**Diya Goyal 102215255 2NC11**

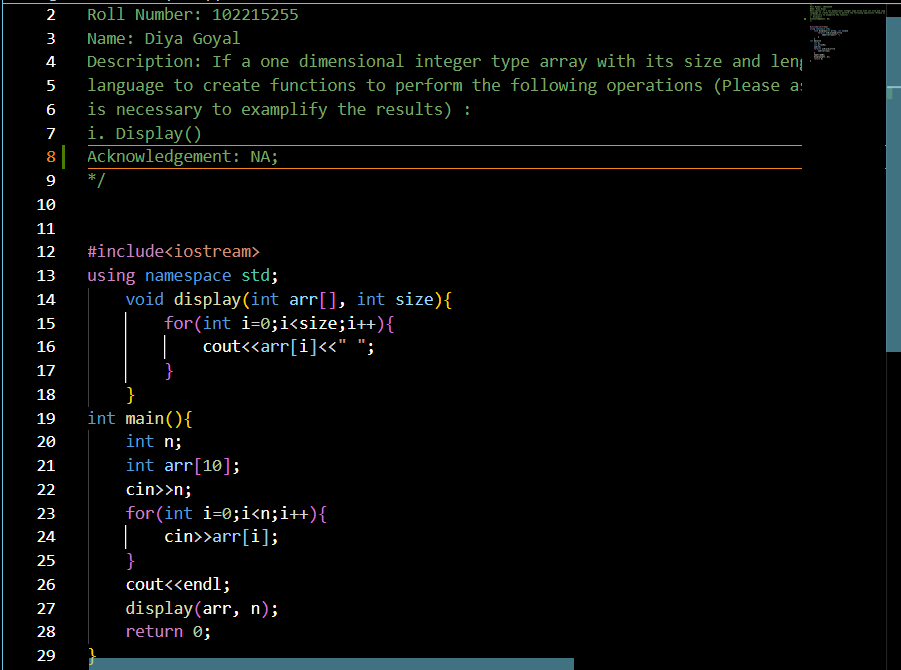
**Lab Assignment 2**

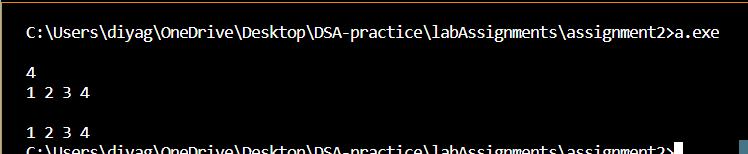
**UCS 406 Data Structures and Algorithms**

Note: Use C/C++ or JAVA programming language.

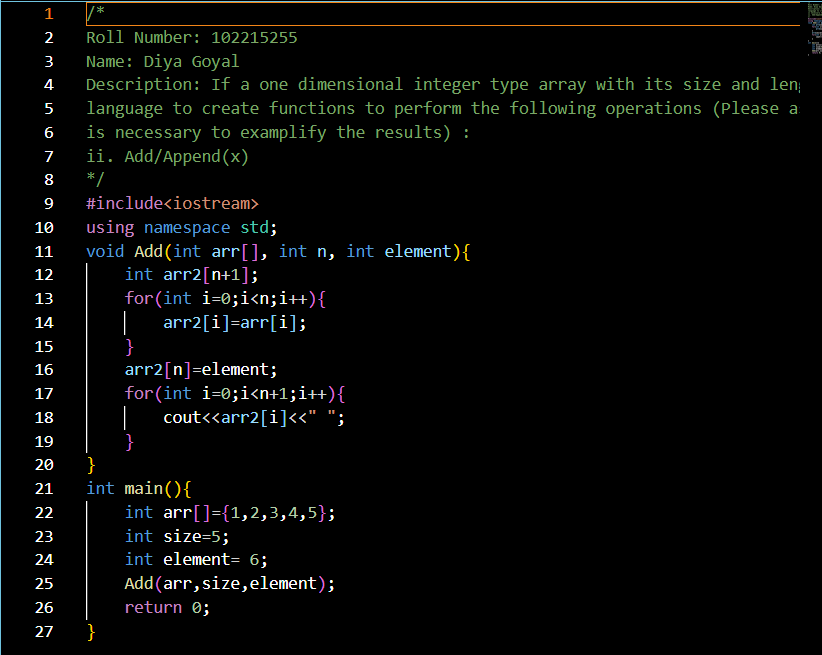
Q1. If a one dimensional integer type array with its size and length given, write the code in C++ language to create functions to perform the following operations :

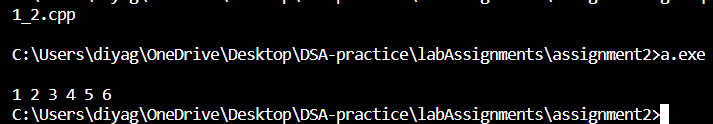
1. Display()



