Data Structures and Algorithms Lab

Lab 05 **Marks 10**

Instructions

Work on this lab individually. You can use your books, notes, handouts etc. but you are not allowed to borrow anything from your peer student.

Marking Criteria

Show your work to the instructor before leaving the lab to get some or full credit.

What you must do

Program the following task in your C++ compiler and then compile and execute them. *Write **main** function first and keep on testing the functionality of each function once created.*

ADT: PointList

Given the **Point3D** structure

```
struct Point3D
{
   float x, y, z;
};
```

Write a class to store data items in an unordered list (**PointList**). Each data item in a point list is of type **Point3D**.

The class should have following **four private data members**:

1. A pointer to a **Point3D** structure that holds an **array** of **Point3D** allocated dynamically according to the specified maxSize
2. An integer maxSize that holds the capacity *(maximum number of elements a list can hold)*.
3. An integer curSize that holds a current number of elements exist in a list.
4. An integer cursor to hold the current position *(index)* of the list. If the list is empty, it should hold -1.

The class should support the following operations:

A. A **constructor** who accepts an **integer** as argument to represent the **capacity *(maximum size)*** of a list and initializes it to the so-called "empty list," i.e., a list whose all elements are set to **zero**.

B. A **destructor** to **free any memory resources** occupied by the **PointList** object.

C. `bool isEmpty()` returns *true* if a list is empty, *false* otherwise.

D. `bool isFull()` returns *true* if a list is full, *false* otherwise.

E. `void insert(Point3D newPoint)` inserts newPoint into the list. If list is not empty, then inserts newPoint at the end *(next available slot)*. Otherwise, inserts newPoint as the first (and only) data item in the list. In either case, moves the cursor to newPoint and increment the current size.

F. `void showStructure()` outputs the data items in a list. If the list is empty, outputs "Empty list".

G. `Point3D getCursor()` if a list is not empty, returns a copy of data item marked by the cursor. Give appropriate error message otherwise and return a Point3D object having all values set to **-1**.

H. `void gotoBeginning()` if a list is not empty, then moves the cursor to the **beginning** of the list.

I. `void gotoEnd()` if a list is not empty, then moves the cursor to the data item at the end *(current size)* of the list.

J. `bool gotoNext()` if the cursor is not at the end *(current size)* of a list, moves the cursor to the next data item in the list and returns *true*, *false* otherwise.

K. `bool gotoPrior()` if the cursor is not at the beginning of a list, moves the cursor to the **previous** item in the list and returns *true*, *false* otherwise.

L. `void clear()` make the list empty *(a list having no elements)*.

M. `bool replace(Point3D newPoint)` replaces the data item marked by the cursor with newPoint and returns *true*, *false* otherwise. The cursor remains on its current position.

N. `void remove()` removes the data item marked by the cursor from a list. If the resulting list is not empty, the cursor should remain on its current position. If the deleted data item was at the end *(current size)* of the list, moves the cursor to the beginning of the list.

In ***main*** function, create few objects of ***PointList*** class and demonstrate the working of each function clearly.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