

```

1 *****
2 * PROGRAMMED BY : Nick Reardon
3 * CLASS       : CS1D
4 * SECTION      : MW - 2:30p
5 * Assignment #1 : Recursion
6 *****
7
8             Assignment #1 - Recursion
9
10    Write a recursive function that takes a string as an argument and
11    returns a TRUE if the string is a palindrome otherwise FALSE is
12    returned. A palindrome is a string that reads the same forward
13    and backward (dad, mom, radar). Test with a main program that
14    calls the function. The main function could solicit for input and
15    terminate after receiving an EOF or one can store the strings in a
16    vector or array.
17
18    The recursive function should ignore white spaces, capital letters
19    (case insensitive), and punctuation. For example: ôA man a plan a
20    canal Panamaö would be a palindrome.
21
22    -Use paragraphing and comment each logical block of source code.
23    -Do not use global variables. Your output should be descriptive.
24
25    Use the following input:
26    A man a plan a canal Panama
27    The rain in Spain
28    No lemon, no melon
29    radar
30    CS1D
31    Was it a cat I saw?
32    Racecar
33    Saddleback
34    dad
35
36    Due January 22nd
37 *****
38
39    ** Populating vector with strings read from file
40    ** Testing each string in the vector for palindrome
41
42    -----
43
44    | A man a plan a canal Panama | MATCH
45    | ^                           ^ |
46    | A man a plan a canal Panama | MATCH
47    | ^                           ^ |
48    | A man a plan a canal Panama | MATCH
49    | ^                           ^ |
50    | A man a plan a canal Panama | MATCH
51    | ^                           ^ |
52    | A man a plan a canal Panama | MATCH

```

```

53      ^           ^
54 | A man a plan a canal Panama | MATCH
55      ^           ^
56 | A man a plan a canal Panama | MATCH
57      ^           ^
58 | A man a plan a canal Panama | MATCH
59      ^           ^
60 | A man a plan a canal Panama | MATCH
61      ^           ^
62 | A man a plan a canal Panama | MATCH
63      ^           ^
64 | A man a plan a canal Panama | MATCH
65      ^
66 This IS a palindrome
67
68 -----
69
70 | The rain in Spain | NO MATCH
71      ^           ^
72 This is NOT a palindrome
73
74 -----
75
76 | No lemon, no melon | MATCH
77      ^           ^
78 | No lemon, no melon | MATCH
79      ^           ^
80 | No lemon, no melon | MATCH
81      ^           ^
82 | No lemon, no melon | MATCH
83      ^           ^
84 | No lemon, no melon | MATCH
85      ^           ^
86 | No lemon, no melon | MATCH
87      ^           ^
88 | No lemon, no melon | MATCH
89      ^           ^
90 This IS a palindrome
91
92 -----
93
94 | radar | MATCH
95      ^ ^
96 | radar | MATCH
97      ^ ^
98 | radar | MATCH
99      ^
100 This IS a palindrome
101
102 -----
103
104 | CS1D | NO MATCH