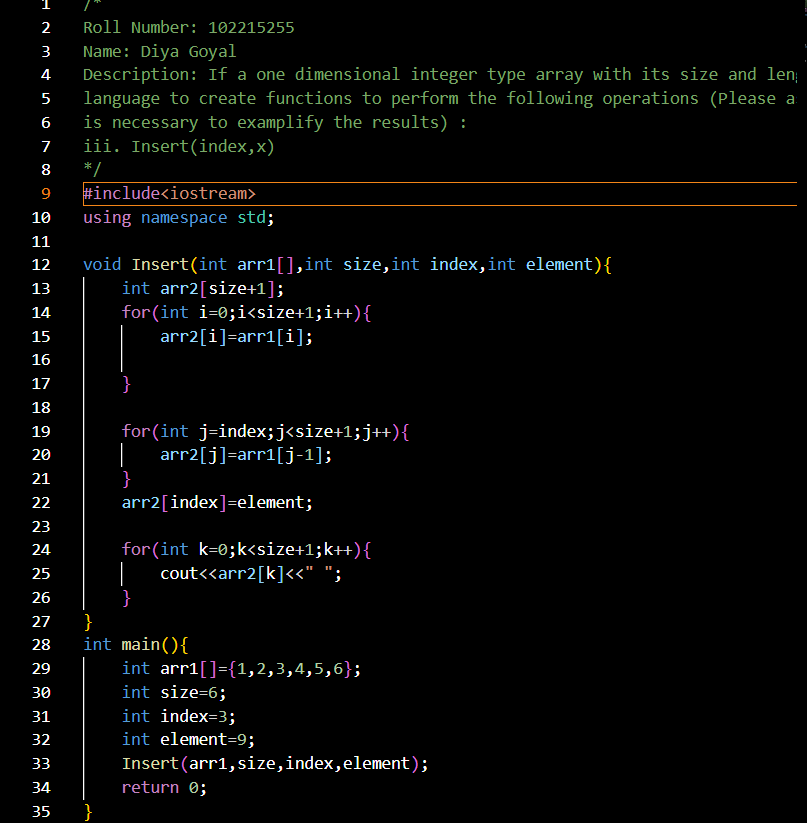


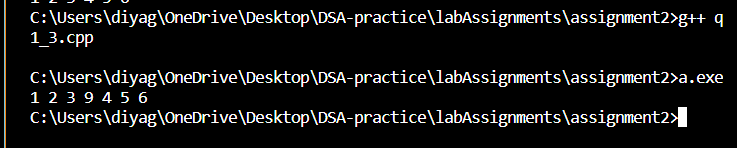
1. Add/Append(x)



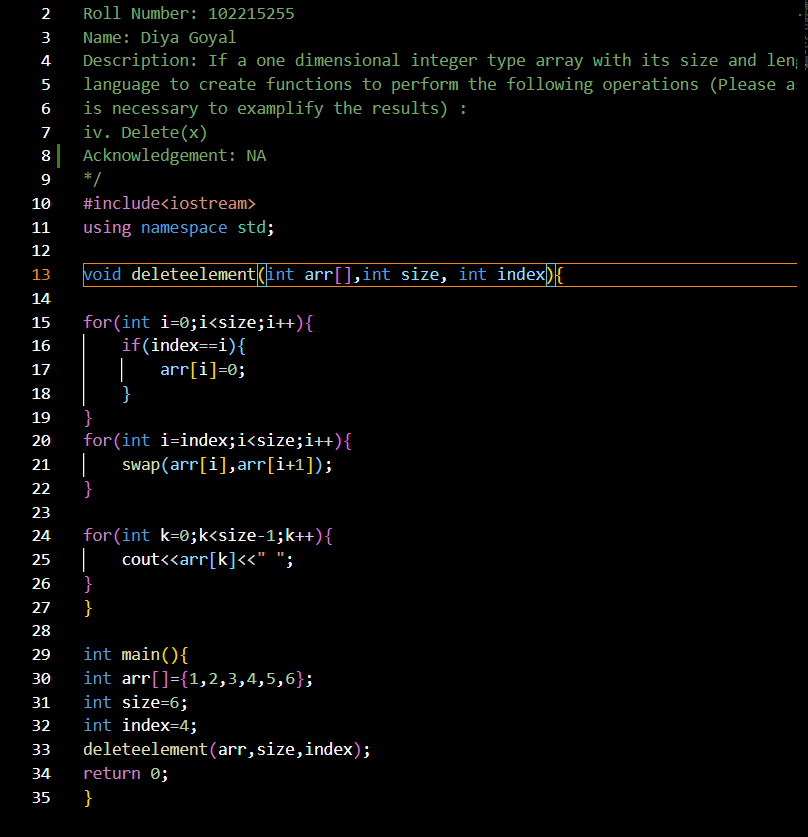


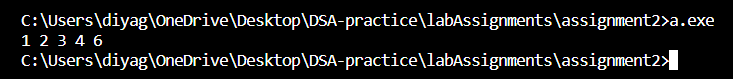
1. Insert(index,x)



