Arrays

April 28, 2018

An array is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed.

1 Min Max

For this programming quiz, we are going to find the min and max and average of 15 numbers that a user will input.

The numbers range from 0 to 100.

We will do it now for practice and then we will do it with arrays. So you do not have to keep all fifteen numbers stored in memory. Find the min and max and average of 15 numbers that a user will input. The numbers range from 0 to 100. So you do not have to keep all fifteen numbers stored in memory.

In [1]:
```
#include <iostream>

int main()
{
    int userInput = 0;
    int maxNumber = 0;
    int minNumber = 100;
    int sumTotal = 0;
    float average = 0;

    //get the numbers from the user
    for(int i = 0; i < 15; i++)
    {
        std::cout << "Enter a number: ";
        std::cin>>userInput;
        std::cout << userInput << "\n";
        if(userInput > maxNumber)
        {
            maxNumber = userInput;
        }
        if(userInput < minNumber)
        {
            minNumber = userInput;
        }
```

```
            sumTotal = sumTotal + userInput;
        }
        std::cout << "Maximum number = " << maxNumber << "\n";
        std::cout << "Minimum number = " << minNumber << "\n";
        average = sumTotal / 15.0;
        std::cout << "Average = " << average << "\n";
        return 0;
    }
```

Arrays can be declared as:
variableType arrayName [ ] = {variables to be stored in the array};
or as:
variableType arrayName[array size]
Information about arrays can be found at: C++ Arrays
We can access the values in an array by identifying the specific index.
variableType arrayName[index number]

2 Sort an array

The user will input 40 integers.

Put them into an array. Then print the array in the order the numbers were entered. Then print in reverse order. Then sort the array in ascending order and print it.

The each print of the array should separate the numbers in the array by one space.

For example: the array were [3,4,55] the printout would be 3 4 55

In [ ]:
```
#include <iostream>
#include <stdio.h>

 int main()
 {
     int userInput[40];

     //Enter the numbers into an array called userInput
     for(int i = 0; i < 40; i++)
     {
         scanf("%d", &userInput[i]);
     }
     //print the array
     std::cout<<"\nThe array\n";
     for(int i = 0; i < 40; i++)
     {
         std::cout << userInput[i] <<" ";
     }
     //print the array in reverse order
     std::cout<<"\n\nThe array in reverse order\n";
     for(int i = 39; i >= 0; i--)
     {
         std::cout << userInput[i] <<" ";
```

```
        }
        //sorting the array
         for(int i = 0; i < 40; i++)
        {
            for(int j = 0; j < 39 - i; j++)
                {
                    if(userInput[j] > userInput[j + 1])
                    {
                        int temp;
                        temp=userInput[j];
                        userInput[j]=userInput[j + 1];
                        userInput[j + 1]=temp;
                    }
                }
        }
    std::cout<<"\n\nThe array sorted\n";
        for(int i = 0; i< 40; i++)
        {
            std::cout << userInput[i] <<" ";
        }
    return 0;
    }
```

3 Search

Find the location of the element in the array

In [ ]:
```
    #include <iostream>
    #include <stdio.h>

    int main()
    {
        int userInput = 0;
        int searchArray[10] = {10, 5, 6, 7, 12 ,19 ,18 ,20, 21, 2};
        //use searchKey for the number to be found
        //use location for the array index of the found value
       int searchKey, location;

        //TODO: write code to determine if integers entered by
        //the user are in searchArray
       location = -1;
       std::cout << "Enter a number: ";
       std::cin>>userInput;
       std::cout << userInput << "\n";

       for(int i = 0; i < 11; i++)
       {
```

```
            if(userInput == searchArray[i])
            {
                location = i;
            }
            }

            //Use these commands to give feedback to the user
            if(location != -1)
            {
                 std::cout<<searchKey<<" is at location "<<location<<" in the array.\n";
            }
            else
            {
                 std::cout<<searchKey<<" is not in the array.\n";
            }
           }
```

4 Multidimensional Arrays

C++ arrays can be of any dimension: 1 to n.
They are initialized with the format:
typeOfVariable arrayName[size of dim.1][size of dim. 2] ...[size of dim. n];
For example:
int array2Dimensions[2][3];
Creates a [2 rows x 3 columns] array of integers.

```
In [2]: int array2Dim[2][3] = {0,1,2,3,4,5};
        for(int i=0; i<2;i++)
        {
            for(int j=0;j<3;j++)
            {
                std::cout<<"array2Dim["<<i<<"]["<<j<<"] = " << array2Dim[i][j]<<"\n";
            }
        }
```

```
array2Dim[0][0] = 0
array2Dim[0][1] = 1
array2Dim[0][2] = 2
array2Dim[1][0] = 3
array2Dim[1][1] = 4
array2Dim[1][2] = 5
```

In this program you will initialize a 4x4 array and a one dimensional array of four elements (also known as a vector). Multiply the array by the vector and print out the resultant vector. It should contain four elements.

```
In [ ]: /*Goal: practice using multidimensional arrays.
        **Write a program that will accept values for a 4x4 array
```

```
**and a vector of size 4.
**Use the dot product to multiply the array by the vector.
**Print the resulting vector.
*/

#include <iostream>

int main()
{
    //TODO: multiply a 4x4 array with vector of size 4.
    //Print the resultant product vector
    const int row = 4;
    const int col = 4;

    int arr[row][col];
    int vector[row], product[row];
    int sum;

    //get the values for the array from the user
    for(int i=0;i<row;i++)
    {
        for(int j=0;j<row;j++)
        {
            std::cout<<"arr["<<i<<"]["<<j<<"] = \n";
            std::cin>>arr[i][j];
            //std::cout<<"arr["<<i<<"]["<<j<<"] ="<<arr[i][j]<<"\n";
        }
    }
     //getting the values for the vector from the user
    for(int i=0; i<row; i++)
    {
        std::cout<<"vector["<<i<<"] = \n";
        std::cin>>vector[i];
        //std::cout<<"vector["<<i<<"] = "<<vector[i]<<"\n";
    }

    sum = 0;

    for(int i=0;i<row;i++)
    {
        for(int j=0; j<row;j++)
        {
            sum = (arr[i][j] * vector[i]) + sum;
        }
        product[i] = sum;
        sum=0;
    }
    for(int i=0;i<row;i++)
```

```
    {
        std::cout<<"product["<<i<<"] = "<<product[i]<<"\n";
    }
    return 0;
}
```