

ACSE-5

Final Individual Coursework

Mock

**Answer All 4 questions in Part A and
2 out of the 3 questions in part B and
1 out of the 2 questions in part C**

Total Marks: 100

Total time for activity: 3hrs

Section A: (Answer all 4 questions)**(10 marks each)****Question A.1**

Write a program that uses loops to add the following two matrices:

$$A = \begin{bmatrix} 3 & 2 & 8 \\ 1 & 7 & 5 \\ 4 & 6 & 9 \end{bmatrix} \qquad B = \begin{bmatrix} 2 & 3 & 6 \\ 6 & 1 & 9 \\ 8 & 4 & 7 \end{bmatrix}$$

The values in the matrices must be set where they are declared. Once the answer has been calculated it should be displayed on the screen.

Question A.2

Write a program that asks the user for an integer. The program must make sure that the integer is greater than zero and ask the user again if it isn't. It must then ask for a second integer and this time it must ensure that the integer is both greater than zero and less than or equal to the first integer.

Once the program has the 2 integers it should tell the user if the first integer is exactly divisible by the second (i.e. if it is a factor of the first).

Question A.3

Write a **function** that takes in a string and 2 additional single characters. The function should change every instance of the 1st character into the second character. The main body of the program should read in the string and the 2 characters from the user, run the function and then return the result. The following is an example of the how it should run (bold is user input):

Enter a string: **Chonge my chorocters**

Enter character to change: **o**

Enter character to change to: **a**

Result: Change my characters

Question A.4

Create a class that contains the x, y and z coordinates of a 3D point in space. It should have a member function that takes in another point of the same class type and returns the scalar distance between the current point and the point passed to the member function. There must also be another member function to read in the coordinates from the user. In the main body of the program use these member functions to read in two points from the user and display the distance between them.

Section B: (Answer 2 out of 3 questions only)**(20 marks each)****Question B.1**

Write a program to store names and telephone numbers. These should be stored in a class that holds a person's first name, surname and telephone number. There should be member functions to read in the data, to display the data and to write the data to a text file (note that the member function should only write its own data to file).

The user should be asked how many people there are in the directory and then dynamically allocate enough memory to store that much data. The appropriate member function should then be called for each of the items to get the data from the user. All the data should also be written to a text file.

Once the data has been read in, the user should be allowed to search for a phone number. The user should be able to enter a surname and the data for all people with that surname should be displayed. The user should then be asked if they wish to search for another surname.

Question B.2

For an array of integers write to the screen how many numbers occur a particular number of times in the array. For instance, if the list of numbers is as follows:

3 1 14 1 2 7 1 3 11 14 7 4 8

..., then the output from the program should be as follows:

4 number/s appear 1 times

3 number/s appear 2 times

1 number/s appear 3 times

No numbers appear more than 3 times

Test your program using a statically declared (i.e. normal) array containing the above numbers.

To do this calculation you will need to have a second array which has at least one more entry than the largest number in the first array (i.e. the second array should have at least 15 entries in this case). All the values in this array should initially be set to zero. You should then go through the main array and for each number in that array increment the number at the corresponding index in the second array (i.e. if the number at the i th location in the first array is n , then increment the n th number in the second array).

At this point you have enough information to create the above table. Do this by finding the highest number in the 2nd array (no numbers appear in the main array times more than this number). For each number less than or equal to this number count how many times it appears in the second array and display the result.

Question B.3

Write a function to calculate the greatest common divisor of 2 integers. One possible way to do this is by using a recursive algorithm, known as the Euclidian algorithm. The algorithm can be expressed as follows:

The greatest common divisor of x and y is y if the remainder when x is divided by y is zero otherwise it is the greatest common divisor of y and the remainder when x is divided by y .

Allow the user to enter pairs of numbers and display the greatest common divisor. The 3 numbers should also be written to file in text format. Ask the user if they wish to repeat the calculation. When the program exits all the entered numbers and their corresponding greatest common divisors should be stored in a single file (as text).

Section C: (Answer 1 out of 2 questions only)

(20 marks each)

Question C.1

Implement a method called *checkIfSameType* that takes in two objects A and B as inputs. The two input parameters to this method should be templated. The method *checkIfSameType* should return true if A and B are objects of the same type, and false if A and B are objects of different types.

Question C.2

Write your own version of the `std::vector<T>` templated container, called *myVector* (note you cannot use the `std::vector<T>` in your implementation). This should be a templated class, where you are required to implement a constructor, destructor, and a *push_back* method that takes a one input argument, a pointer to an object of the templated type T. The *push_back* method then copies the object the input pointer points at, and stores this in dynamically allocated memory, which should be a private variable in your *myVector* class. You should use a smart pointer to keep track of this private heap memory. This heap memory should also grow in size when *push_back* is called, where needed.