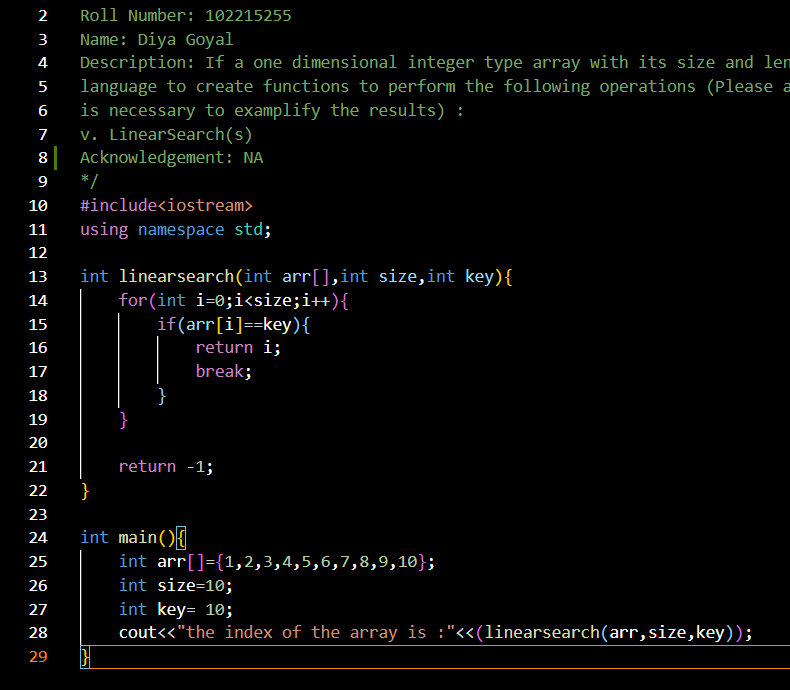


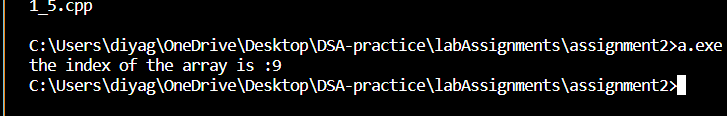
1. Delete(x)



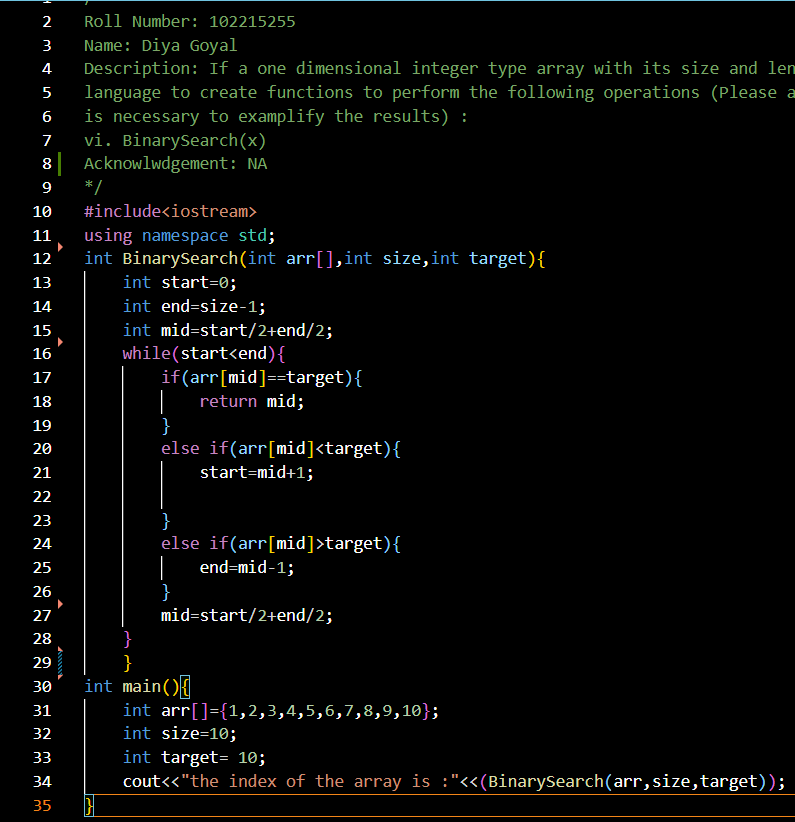


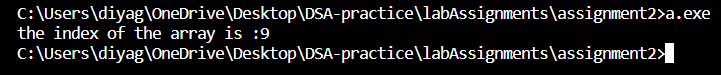
1. LinearSearch(x)



