SHEET 4 SOLUTIONS

SHEHAB MAHMOUD SALAH | 2100320

> **GENERAL NOTE**: to **copy** code **from** this PDF document, **copy each block of code** separately to **not** lose the code's formatting, alternatively you can find all the **source code** to **PROGRAMMING EXERCISES** on my **GitHub**: https://bit.ly/CSE131Sheets **happy compiling** 😄

1. This program fragment initializes an **integer array a** with 10 elements, fills it with the **square** of each index, and then prints the elements of the array. The output will be as follows:

```
0149162536496481
```

Here's a breakdown of the output:

```
a[0] = 0*0 = 0
a[1] = 1*1 = 1
a[2] = 2*2 = 4
a[3] = 3*3 = 9
a[4] = 4*4 = 16
a[5] = 5*5 = 25
a[6] = 6*6 = 36
a[7] = 7*7 = 49
a[8] = 8*8 = 64
a[9] = 9*9 = 81
```

There are **no spaces** or separators between the numbers in the output, as the program does not include any code to print them.

2. **(Exercise(1)** on GitHub)

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

int main() {
    vector<float> values(10);

    // Read 10 float values
    for (int i = 0; i < 10; i++) {
        cin >> values[i];
    }

    // Find the maximum value
    float maxValue = values[0];
    for (int i = 1; i < 10; i++) {
        if (values[i] > maxValue) {
            maxValue = values[i];
        }
    }

    // Find the minimum value
    float minValue = values[0];
    for (int i = 1; i < 10; i++) {
        if (values[i] < minValue) {
            minValue = values[i];
        }
    }

    // Calculate the average
    float sum = 0;
    for (int i = 0; i < 10; i++) {
        sum += values[i];
    }
    float average = sum / values.size();

    // Print the results
    cout << "Maximum: " << maxValue << endl;
    cout << "Minimum: " << minValue << endl;
    cout << "Average: " << average << endl;

    return 0;
}
```

BONUS: A simpler, shorter code using the premade libraries **`<algorithm>`** & **`<numeric>`**:

[(**Exercise(1) v2** on GitHub)]

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
#include <numeric>

using namespace std;

int main() {
    vector<float> values(10);

    // Read 10 float values
    for (int i = 0; i < 10; i++) {
        cin >> values[i];
    }

    // Find the maximum value
    float maxValue = *max_element(values.begin(), values.end());

    // Find the minimum value
    float minValue = *min_element(values.begin(), values.end());

    // Calculate the average
    float sum = accumulate(values.begin(), values.end(), 0.0);
    float average = sum / values.size();

    // Print the results
    cout << "Maximum: " << maxValue << endl;
    cout << "Minimum: " << minValue << endl;
    cout << "Average: " << average << endl;

    return 0;
}
```

3. (Exercise(2) on GitHub)

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string names[5] = {"Ahmed", "Ali", "Doaa", "Mohamed", "Osama"};
    float grades[5] = {15, 17, 10, 12, 16};

    // a. Read a name from user and find and print his/her grade
    string inputName;
    cout << "Enter a student name: ";
    cin >> inputName;
    bool found = false;
    for (int i = 0; i < 5; i++) {
        if (names[i] == inputName) {
            cout << inputName << "'s grade: " << grades[i] << endl;
            found = true;
            break;
        }
    }
    if (!found) {
        cout << "Student not found." << endl;
    }

    // b. Print the name of the student with the highest grade
    int highestGradeIndex = 0;
    for (int i = 1; i < 5; i++) {
        if (grades[i] > grades[highestGradeIndex]) {
            highestGradeIndex = i;
        }
    }
    cout << "Student with the highest grade: " <<
names[highestGradeIndex] << " (Grade: " << grades[highestGradeIndex]
<< ")" << endl;

    return 0;
}
```

a. The code reads a name from the user, searches for the name in the **names** array, and if found, prints the corresponding grade from the **grades** array.

b. The code iterates through the **grades** array to find the index of the student with the highest grade, then prints the student's name and grade using the **names** array and the found index.

4. This program fragment will display the **sum of the row and column indices** of the **diagonal elements** of the array `a`. The output will be:

`02468`

Here's a breakdown of the output:

`a[0][0] = 0 + 0 =` **`0`**

`a[1][1] = 1 + 1 =` **`2`**

`a[2][2] = 2 + 2 =` **`4`**

`a[3][3] = 3 + 3 =` **`6`**

`a[4][4] = 4 + 4 =` **`8`**

Here is a graphical representation of the array:

0	**1**	**2**	**3**	**4**
1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8

The diagonal elements are marked with asterisks:

0	**1**	**2**	**3**	**4**
1	*2*	3	4	5
2	3	*4*	5	6
3	4	5	*6*	7
4	5	6	7	*8*

Where, in the modified array `a`*:*

`a[0][0] =` **`0`**

`a[1][1] =` **`2`**

`a[2][2] =` **`4`**

`a[3][3] =` **`6`**

`a[4][4] =` **`8`**

5. (**Exercise(3)** on GitHub)

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

int main() {
    const int maxNumbers = 100;
    int numbers[maxNumbers];
    int input, count = 0;

    // The do-while loop ensures the user enters at least 3 positive numbers, so it
keeps prompting the user to enter numbers until the user enters at least 3 positive
numbers before the negative sign
    do {
        count = 0;
        cout << "Enter positive integers (enter a negative number to stop):" << endl;

          // Read positive integers until the user enters a negative value or reaches
the maximum limit
        while (count < maxNumbers) {
            cin >> input;
            if (input < 0) {
                break;
            }
            numbers[count++] = input;
        }

        if (count < 3) {
            cout << "You must enter at least 3 positive numbers." << endl;
        }
    } while (count < 3);

    // Find and print the three largest integers
    int first = 0;
    int second = 0;
    int third = 0;

    for (int i = 0; i < count; i++) {
        int number = numbers[i];
        if (number > first) {
            third = second;
            second = first;
            first = number;
        }
        else if (number > second) {
            third = second;
            second = number;
        }
        else if (number > third) {
            third = number;
        }
    }

    cout << "The three largest integers are: " << first << ", " << second << ", and "
<< third << "." << endl;

    return 0;
}
```

This concludes Sheet (4) Solutions, this document + source code to all programming exercises available on https://bit.ly/CSE131Sheets .