# CA146 Introduction to Programming in C++

## CA146 Tutorial 7

### 1 Practice with one-dimensional arrays

*Note: This question is repeated here from the previous lab. Those who did not have time to finish it then should do so now.*

Data below are from http://scrippsco2.ucsd.edu/data/in_situ_co2/monthly_mlo.csv. They define the Keeling Curve, which depicts global $CO_2$ emissions (probably?) due to human emissions. The values are those from the rightmost column.

2.1 Type in, compile and run the following program, the purpose of which is to illustrate some aspects of the Keeling Curve.

```
// Depicting aspects of the Keeling Curve (CO2 units of ppm)
#include <iostream>
using namespace std;
const int MONTHS = 12;

int
main()
{
   float year1960[MONTHS] = {316.44, 316.37, 316.27, 316.72, 317.29, 317.50,
                             317.46, 317.12, 317.02, 316.82, 316.89, 316.99};
   float year1980[MONTHS] = {338.02, 337.70, 338.65, 338.25, 338.48, 338.89,
                             338.78, 338.91, 338.99, 339.26, 339.15, 339.08};
   float year2000[MONTHS] = {369.15, 368.75, 368.97, 368.94, 368.60, 369.24,
                             369.28, 369.54, 369.98, 370.24, 370.51, 370.47};

   float mean1960, mean1980, mean2000;
   float total1960, total1980, total2000;
   int month;

   // Calculate totals and averages for each year:
   total1960 = 0; total1980 = 0; total2000 = 0;
   for (month = 0; month < MONTHS; month++) {
         total1960 = total1960 + year1960[month];
         total1980 = total1980 + year1980[month];
         total2000 = total2000 + year2000[month];
   }

   mean1960 = total1960 / MONTHS;
   mean1980 = total1980 / MONTHS;
   mean2000 = total2000 / MONTHS;

   // Print monthly, annual total and mean concentrations of CO2:
   cout << "Representative Concentrations of CO2 (Keeling)" << endl << endl;
   cout<<"Month    1960    1980   2000"<<endl;

   for(month = 0; month < MONTHS; month++) {
```

```
        cout << month + 1 << "     " << year1960[month] << "   " <<
           year1980[month] << "   " << year2000[month] << endl;
   }
   cout << endl;
   cout << "Totals" << " " << total1960 << "  " << total1980 << "  " <<
      total2000 << endl;
   cout << "Means " << " " << mean1960 << "  " << mean1980 << "  " <<
      mean2000 << endl;

   return (0);
}
```

2.2 Extend the above program so that it also includes the data for years 1970 and 1990. To obtain the data, refer to the link given above. You should be able to see that there is an acceleration in the concentrations of $CO_2$!

## 2 More practice with one-dimensional arrays

Write a C++ program that:

- Reads in 10 integers from the user and stores them in an array
- Computes and displays the sum and average of the numbers in the array
- Computes and displays the minimum and maximum values in the array and their locations in the array
- Prints out the contents of the array in reverse order

## 3 Even more practice with one-dimensional arrays

3.1 Type in, compile & run the following program:

```
#include <iostream>
using namespace std;

const int SIZE = 5;
int
main()
{
   int numbers[SIZE];
   int i, j;
   int index, value;

   cout << "Enter " << SIZE << " random numbers:" << endl;
   for (i = 0; i < SIZE; i++) {
         cin >> numbers[i];
   }

   for (i = 0; i < SIZE; i++) {
         index = i;
         value = numbers[i];
         for (j = i + 1; j < SIZE; j++) {
            if (numbers[j] < value) {
                  value = numbers[j];
                  index = j;
            }
         }
         numbers[index] = numbers[i];
         numbers[i] = value;

         for (j = 0; j < SIZE; j++) {
            cout << numbers[j] << " ";
         }
         cout << endl;
   }
   return (0);
}
```

3.2 Run the program and see what it does. Add comments to the code to describe what it is doing. You should pay particular attention to accurately commenting the two for loops.

3.3 Modify the program so that it prints the median of the values entered. For an odd number of values the median is the middle value of the sorted sequence.

3.4 Finally modify the program so that it reads in an even number of values and calculates their median. The median of an even number of values is the average of the two middle values of the sorted sequence.