

# Data Structures and Algorithms Lab

## Instructions

Work on this lab individually. *Write main function first and keep on testing the functionality of each function once created.*

Program the following tasks in your C++ compiler and then compile and execute them.

Email your solution (.cpp) file only to the following respective recipient till **Thursday, March 04, 2021**.

**DO NOT** compress/zip your solution.

The email must be sent from your **official PUCIT email id**, otherwise it will **NOT BE ACCEPTED** and will be marked **ZERO**.

**2 MARKS** will be **DEDUCTED** if submission instructions are not followed.

The subject of the email should be the name of the lab e.g. **Lab 01**.

Degree	Recipient Email	Subject of Email
BSIT Morning	<a href="mailto:dsaubt01@gmail.com">dsaubt01@gmail.com</a>	Lab 01
BSIT Afternoon	<a href="mailto:dsaubt02@gmail.com">dsaubt02@gmail.com</a>	

You are strictly not allowed to add any other data-member/constructor/function in the class. You are also not allowed to change the name or prototype of any data-member/constructor/function.

## ADT: Collection

Write a class named **Collection** for which each object can hold **negative integers** and **zero** as a default value.

- The class should have following **two private data members**.
  - An **integer pointer** named **data** that holds an **array of integers** allocated dynamically according to the specified **size**.
  - An **integer** named **size** that holds the **size of the array** (amount of memory allocated to data).
- Provide the implementation of following **constructors** and a **destructor**
  - A **constructor** which accepts an **integer** as argument to represent the **size of an array** and initializes it to the so-called "empty collection," i.e., a collection whose array representation **contains all zeroes**.
  - An additional **constructor** that receives an **array of integers** and the **size of that array** as its arguments and uses the array to initialize a **collection object**.
  - A **copy constructor** to initialize a collection object with already existing object.
  - A **destructor** to **free any memory resources** occupied by the **collection** object.
- Provide following member functions for the common operations
  - getSize** returns the size of collection.
  - setElement** that **inserts** a new integer **k** at index **i** (both passed as argument) into a **collection**, if possible, otherwise give an appropriate error message.
  - countElement** accepts an integer **key** as argument and **count and return** the **total occurrences** of it in a collection, **-1** otherwise, if the key does not exist.
- Provide the implementation of following overloaded operators
  - Stream insertion (<<)** to display the contents of **data** on the screen of a **collection**.
  - Stream extraction (>>)** to take input from user for the **data** of a **collection**.
  - Assignment (=)** which copies the data of one object to another. This assignment should only be made, if both objects have same size, give appropriate error message otherwise.
  - Addition (+) binary** which perform the addition of two collections (right hand side from left hand side) and **return** the result. This addition should only be performed, if both the objects have same size, give appropriate error message otherwise.
  - Comparison (==)** that determines whether **two collections are equal or not**. The operator should returns **true** if both the collections have same data, **false** otherwise.

**No submission will be accepted after the DUE DATE.**

B E S T O F L U C K