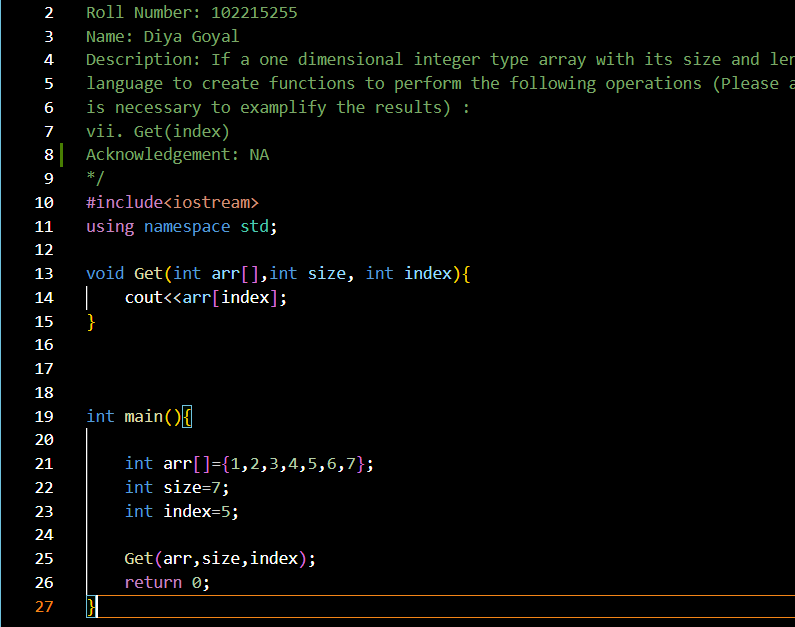


1. BinarySearch(x)



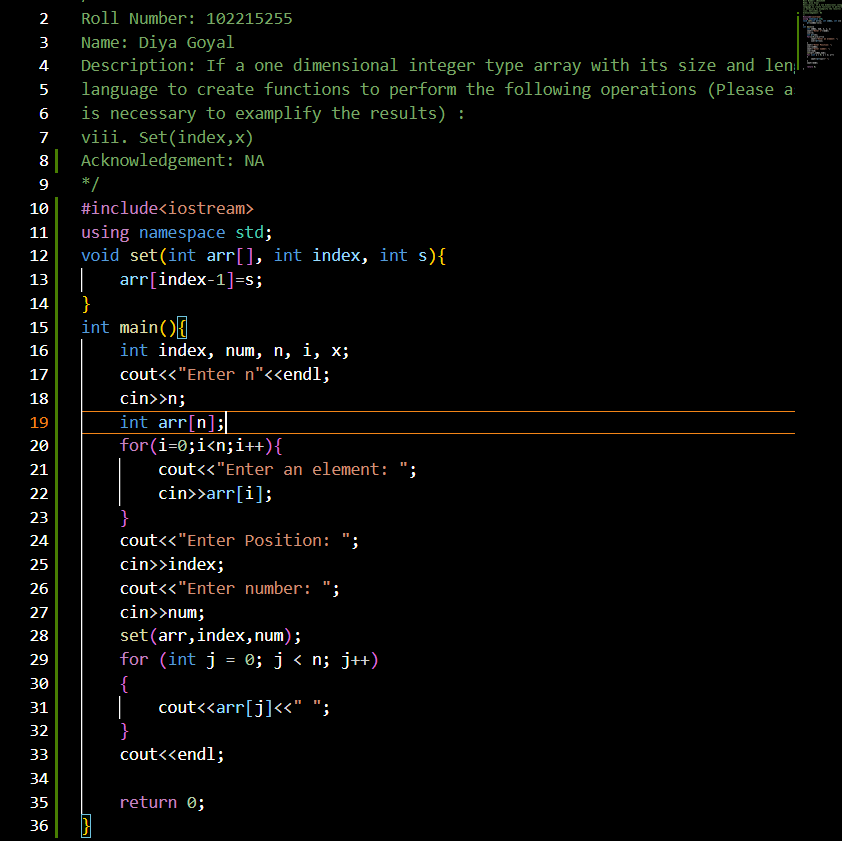


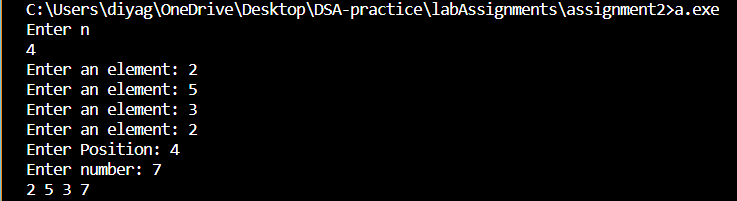
1. Get(index)



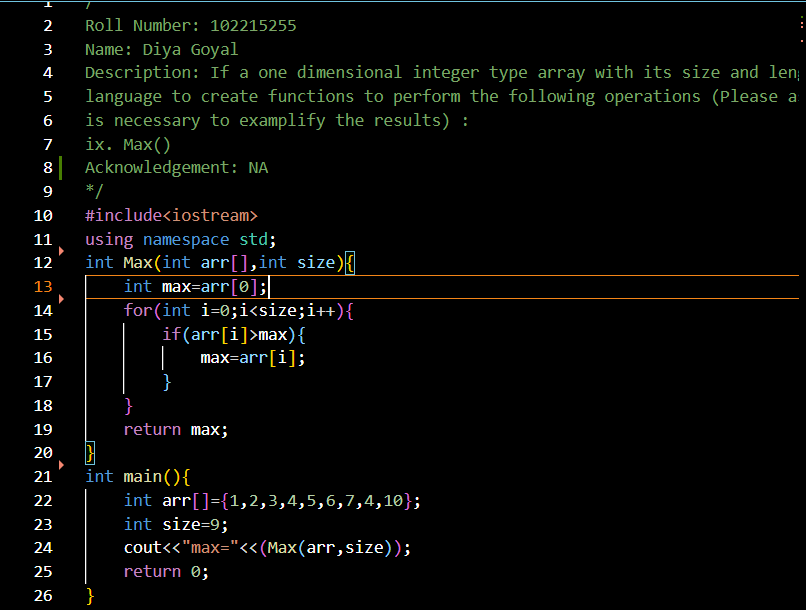


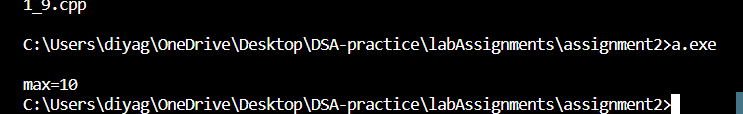
1. Set(index,x)



