else if ( ld == 3)
    36      {
    37          if (s_ld == 1)
    38          {
    39              return suffix = "th";
    40          }
    41          return suffix = "rd";
    42      }
    43      else
    44      {
    45          return suffix = "th";
    46      }
    47
    48  }
bj94684@ares:~$ show-code ns.cpp


ns.cpp:


     1  #include <iostream>
     2  #include <fstream>
     3  #include <string>
     4  #include <vector>
     5  #include <limits>
     6  #include "strextra.h"
     7  #include "suffix.h"
     8
     9
    10  using namespace std;
    11
    12  int main()
    13  {
    14      cout << "\n\t\tWelcome to the Name Searching Program\n\n";
    15      ifstream file;
    16      string fn;
    17      cout << "\nPlease enter the name of your names files:  ";
    18      getline(cin, fn);
    19      file.open(fn);
    20      while ( ! file )
    21      {
    22          file.close();
    23          file.clear();
    24          cout << "\nIm sorry I could open " << '"' << fn << '"'
    25               << ". Please enter another name:  ";
    26          getline(cin, fn);
    27          file.open(fn);
    28      }
    29      cout << "\nFile " << '"' << fn << '"' << " was opened succesfully\n";
    30      string s;
    31      vector<string> names;
    32      while (!file.eof() )
    33      {
    34          getline(file, s);
    35          names.push_back(s);
    36      }
    37      file.close();
    38      file.clear();
    39      /*
    40      for (vector<string>::size_type pos = 0; pos != names.size(); ++pos)
    41      {
    42          cout << names[pos] << '\n';
    43      }
    44      */
    45      bool ch;
    46      bool done;
    47      do{
    48      string f;
    49      cout <<"\nWhat name would you like to find:  ";
    50      getline(cin,f);
    51      cout << '\n';
    52      short indicator = 0;
    53      for (vector<string>::size_type pos = 0; pos != names.size(); ++pos)
    54      {
    55          if ( find(names[pos], f,false) != (names[pos]).length() )
    56          {
    57
    58              string suffix = get_suffix(pos+1);
    59              cout << '"' << names[pos] << '"' << " was the " << pos + 1<< su
    60                   << " name of the list\n";
    61              ++indicator;
    62          }
    63      }
    64      if (indicator == 0)
    65      {
    66          cout <<'"' << f << '"' << " was not found in the list\n";
    67      }
    68
    69      do{
    70      char choice;
    71      cout << "\nWould you like to search again?  ";
    72      cin >> choice;
    73      choice = static_cast<char>( toupper( choice ) );
    74      cin.ignore(numeric_limits<streamsize>::max(), '\n');
    75      if ( choice == 'Y')
    76      {
```

```
    77          done = false;
    78          ch = true;
    79      }
    80      else if ( choice == 'N')
    81      {
    82          done = true;
    83          ch = true;
    84      }
    85      else
    86      {
    87          cout << "\nYou did not input any correct answer."
    88               << " Please Try Again\n";
    89          ch = false;
    90      }
    91      }while ( ! ch);
    92
    93      } while ( ! done);
    94      cout << "\nThank you for using NSP\n\n";
    95
    96      return 0;
    97  }
bj94684@ares:~$ CPP strextra suffix ns
ns.cpp***
strextra.cpp...
suffix.cpp...
In file included from ns.cpp:6:
strextra.h: In function
‘std::__cxx11::basic_string<char>::size_type
strcom(std::vector<long unsigned int>, std::vector<long unsigned int>,
std::string, std::string)’:
strextra.h:66:17: warning: statement
has no effect [-Wunused-value]
   66 |                 num;
      |                 ^~~
strextra.h:75:17: warning: statement
has no effect [-Wunused-value]
   75 |                 num;
      |                 ^~~
ns.cpp: In function ‘int main()’:
ns.cpp:58:43: warning: conversion
from ‘std::vector<std::__cxx11::basic_string<char>
>::size_type’ {aka ‘long unsigned
int’} to ‘int’ may change value
[-Wconversion]
   58 |             string suffix = get_suffix(pos+1);
      |                                        ~~~^~
In file included from strextra.cpp:5:
strextra.h: In function
‘std::__cxx11::basic_string<char>::size_type
strcom(std::vector<long unsigned int>, std::vector<long unsigned int>,
std::string, std::string)’:
strextra.h:66:17: warning: statement
has no effect [-Wunused-value]
   66 |                 num;
      |                 ^~~
strextra.h:75:17: warning: statement
has no effect [-Wunused-value]
   75 |                 num;
      |                 ^~~


