CONFIDENTIAL

UTM Faculty of Computing
UNIVERSITI TEKNOLOGI MALAYSIA

MID TERM TEST II (PRACTICAL)
SEMESTER II, 2023/2024

SUBJECT CODE	:	**SECJ1023**
SUBJECT NAME	:	**PROGRAMMING TECHNIQUE II**
SECTION	:	**1 - 10, and 15**
TIME / DURATION	:	**8PM – 10PM / 2 HOURS**
DATE / DAY	:	**14 MAY 2024 / TUESDAY**
VENUE	:	**COMPUTER LABS**

INSTRUCTIONS

- This is a CLOSED-BOOK practical test. You will attempt the test by coding on a computer.
- You will be provided with some reference resources. You are only allowed to refer to the provided resources.
- References to other resources including any source code, any websites, etc., are strictly prohibited.
- The use of any kind of artificial intelligence tool is strictly prohibited. These tools include ChatGPT, AI-based VS Code extensions such as Co-Pilot, etc.
- You should code offline using a C++ IDE such as VS Code, Dev C++, etc.
- You must use a PC provided in the lab. The use of your own computer is not allowed.
- You will only use e-learning to download the provided resources and submit your answers. Please refer to the e-learning for your section.
- All COMMENTS in your program WILL NOT BE EVALUATED.
- Programs that CANNOT COMPILE will be penalized by 50%.
- Be warned that your program will be subject to a screening process for PLAGIARISM DETECTION.
- NO TECHNICAL ASSISTANCE will be provided during the test. This also applies to problems such as the IDE not working, the program not compiling, etc. If you encounter a problem, you must solve it yourself.
- All questions related to the test question will not be entertained at all. Please read the question carefully.

DOWNLOAD THE RESOURCES PROVIDED FROM ELEARNING

There is **no starter code** provided for this test. You have to write the program from scratch. Write the program using inline style and use only a single source code file. Indicate the question number (or task number) by writing it as a comment in your code.

Question **[100 Marks]**

Define a class named **`Vector`** to model a mathematical vector. This class should hold its elements in a dynamically created integer array. You must **NOT** use the built-in C++ `vector` class. Declare the class according to the following code snippet:

```
class Vector {
  private:
    int *items; // the list elements
    int size;   // number of elements
  public:
     // other members should go here
};
```

Complete the code for the class in your program by completing the following tasks:

1. Define multiple constructors as follows:
   a. A default constructor that sets all the attributes to appropriate values. (2 marks)
   b. A constructor that sets the attribute **`size`** with a value passed as a parameter. This constructor should also allocate memory for the array **`items`**. (3 marks)
   c. A constructor that sets both attributes, **`size`** and **`items`** from parameters. (4 marks)
   d. A constructor that accepts a string that specifies a list of digits and uses these digits to set the array. For example, if the string passed is **`"5678"`**, then the attributes will be set as **`items = {5,6,7,8}`** and **`size = 4`**. (6 marks)
   e. A copy constructor that creates a copy of another vector. The copied version should have its own array. (4 marks)
2. Define a destructor that deallocates the memory of the array. (2 marks)

3. Define several conversion methods as follows:

   a. A method that evaluates the vector to an integer value. In this context, this method returns the size of the array (i.e., the number of elements). (2 marks)

   b. A method that evaluates the vector to a **`double`** value. This method returns the magnitude of the vector. The magnitude of a vector is the square root of the sum of the squares of the elements, as shown in the following equation:

$$magnitude = \sqrt{\{v_0^2 + v_1^2 + \ldots + v_{n-1}^2\}}$$

   where v_i is the element at i-th index in the array **`items`** of size n. (5 marks)

   c. A method that converts the vector to a **`string`** object. For example, converting a vector whose **`items = {8,9,10}`** to a string will result in the string **`"8,9,10"`**.