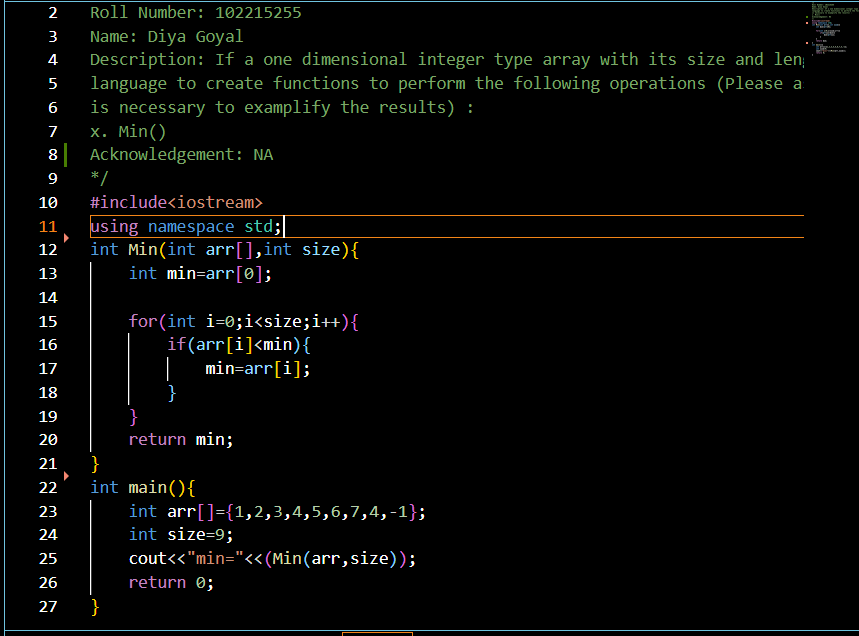


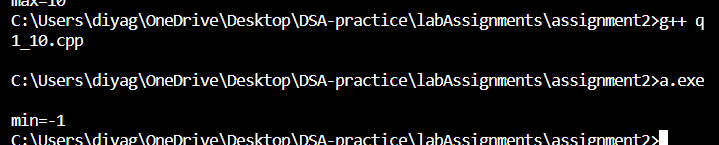
1. Max()



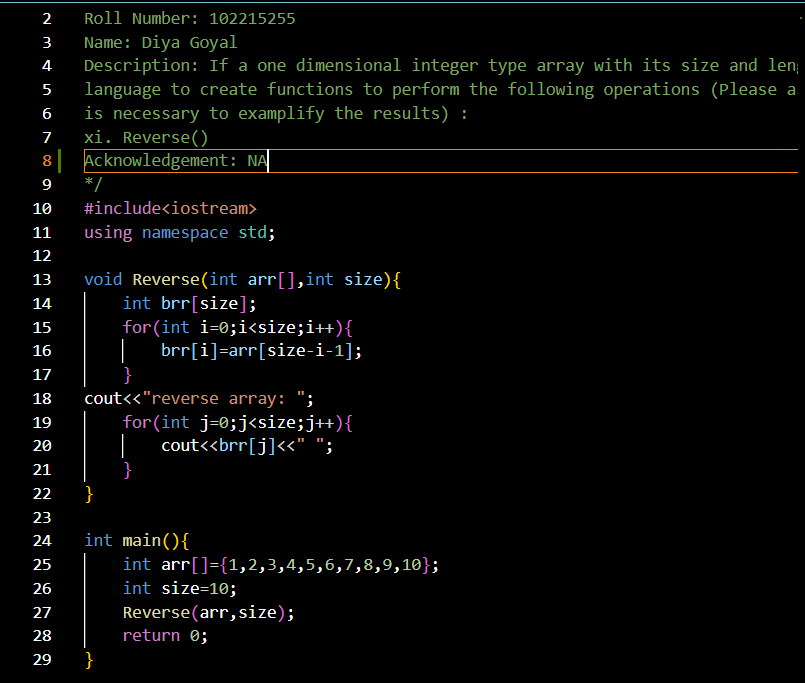


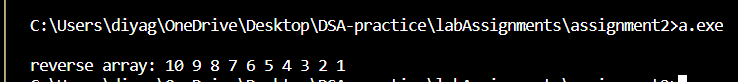
1. Min()



