Special Session for test

Arrays recap

- What is an array?
- Let's suppose we have the following numbers: 1, 1.0, 2.3. Can we store all of these numbers in the same array?

Exercises

1. Create a C++ program which sorts a given array of integers in ascending order.
    - Input: `int values[5] = {1, 2, 3, 4, 0};`
    - Expected output: `0 1 2 3 4`;
    - Solution:

```
#include <iostream>
using namespace std;
int main() {
    int values[5] = {1,2, 3, 4, 0};
    int length = sizeof(values)/ sizeof(int);
    for(int i = 0; i < length; i++){
        for(int j = 0;j < length-1; j++){
            if(values[j] > values[j+1]){
                int temp = values[j];
                values[j] = values[j+1];
                values[j+1] = temp;
            }
        }
    }
    for(int i =0; i < length; i++){
        cout << values[i] << " ";
    }
}
```

2. Create a C++ program which sorts a given array of integers in descending order.
    - Input: `int values[5] = {1, 2, 3, 4, 0};`
    - Expected output: `4 3 2 1 0`;
    - Solution:

```
    #include <iostream>
    using namespace std;
    int main() {
        int values[5] = {1,2, 3, 4, 0};
        int length = sizeof(values)/ sizeof(int);
        for(int i = 0; i < length; i++){
```

```
        for(int j = 0;j < length-1; j++){
            if(values[j] < values[j+1]){
                int temp = values[j];
                values[j] = values[j+1];
                values[j+1] = temp;
            }
        }
    }
    for(int i =0; i < length; i++){
        cout << values[i] << " ";
    }
}
```

3. Create a C++ program which finds the maximum element from an array of float numbers.

    - Input: `float numbers[5] = {1.2, 1.4, 1.89, 1.41, 1.90}`;
    - Expected output: `1.9`
    - Solution:

```
#include <iostream>
using namespace std;
int main() {
    float numbers[5] = {1.2, 1.4, 1.89, 1.41, 1.90};
    int length = sizeof(numbers)/ sizeof(float);
    float max = numbers[0];
    for(int i = 1; i < length; i++){
    if(numbers[i] > max){
        max = numbers[i];
    }
    }
cout << max;
}
```

4. Create a C++ program which displays the three biggest numbers in an array of doubles.

    - Input: `double numbers[6] = {1.4, 2.5, 2.3, 2.9, 4.1, 6.2};`
    - Output: `6.2, 4.1, 2.9`;
    - Solution:

```
#include <iostream>
using namespace std;
int main() {
    double numbers[6] = {1.4, 2.5, 2.3, 2.9, 4.1, 6.2};
    int length = sizeof(numbers)/ sizeof(double);
    for(int i = 0; i < length; i++){
        for(int j = 0;j < length-1; j++){
            if(numbers[j] < numbers[j+1]){
                double temp = numbers[j];
                numbers[j] = numbers[j+1];
```

```
                    numbers[j+1] = temp;
                }
            }
        }
        for(int i =0; i < 3; i++){
            cout << numbers[i] << " ";
        }
    }
```

5. Write a C++ program which merges two arrays of the same size and sorts them in descending order.

   - Input:

   ```
       int arr1[5] = {3,5,3,9,8}
       int arr2[5] = {4,8,1,3,5};
   ```

   - Output: `9 8 8 5 5 4 3 3 3 1;`
   - Solution:

   ```
   #include <iostream>
   using namespace std;
   int main() {
       int arr1[5] = {3,5,3,9,8};
       int arr2[5] = {4,8,1,3,5};
       int numbersLength = sizeof(arr1) / sizeof(arr1[0]);
       int result [2 *numbersLength];

       for(int i = 0; i < numbersLength;i++) {
           result[i] = arr1[i];
           result[i+numbersLength] = arr2[i];
       }


       //Sort the array
       int resultLength = sizeof(result) / sizeof(int);
       for(int i = 0; i <resultLength; i++){
           for (int j = 0; j < resultLength-1;j++) {
               if(result[j] < result[j+1]){
                   int temp = result[j];
                   result[j] = result[j+1];
                   result[j+1] = temp;
               }
           }
       }

       //display the array
       for (int i = 0; i < resultLength; i++) {
           cout << result[i] << " ";
   ```

```
    }
}
```

6. Write a C++ program which merges arrays of different sizes into one array and sorts them in ascending.
    - Input:

    ```
    int arr1[5] = {3,5,3,9,8};
    int arr2[3] = {1,3,5};
    ```

    - Output: `1 3 3 3 5 5 8 9`
    - Solution:

    ```
    #include <iostream>
    using namespace std;
    int main() {
        int arr1[5] = {3,5,3,9,8};
        int arr2[3] = {1,3,5};
        int firstArrLength = sizeof(arr1) / sizeof(arr1[0]);
        int secondArrLength = sizeof(arr2) / sizeof(arr2[0]);
        int resultLength = firstArrLength + secondArrLength;
        int result [resultLength];

        for(int i = 0; i < firstArrLength;i++) {
            result[i] = arr1[i];
        }

        for(int i = 0; i < secondArrLength; i++){
            result[firstArrLength+i] = arr2[i];
        }


        //Sort the array
        for(int i = 0; i <resultLength; i++){
            for (int j = 0; j < resultLength-1;j++) {
                if(result[j] > result[j+1]){
                    int temp = result[j];
                    result[j] = result[j+1];
                    result[j+1] = temp;
                }
            }
        }

        //display the array
        for (int i = 0; i < resultLength; i++) {
            cout << result[i] << " ";
        }
    }
    ```

7. Create a C++ program which reads digits from 0 to 9, from the standard input and then displays their frequency. For now, allow the user to type 10 numbers.

    - Input: `1 3 2 4 2 6 3 6 3 6`
    - Output:

    ```
    Digit:     0 1 2 3 4 5 6 7 8 9
    Frequency: 0 1 2 3 1 0 3 0 0 0
    ```

    - Solution:

    ```
    #include <iostream>
    using namespace std;
    int main() {
        int frequency[10]= {0};
        for(int i =0; i < 10; i++){
            int number;
            cin >> number;
            frequency[number]++;
        }

        cout<<"Digit:\t\t";
        for(int i =0; i < 10; i++){
            cout<<i<<" ";
        }
        cout<<endl;
        cout<<"Frequency:\t";
        for(int i = 0; i < 10; i++){
            cout <<frequency[i]<< " ";
        }
    }
    ```

8. Create a C++ program which reads digits from 0 to 9, from the standard input and then displays their frequency. Allow the user to enter as many as She/He wants but stop reading when the user presses `q`.

- Input: `1 3 2 4 2 6 3 6 3 6 q`
    - Output:

    ```
    Digit:     0 1 2 3 4 5 6 7 8 9
    Frequency: 0 1 2 3 1 0 3 0 0 0
    ```

    - Solution:

    ```
    #include <iostream>
    using namespace std;
    int main() {
    ```

```
        int frequency[10] = {0};
        int digit;

        do {
            digit = getchar();
            if(digit !='\n'){
                int position =  digit - '0';
                frequency[position]++;
            }
        } while (digit != 'q' && digit != EOF);

        cout << "Digit:\t\t";
        for (int i = 0; i < 10; i++) {
            cout << i << " ";
        }
        cout << endl;
        cout << "Frequency:\t";
        for (int i = 0; i < 10; i++) {
            cout << frequency[i] << " ";
        }
    }
```