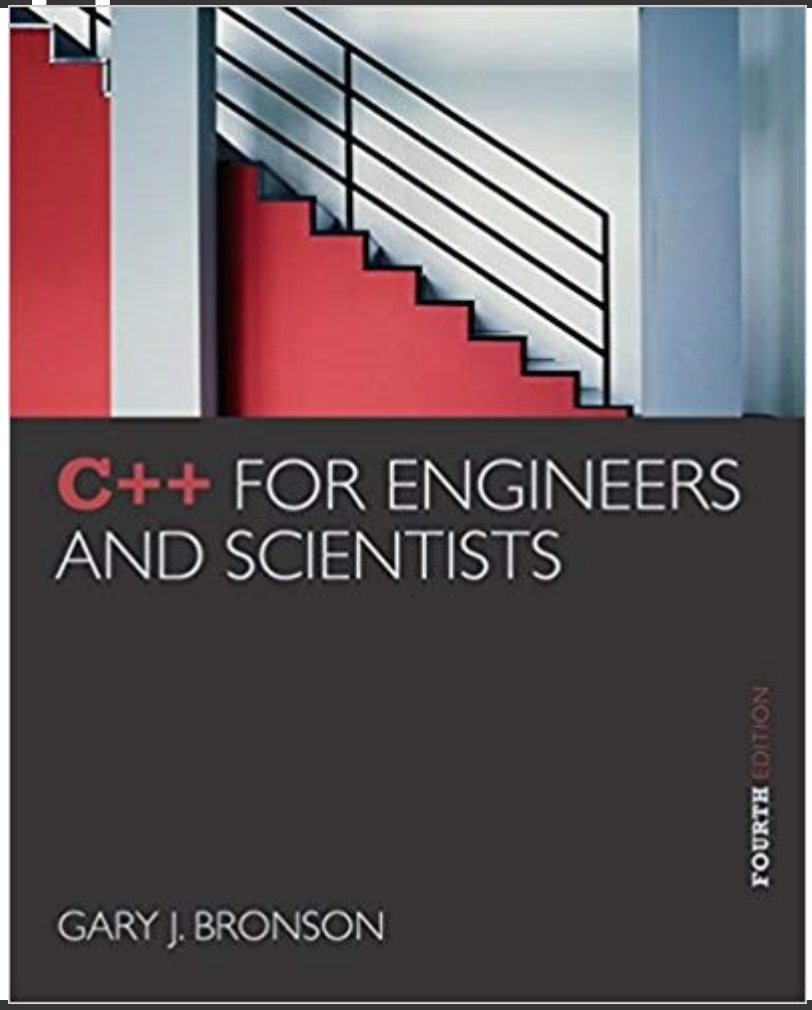
ELEG 1043
Computer Applications in Engineering



Lab Course 6

Acknowledgement

- Some of the slides or images are from various sources. The copyright of those materials belongs to their original owners.

Chapter 7: Arrays

C++ FOR ENGINEERS AND SCIENTISTS

Case Study

- Arrays are useful in applications that require multiple passes through the same set of data elements
  - Statistical Analysis
  - Array: *X* = [98, 82, 67, 54, 78, 83, 95, 76, 68, 63]
  - Calculating
    - Mean value
    - Standard Deviation

Case Study

- Mean value

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N}$$

- Standard Deviation

$$\delta = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N-1}}$$

Mean value

```
double findAvg(int nums[], int numel)
{
    int i;
    double sumnums = 0.0;
    for (i = 0; i < numel; i++)
        sumnums = sumnums + nums[i];
    return (sumnums / numel);
}
```

Standard Deviation

```
double stdDev(int nums[], int numel, double avr)
{
    int i;
    double sumdevs = 0.0;
    for (i = 0; i < numel; i++)
        sumdevs = sumdevs + pow((nums[i] – avr),2);
    return (sqrt(sumdevs/(numel – 1.0)));
}
```