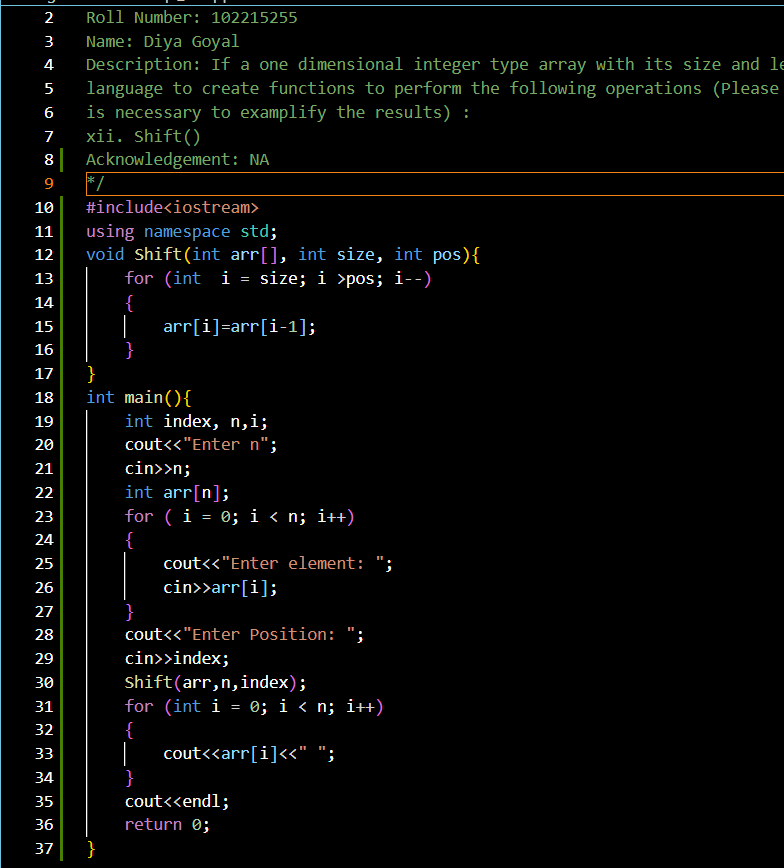


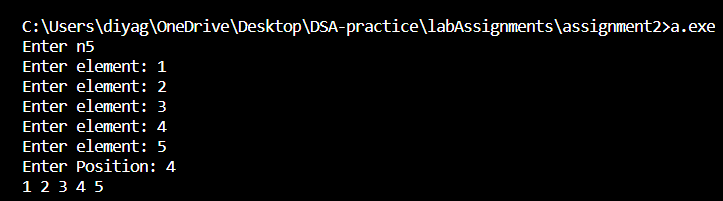
1. Reverse()



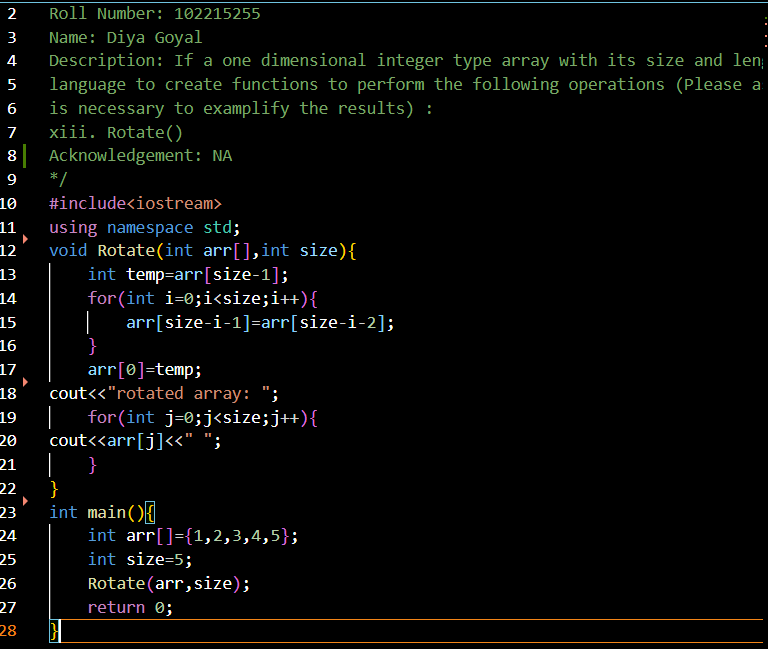


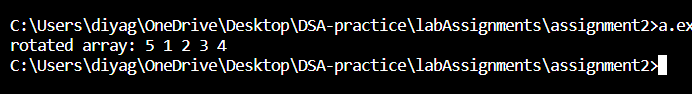
1. Shift()





1. Rotate()

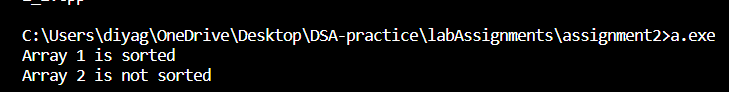




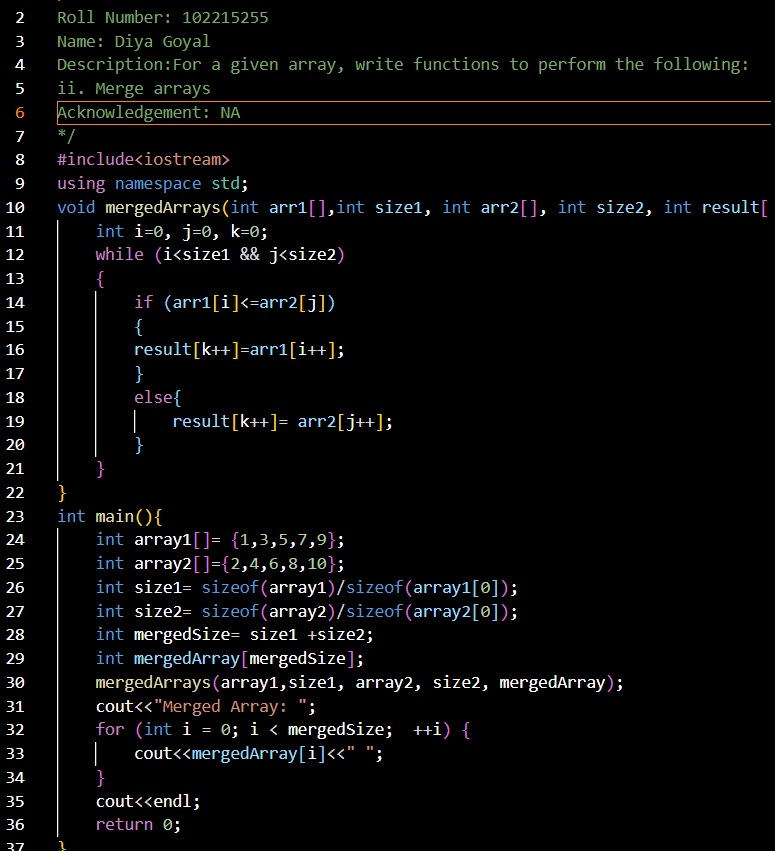
Q2. For a given array, write functions to perform the following:

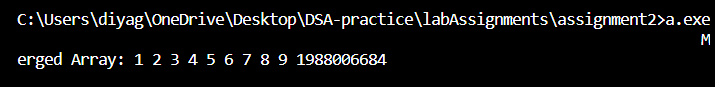
1. Check if an array is sorted

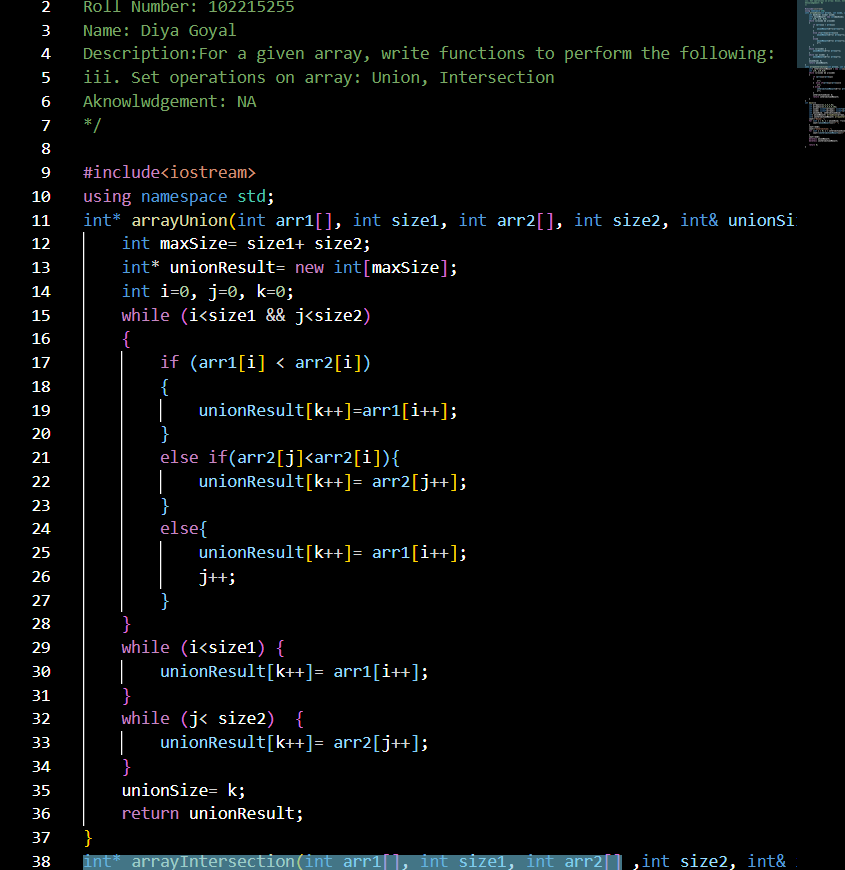




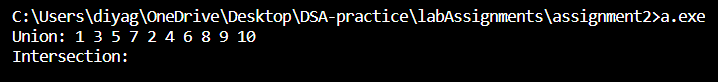
1. Merge arrays





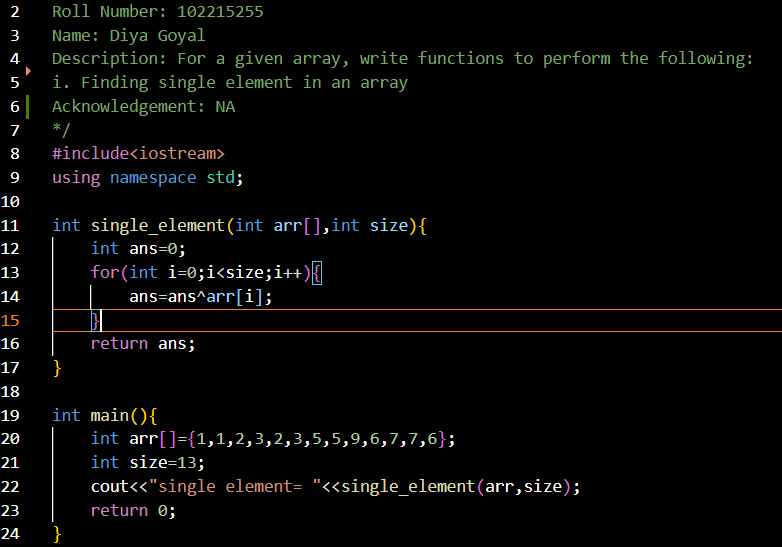
1. Set operations on array: Union, Intersection