```

```
105      ^ ^
106  This is NOT a palindrome
107
108  -----
109
110  | Was it a cat I saw? | MATCH
111      ^           ^
112  | Was it a cat I saw? | MATCH
113      ^           ^
114  | Was it a cat I saw? | MATCH
115      ^           ^
116  | Was it a cat I saw? | MATCH
117      ^           ^
118  | Was it a cat I saw? | MATCH
119      ^           ^
120  | Was it a cat I saw? | MATCH
121      ^ ^
122  | Was it a cat I saw? | MATCH
123      ^
124  This IS a palindrome
125
126  -----
127
128  | Racecar | MATCH
129      ^     ^
130  | Racecar | MATCH
131      ^     ^
132  | Racecar | MATCH
133      ^     ^
134  | Racecar | MATCH
135      ^
136  This IS a palindrome
137
138  -----
139
140  | Saddleback | NO MATCH
141      ^         ^
142  This is NOT a palindrome
143
144  -----
145
146  | dad | MATCH
147      ^ ^
148  | dad | MATCH
149      ^
150  This IS a palindrome
151
152  -----
153
154  Press any key to continue . . .
```

```
1  /*****
2  * AUTHOR          : Nick Reardon
3  * Assignment #1   : Vectors
4  * CLASS           : CS1D
5  * SECTION         : MW - 2:30p
6  * DUE DATE        : 01 / 22 / 20
7  *****/
8  #ifndef _MAIN_H_
9  #define _MAIN_H_
10
11  //Standard includes
12  #include <iostream>
13  #include <iomanip>
14  #include <string>
15  #include "PrintHeader.h"
16
17  //Program Specific
18  #include <fstream>
19  #include <vector>
20
21
22  // Setup function for PalindromeRecursion
23  // returns true if the given string IS a palindrome, else it returns false
24  // No change to given string
25  bool CheckPalindrome(const std::string& input);
26
27
28  // Recursively checks if a given string is a palindrome
29  // Uses setup function CheckPalindrome
30  // returns true if the given string IS a palindrome, else it returns false
31  // No change to given string
32  // Case insensitive, ignores whitespace and any non alpha numeric character
33  bool PalindromeRecursion(const std::string& input, int front, int back);
34
35
36
37  // Outputs a given string along with given index locations
38  // Used to indicate current character comparisons for palindromes
39  // Indicates two given indices unless both indices match
40  // No change to given string
41  void PrintStringPositions(const std::string& input, int front, int back);
42
43  #endif // _HEADER_H_
44
```

```

1  /*****
2  * AUTHOR          : Nick Reardon
3  * Assignment #1   : Vectors
4  * CLASS           : CS1D
5  * SECTION         : MW - 2:30p
6  * DUE DATE        : 01 / 22 / 20
7  *****/
8  #include "main.h"
9
10 using std::cout; using std::endl;
11
12
13 int main()
14 {
15     /*
16     * HEADER OUTPUT
17     */
18     PrintHeader(cout, "Prompt.txt");
19
20     /*****
21     // input file variable setup
22     std::ifstream iFile;
23     iFile.open("Input.txt");
24
25     std::vector<std::string> inputVect;
26     cout << "    ** Populating vector with strings read from file" << endl;
27
28     // reading from input file into vector
29     while (iFile)
30     {
31         std::string temp;
32         std::getline(iFile, temp);
33         inputVect.push_back(temp);
34     }
35
36     cout << "    ** Testing each string in the vector for palindrome" << endl;
37
38     //Checking each string for palindromes in a loop by calling CheckPalindrome
39     function
40     for (int i = 0; i < inputVect.size() - 1; i++)
41     {
42         cout << endl << "-----" << endl << endl;
43
44         if (CheckPalindrome(inputVect[i]))
45         {
46             cout << " This IS a palindrome" << endl;
47         }
48         else
49         {
50             cout << " This is NOT a palindrome" << endl;