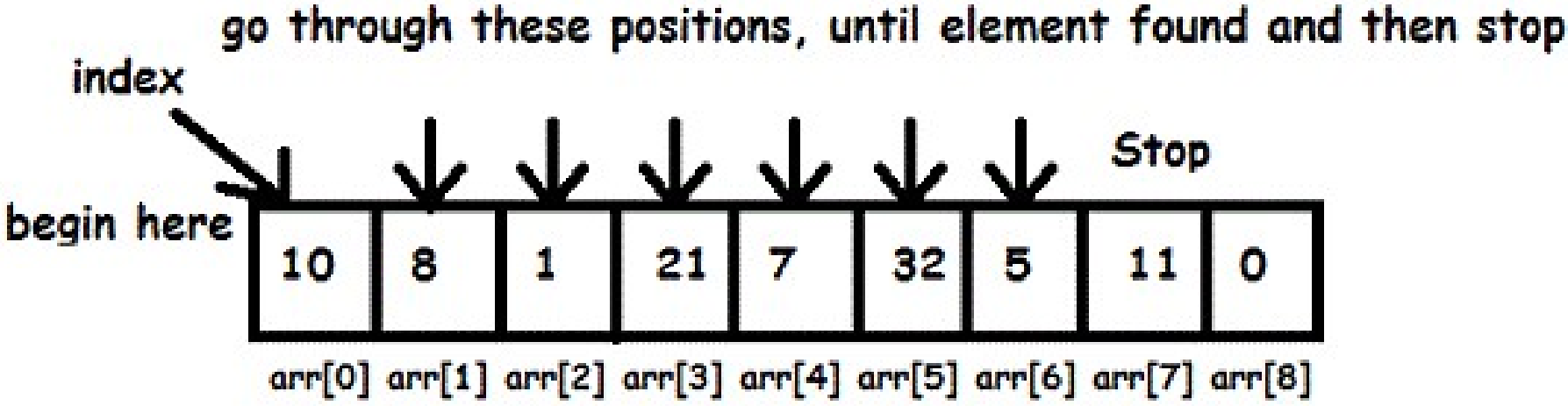
Main Function

```
#include <iosteam>
using namespace std;
int main(){
    const int NUMELS = 10;
    int values[NUMELS] = {98, 82, 67, 54, 78, 83, 95, 76, 68, 63};
    double average, sDev;
    average = findAvg(values, NUMELS); // call the function
    sDev = stdDev(values, NUMELS, average); // call the function
    cout << "The average of the numbers is “<<average << endl;
    cout << "The standard deviation of the numbers is “<<sDev << endl;
    return 0;
}
```

Linear Search

- Each item in the list is examined in the order in which it occurs
- Not a very efficient method for searching
- Advantage is that the list does not have to be in sorted order

Linear Search (continued)



Element to search : 5

Linear Search (continued)

```
#include <iostream>
using namespace std;
/* Linear Search Function */
int linear_search(int arr[], int length, int val);

int main(){
int arr[5] = {3,7,10,6,9};
int val = 6;
cout<<“The index of ”<<val <<“ int the array is ”<<
linear_search(arr, 5, val)<<endl;
return 0;}
```