bj94684@ares:~$ ./ns.out

                Welcome to the Name Searching Program


Please enter the name of your names files:  names

File "names" was opened succesfully

What name would you like to find:  veroNica

"Veronica Stoltzfus" was the 1st name of the list

Would you like to search again?  Joe

You did not input any correct answer. Please Try Again

Would you like to search again?  yes

What name would you like to find:  Joe

"Joe Early" was the 30th name of the list

Would you like to search again?  no

Thank you for using NSP

bj94684@ares:~$ ./ns.out

                Welcome to the Name Searching Program


Please enter the name of your names files:  names

File "names" was opened succesfully

What name would you like to find:  muniz

"Mandy Muniz" was the 10th name of the list
"Leslie Muniz" was the 11th name of the list

Would you like to search again?  sure

You did not input any correct answer. Please Try Again

Would you like to search again?  yes
```

```
What name would you like to find:  Barney

"Myranda Barney" was the 13th name of the list

Would you like to search again?  yes

What name would you like to find:  alex cobb

"Alex Cobb" was the 23rd name of the list

Would you like to search again?  yes

What name would you like to find:  jose

"jose" was not found in the list

Would you like to search again?  no

Thank you for using NSP

bj94684@ares:~$ ./ns.out

                Welcome to the Name Searching Program


Please enter the name of your names files:  names

File "names" was opened succesfully

What name would you like to find:  rayne

"Rayne Phan" was the 21st name of the list

Would you like to search again?  no

Thank you for using NSP

bj94684@ares:~$ cat ns.tpq
/*****************************************************************************
*                                                                            *
* TPQs:                                                                      *
* 1. You handle not knowing by making all adding all the names to a          *
* vector or a container, then you can know how much names are on the list    *
* 2. I used a while loop with a getline to process the file.                 *
* 3. If the person is not found then I output a message saying that no       *
* name was found with what the user input to search.                         *
* 4. My program assumes that there isnt multiple names per line. Since,      *
* I use getline, then if there were more than one then my program would      *
* think that all those names in one line is actually one name.               *
* Additional TPQs:                                                           *
* 1. The last two digits of the integer determine which suffix it should     *
* have.                                                                      *
* 2. The second to last digit is used for the special case that if the       *
* last two digits are 2 or 3 and if the second to last digit is 1 then       *
* the suffix is th not nd or rd.                                             *
* 3. You extract the digits by using modulo and division on the integer.     *
* 4. 5 branches are required to differentiate the five cases.                *
* 5. By using a while loop to make sure the first letter is either y or n.   *
* 6. By doing the same loop, but making the response all caps or lowercase   *
* then choosing either one to use to interpret response.                     *
* 7. There isnt other loops beside yes/no loop.                              *
* 8. You would need atleast 5 tests to test the 5 special cases.             *
* 9. This lab is designed to interpret an integer value and return a         *
* the suffix of it which is useful for many applications not only to make    *
* it looks good.                                                             *
* 10. By having random messages, you are able to change what the program     *
* prints for each suffix.                                                    *
*****************************************************************************/
bj94684@ares:~$ exit
exit

Script done on 2023-07-14 02:53:38-05:00 [COMMAND_EXIT_CODE="0"]
```