```

```
52
53     }
54 }
55 cout << endl << "-----" << endl << endl;
56
57
58     system("pause");
59     return 0;
60 }
61
62
63 // Setup function for PalindromeRecursion
64 // returns true if the given string IS a palindrome, else it returns false
65 // No change to given string
66 bool CheckPalindrome(const std::string& input)
67 {
68     int back = input.length() - 1;
69
70     return PalindromeRecursion(input, 0, back);
71 }
72
73
74 // Recursively checks if a given string is a palindrome
75 // Uses setup function CheckPalindrome
76 // returns true if the given string IS a palindrome, else it returns false
77 // No change to given string
78 // Case insensitive, ignores whitespace and any non alpha numeric character
79 bool PalindromeRecursion(const std::string& input, int front, int back)
80 {
81     bool match;
82     bool validChar = false;
83
84     while (validChar == false)
85     {
86         if (input[front] < '0' ||
87             (input[front] > '9' && input[front] < 'A') ||
88             (input[front] > 'Z' && input[front] < 'a') ||
89             input[front] > 'z')
90         {
91             front++;
92         }
93         else
94         {
95             validChar = true;
96         }
97
98         if (input[back] < '0' ||
99             (input[back] > '9' && input[back] < 'A') ||
100             (input[back] > 'Z' && input[back] < 'a') ||
101             input[back] > 'z')
102         {
103             back--;
```

```
104         validChar = false;
105     }
106     else
107     {
108         validChar = true;
109     }
110
111     if (!validChar && back <= front)
112     {
113         return true;
114     }
115 }
116
117 PrintStringPositions(input, front, back);
118
119 if (toupper(input[front]) != toupper(input[back]))
120 {
121     return false;
122 }
123 else if ((back - front) < 2)
124 {
125     return true;
126 }
127 else
128 {
129     return PalindromeRecursion(input, ++front, --back);
130 }
131 }
132 }
133
134
135 // Outputs a given string along with given index locations
136 // Used to indicate current character comparisons for palindromes
137 // Indicates two given indices unless both indices match
138 // No change to given string
139 void PrintStringPositions(const std::string& input, int front, int back)
140 {
141
142     if ((front - back) == 0)
143     {
144         cout << " | " << input << " | ";
145         if (toupper(input[front]) == toupper(input[back]))
146         {
147             cout << " MATCH" << endl;
148         }
149         else
150         {
151             cout << " NO MATCH" << endl;
152         }
153         cout << "    " << std::string(front, ' ') << '^' << endl;
154     }
155     else
```

```
156     {
157         cout << " | " << input << " | ";
158         if (toupper(input[front]) == toupper(input[back]))
159         {
160             cout << " MATCH" << endl;
161         }
162         else
163         {
164             cout << " NO MATCH" << endl;
165         }
166         cout << "    " << std::string(front, ' ') << '^'
167             << std::string((back - front) - 1, ' ') << '^' << endl;
168     }
169
170
171 }
```



```

1  /*****
2  * AUTHOR          : Nick Reardon
3  * Assignment #1   : Vectors
4  * CLASS           : CS1D
5  * SECTION         : MW - 2:30p
6  * DUE DATE        : 01 / 22 / 20
7  *****/
8  #ifndef _PRINTER_H_
9  #define _PRINTER_H_
10
11 #include <iostream>
12 #include <iomanip>
13 #include <ostream>
14 #include <string>
15 #include <fstream>
16
17 /*****
18 * PrintHeader
19 * -----
20 * This function will output a class header through the use of ostream.
21 * It also will output the program description
22 * -----
23 * Call
24 * -----
25 * The function call requires 1 parameters. The following example uses an
26 * output file in the ostream parameter. Ex:
27 *
28 *     PrintHeader (oFile);
29 *
30 * -----
31 * Output
32 * -----
33 * The function will output as follows. Ex:
34 *
35 *     *****/
36 *     * PROGRAMMED BY : Parsa Khazravi and Nick Reardon
37 *     * CLASS         : CS1B
38 *     * SECTION       : MW: 7:30pm
39 *     * Lab #3        : Functions - GCD
40 *     *****/
41 *
42 * -----
43 * CONSTANTS
44 * -----
45 * OUTPUT - USED FOR CLASS HEADING
46 * -----
47 * PROGRAMMER       : Name(s) of programmer(s) - Nick Reardon
48 * SECTION          : Class times - MW - 7:30p
49 * CLASS            : Class label - CS1B
50 * PROGRAM_NUM      : # of the program
51 * PROGRAM_NAME     : Title of the program
52 * PROGRAM_TYPE     : Type of program - Lab, Assignment, etc.

```

```

53 *
54 * -----
55 * MAX_OUTPUT      : Max movies to be output at once
56 *****/
57 const std::string PROGRAMMER = "Nick Reardon";
58 const std::string SECTION = "MW - 2:30p";
59 const std::string CLASS = "CS1D";
60 const int PROGRAM_NUM = 1;
61 const std::string PROGRAM_NAME = "Recursion";
62 const std::string PROGRAM_TYPE = "Assignment";
63
64
65 void PrintHeader(std::ostream &output, std::string inputText)
66 {
67     std::string typeNum = PROGRAM_TYPE + " #" + std::to_string(PROGRAM_NUM);
68
69     output << std::left
70         << std::string(76, '*')
71         << std::endl
72         << "* PROGRAMMED BY : " << PROGRAMMER << std::endl
73         << "* " << std::setw(14) << "CLASS" << ": " << CLASS << std::endl
74         << "* " << std::setw(14) << "SECTION" << ": " << SECTION << std::endl
75         << "* " << std::setw(14) << typeNum << ": " << PROGRAM_NAME << std::endl
76         << std::string(76, '*')
77         << std::endl << std::endl
78         << std::string(((76 - typeNum.length() - PROGRAM_NAME.length() ) / 2), ' ')
79         << typeNum + " - " + PROGRAM_NAME
80         << std::endl << std::endl
81         << std::ifstream(inputText).rdbuf()
82         << std::endl
83         << std::string(76, '*')
84         << std::endl << std::endl;
85
86 }
87
88 #endif //_PRINTHEADER_H_

```