COEN 44 Applied Programming **Fall 2016**

Lab 6

Arrays are a natural structure to store tabular data, and tabular data are a strong motivation for using multidimensional arrays in programs. One source of such tabular data is the US National Center for Health Statistics. This lab will ask to to use multidimensional arrays to store some actual data from the Center, and then perform some calculations using those data.

Objectives:
1. Gain experience with using multidimensional arrays
2. Gain experience using loops to access arrays

Part 1: Design and implement a program to using multidimensional arrays

Among the data maintained by the US National Center for Health Statistics are Life Expectancy at Birth, reported by gender and race for certain years. Actual data at 10-year intervals are shown below.

Your program should read the data values from the keyboard and store them in a 3-dimensional array, where the first dimension represents the decade, the second dimension represents race, and the third dimension represents gender (so, a 7 x 2 x 2 array). Then perform three calculations on the data: the difference in years of life expectancy for women compared to men of the same race in each decade, the difference in years of life expectancy for women of different races and men of different races in each decade, and the percent change in life expectancy for each category from one decade to the next. Use appropriate arrays to store the results of your calculations. Finally, print out the original data, the difference data, and the percent change data.

The actual data are

	Female		Male	
Year	Black	White	Black	White
1950	62.9	72.2	59.1	66.5
1960	66.3	74.1	61.1	67.4
1970	68.3	75.6	60.0	68.0
1980	72.5	78.1	63.8	70.7
1990	73.6	79.4	64.5	72.7
2000	75.2	80.1	68.3	74.9
2010	78.0	81.3	71.8	76.5

NOTES:
You can choose to read in a year as part of the data, or just read in the data in a defined order so the year involved is understood (e.g. four values entered on the first line represent 1950 data). If you read in the year, store the value read in a 1-dimensional integer array, and print those stored values where appropriate for your output. If you don’t read in a year, you will need to generate a correct year value for your output.

You can use \t escape sequences in your printf format statements to produce tabular output. Alternatively, you can use a *field width* specifier in your format placeholder. The format string “%-10d” will print an integer value *left justified* in a 10 column field. The format string “%-10.2f” will print a float value *left justified* in a 10 column field, for a leading 0 for values < 1.0 and > -1.0, and limit the decimal portion to 2 digits. Use %% in your format string to print a single % symbol.

As an example, your output might look like

```
Original data
          Female              Male
Year      Black     White     Black     White
1950      62.9      72.2      59.1      66.5
1960      66.3      74.1      61.1      67.4
1970      68.3      75.6      60.0      68.0
1980      72.5      78.1      63.8      70.7
1990      73.6      79.4      64.5      72.7
2000      75.2      80.1      68.3      74.9
2010      78.0      81.3      71.8      76.5

          Female – Male Life Expectancy Differences
Year      Black     White
1950      3.8       5.7
1960      5.2       6.7
1970      8.3       7.6
1980      8.7       7.4
1990      9.1       6.7
2000      6.9       5.2
2010      6.2       4.8

          White – Black Life Expectancy Differences
Year      Female    Male
1950      9.3       7.4
1960      7.8       6.3
1970      7.3       8.0
1980      5.6       6.9
1990      5.8       8.2
2000      4.9       6.6
2010      3.3       4.7

          Decade Life Expectancy Changes as Percent
          Female              Male
Year      Black     White     Black     White
1960      5.41%     2.63%     3.38 %    1.35%
1970      3.02%     2.02%     -1.80%    0.89%
1980      6.15%     3.31%     6.33 %    3.97%
1990      1.52%     1.66%     1.10 %    2.83%
2000      2.17%     0.88%     5.89 %    3.03%
```

2010	3.72%	1.50%	5.12 %	2.14%

Part 2: Program execution

Lab Steps:

1. Turn in your written program outline to the TA
1. Open Visual Studio and create a new project called Lab6
2. Create your program
3. Debug and test the program
4. When working properly, demonstrate for the TA
5. Turn in a program listing to Camino

NOTE: There are quite a few data values that need to be read each time you test your program. You can start by using dummy data (e.g., 1 2 3 4) for data values until you think your program is working, then enter the real data.