   (8 marks)

4. Define several overload operators as follows:

   a. A multiplication operator that performs element-wise multiplication between a vector object and an integer value. For example, if a vector whose **`items = {1,3,5}`** is multiplied by an integer value of 2, it will result in a new vector object with **`items = {2,6,10}`**. (6 marks)

   b. An addition operator that adds a vector object to another vector object element-by-element. For example, if two vector objects respectively are **`a = {1,3,5}`**, and **`b = {7,8,9}`**. If you apply this operator to both vectors, you will get a new vector whose elements are **`{8,11,14}`**. (6 marks)

   c. Another addition operator that adds a vector object with an integer value. The result is a new vector that combines the elements of the vector and the integer value. For example, if a vector object, **`a = {1,3,5}`** is added to an integer of 88, the result will be a new vector with elements of **`{1,3,5,88}`**. (6 marks)

   d. A subscript operator to access or obtain the value of an element. For example, to access the value of the second element of vector **`v`**, instead of writing **`v.elements[1]`**, you can write it as **`v[1]`**. (2 marks)

e. An equality operator that compares two vectors whether they are exactly equal including their sizes. This operator compares the equality element by element. It should return **True** only if all elements in both vectors are equal. (8 marks)

f. A greater than operator that compares two vectors based on their magnitudes. It should return **True** if the magnitude of the first vector is greater than the second. (4 marks)

5. Define a regular function:

   a. Define a function that prints the information about a vector. Use proper parameters to declare the function. The printed output should follow the format below:

      `title` `( (`**`elements`**`), s:` **`size`**`, m:` **`magnitude`** `)`

      Example: `a: ( (6,7,8,9), s: 4, m: 15.1658 )`

      where:

      `title` is the output title which indicates the purpose of the output.

      `elements` is the elements of the vector, separated by commas.

      `size` is the vector's size (i.e., the number of elements).

      `magnitude` is the vector's magnitude. (5 marks)

   b. Specify this function as a **friend** to the **`Vector`** class. (1 mark)

6. In the main function of the program, use the class to perform the following tasks:

   a. Create a vector named **`a`**, that is initialized with the constructor defined in 1(c). Set the vector's elements with **`{6,7,8,9}`**. (2 marks)

   b. Create another vector named **`b`**, that is initialized with the constructor defined in 1(d). The vector created should contain elements of **`{6,7,8}`**. (1 mark)

   c. Create another vector named **`c`**, which is initialized with the copy constructor defined in 1(e). This vector should be copied from vector **`b`**. (1 mark)

   d. Declare a pointer of **`Vector`** named **`d`** and point it to a dynamically created vector object. The dynamically created object is copied from vector **`b`**. (2 marks)

   e. Using a proper operator from the class, perform an addition of the vector created in 6(d) with the integer value 1, and put the result into another vector named, **`e`**. (2 marks)

   f. Perform a multiplication of the vector **`c`** by the integer value 2, and put the result into another vector, **`f`**. (2 marks)

g. Perform an addition between the vectors **a** and **e** and write the result in another vector, **g**. (2 marks)

h. Using the function defined in Task 5, print out all the vectors created in tasks 6(a) to 6(g). Look at the example program output in Figure 1. (7 marks)

i. Use a proper operator, to test whether the vectors **a** and **b** are equivalent. Give the result in a proper output. See the example output in Figure 1. (3 marks)

j. Use a proper operator, to test whether vector **a** is greater than vector **e** or vice-versa. Display the result in a proper output. See the example output in Figure 1. (3 marks)

k. Destroy the object that was dynamically created in 6(d) in the correct way. (1 mark)

```
a: ( (6,7,8,9), s: 4, m: 15.1658 )
b: ( (6,7,8), s: 3, m: 12.2066 )
c: ( (6,7,8), s: 3, m: 12.2066 )

d: ( (6,7,8), s: 3, m: 12.2066 )

e: ( (6,7,8,1), s: 4, m: 12.2474 )
f: ( (12,14,16), s: 3, m: 24.4131 )
g: ( (12,14,16,10), s: 4, m: 26.3818 )

a is not equivalent to b
e is greater than a
```

Figure 1: An example output of the program.

-End of Question--