```
1  ********************************************************************************
2  * PROGRAMMED BY : Nick Reardon and Danin Namiranian
3  * CLASS         : CS1D
4  * SECTION       : MW - 2:30p
5  * Lab #1        : Templates
6  ********************************************************************************
7
8                                Lab #1 - Templates
9
10   This program will generate 15 random 3 digit numbers and
11   store them in a vector. Then it will:
12     - Sort the numbers
13     - Reverse the digits in each number
14         - Store the reversed numbers in a new vector
15         - Find the some of the 3 digits of each numbers
16         - Store the sums in a new vector
17     - Print all 3 vectors
18
19       Notes:
20         - Numbers less than 100, will display with
21           leading zeroes (eg. 009, rather than just 9)
22         - Sums will be displayed associated with the
23           original number
24
25 ********************************************************************************
26
27  ** Populating with random  numbers
28
29  ** Sorting vectors
30
31  ** Reversing digits and storing in new vector
32
33  ** Summing the digits and storing in new vector
34
35 Original  Reversed
36 103       301       Sum of the digits of 103 is 4
37 185       581       Sum of the digits of 185 is 14
38 271       172       Sum of the digits of 271 is 10
39 321       123       Sum of the digits of 321 is 6
40 370       073       Sum of the digits of 370 is 10
41 377       773       Sum of the digits of 377 is 17
42 467       764       Sum of the digits of 467 is 17
43 522       225       Sum of the digits of 522 is 9
44 640       046       Sum of the digits of 640 is 10
45 642       246       Sum of the digits of 642 is 12
46 647       746       Sum of the digits of 647 is 17
47 656       656       Sum of the digits of 656 is 17
48 816       618       Sum of the digits of 816 is 15
49 961       169       Sum of the digits of 961 is 16
50 994       499       Sum of the digits of 994 is 22
51 Press any key to continue . . .
52
```

```cpp
/*****************************************************************************
 * AUTHOR           : Nick Reardon and Danin Namiranian
 * Lab #1           : Vectors
 * CLASS            : CS1D
 * SECTION          : MW - 2:30p
 * DUE DATE         : 01 / 13 / 20
 *****************************************************************************/
#include "main.h"


int main()
{

    /*
     * HEADER OUTPUT
     */
    PrintHeader(cout, "Prompt.txt");

    /*****************************************************************/


    srand(time(NULL));
    vector<int> intVect(15);
    vector<int> revVect(15);
    vector<int> sumVect(15);

    // Part A and B
    cout << " ** Populating with random  numbers" << endl << endl;
    for (int i = 0; i < 15; i++)
    {
        intVect.at(i) = (rand() % 900 + 100);
    }



    //Part C
    cout << " ** Sorting vectors" << endl << endl;
    sort(intVect.begin(), intVect.end());



    //Part D and E
    cout << " ** Reversing digits and storing in new vector" << endl << endl;

    string reverse;
    char temp;
    for (int i = 0; i < 15; i++)
    {
        reverse = to_string(intVect[i]);
        temp = reverse[0];
        reverse[0] = reverse[2];
        reverse[2] = temp;
```

```
53          revVect[i] = stoi(reverse);
54      }
55
56
57
58      //Part F and G
59      cout << " ** Summing the digits and storing in new vector" << endl << endl;
60
61      int mod;
62      int num;
63      int sum;
64      for (int i = 0; i < 15; i++)
65      {
66          sum = 0;  num = intVect[i];
67          while (num > 0)
68          {
69              mod = num % 10;
70              sum = sum + mod;
71              num = num / 10;
72          }
73          sumVect[i] = sum;
74      } string output;
75
76
77
78      //Part H
79
80      cout << left << setw(10) << "Original" << setw(10) << "Reversed" << endl;
81      for (int i = 0; i < 15; i++)
82      {
83          cout << left << setw(10) << intVect[i];
84          if (revVect[i] < 100)
85          {
86              if (revVect[i] < 10)
87              {
88                  output = string("00" + to_string(revVect[i]));
89                  cout << left << setw(10) << output;
90              }
91              else
92              {
93                  output = string("0" + to_string(revVect[i]));
94                  cout << left << setw(10) << output;
95              }
96          }
97          else
98          {
99              cout << left << setw(10) << revVect[i];
100         }
101
102         cout << "Sum of the digits of " << intVect[i] << " is " << sumVect[i] <<  ↵
              endl;
103     }
```

```
104
105     system("pause");
106     return 0;
107 }
108
```

```
1  /*****************************************************************************
2   * AUTHOR           : Nick Reardon and Danin Namiranian
3   * Lab #1           : Vectors
4   * CLASS            : CS1D
5   * SECTION          : MW - 2:30p
6   * DUE DATE         : 01 / 13 / 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 <vector>
19 #include <stdlib.h>
20 #include <time.h>
21 #include <algorithm>
22
23
24 #endif // _HEADER_H_
25
26
```

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

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