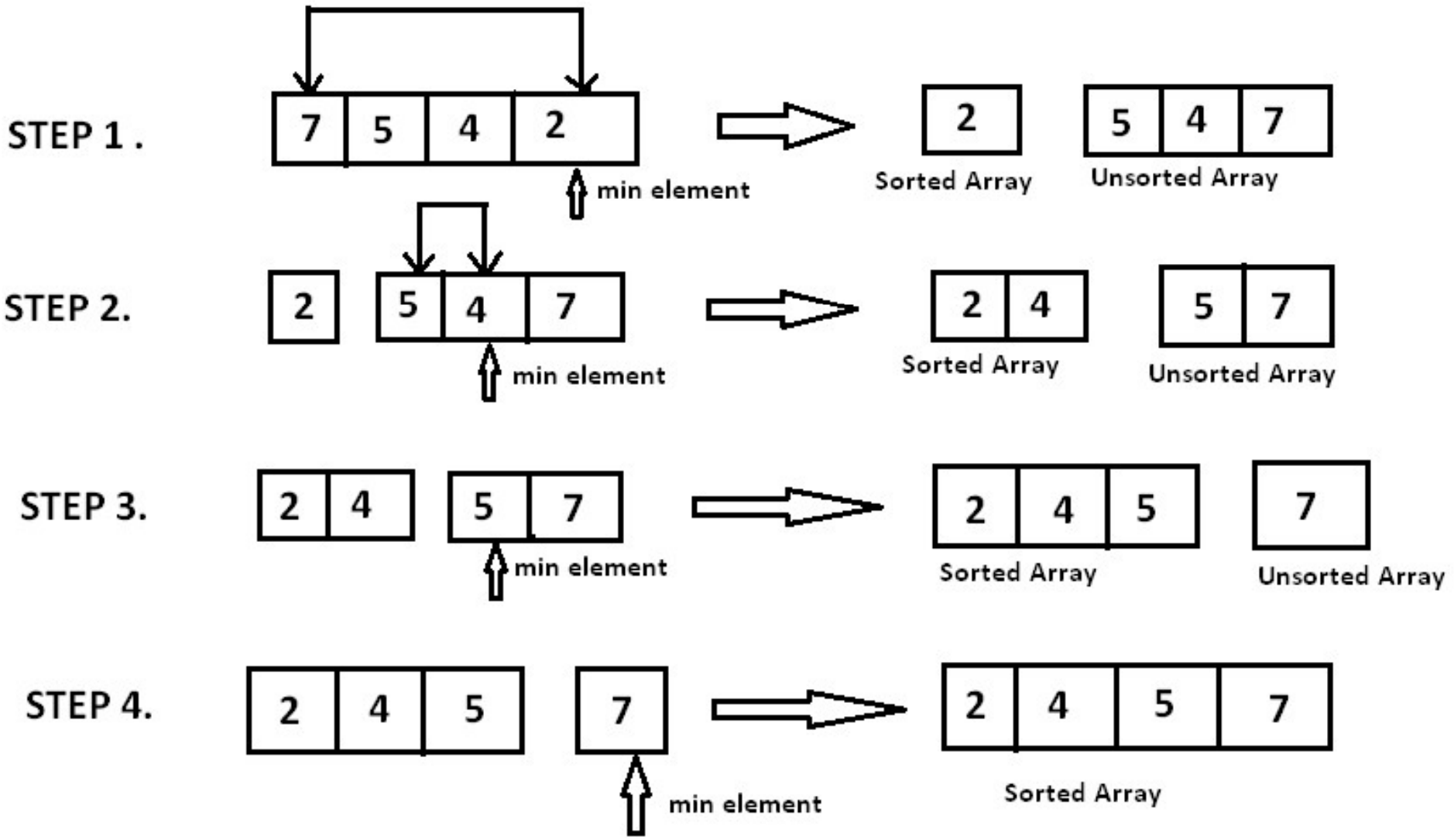
Linear Search (continued)

```
/* Linear Search Function */
int linear_search(int arr[], int length, int val)
{
    int key = -1;
    for (int i = 0; i < length; i++)
    {
     if (arr[i] == val)
     { key = i; break;}
    }
    return key;
}
```

Selection Sort

- Smallest element is found and exchanged with the first element
- Next smallest element is found and exchanged with the second element
- Process continues *n*-1 times, with each pass requiring one less comparison

Selection Sort (continued)



Selection Sort (continued)

```
#include <iostream>
using namespace std;
void selectionSort(int arr[], int length)
int main(){
    int arr[5] = {5,4,3,9,6};
    for(int i = 0; i < 5; i++)
    {cout<<arr[i] <<" ";}
    cout<<endl;
    selectionSort(arr, 5);
    for(int i = 0; i < 5; i++)
    {cout<<arr[i] <<" ";}
    cout<<endl;
    cout<<diff(3,2)<<endl;
    return 0;}
```

Selection Sort (continued)

```
void selectionSort(int arr[], int length)
{
	for(int i = 0; i < length; i++)
	{
		int min = arr[i];
		int minIndex = i;
		for(int j = i; j < length; j++)
		{
		  if(min > arr[j])
		  {
			min = arr[j];
			minIndex = j;
		  }
		}
		cout<<minIndex<<endl;
		int tmp = arr[minIndex];
		arr[minIndex] = arr[i];
		arr[i] = tmp;
	}
}
```

Chapter 10: Pointers

C++ FOR ENGINEERS AND SCIENTISTS

Exercise 1

- Write a program to input 10 positive integer numbers in an array named **Minimum** and determine and display the **Minimum** value entered, where the numbers are received from keyboard

Answer

```
#include <iostream>
using namespace std;
int main(){
    int Minimum[10] = {0};
    int minNum = 10000; \\ assign a large number
    for(int i = 0; i < 10; i++)
    { int num = -1;
      cin>>num;
      Minimum[i] = num;
      if(minNum < num)
      {minNum = num;}
    }
    cout<<minNum<<endl;
    return 0;}
```

Exercise 2

- Write a program to build a function named **multiply** to input the following integer numbers in an array named grades: 12.3, 16.4, and 30.6. As each number is input, multiply the numbers to a variable **mult** and return the **mult** value.

Answer

```
#include <iostream>
using namespace std;
void multiply(int arr[], int length);
int main(){
      double arr[5] = {12.3, 16.4, 30.6};
      cout<< multiply(arr, 3);
      return 0;
}

 void multiply(int arr[], int length);{
        int mul = 1.0;
       for(int i = 0; i < length; i++){
            mul = mul*arr[i];
       }
       return mul;
 }
```