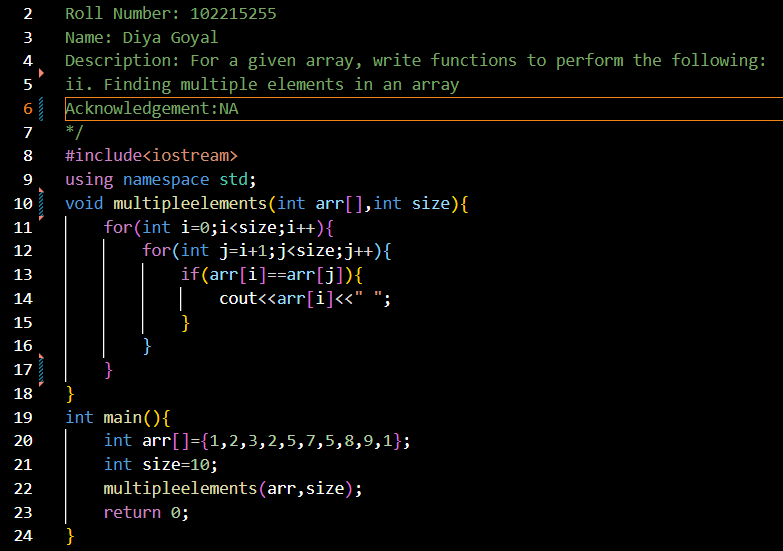
Q3. For a given array, write functions to perform the following:

1. Finding single element in an array



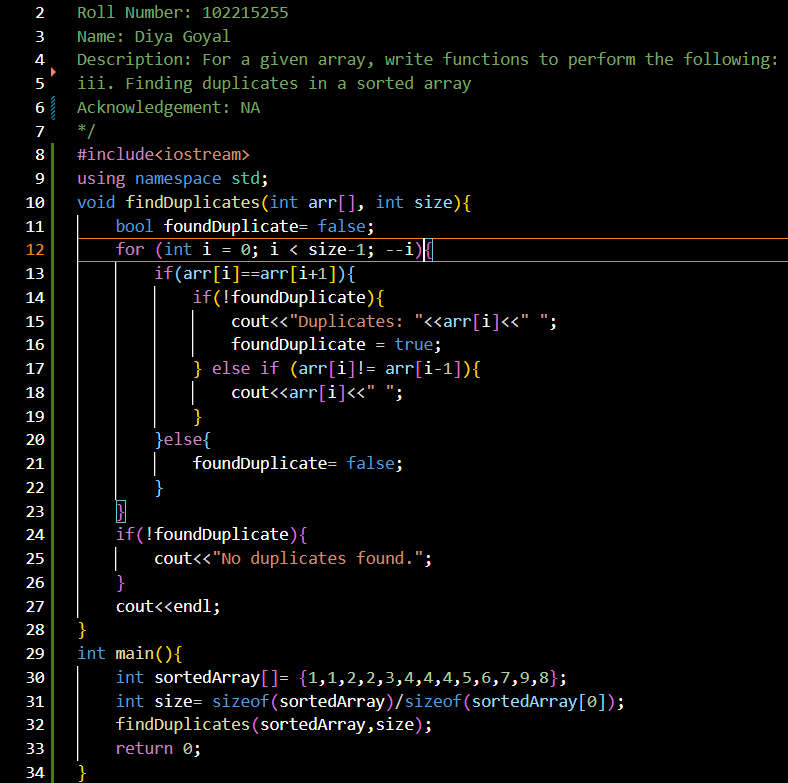


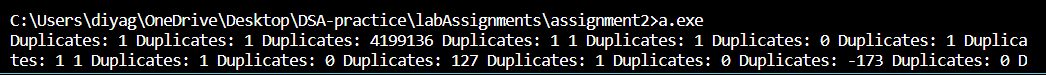
1. Finding multiple elements in an array



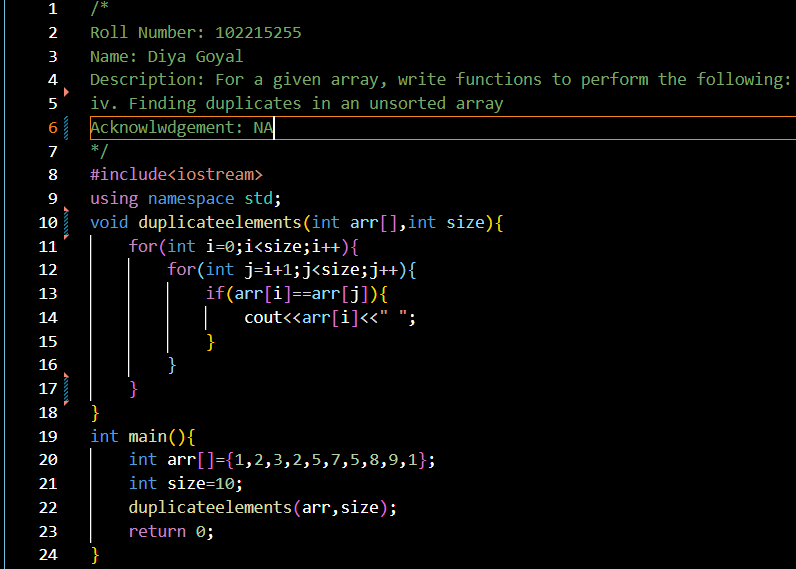


1. Finding duplicates in a sorted array



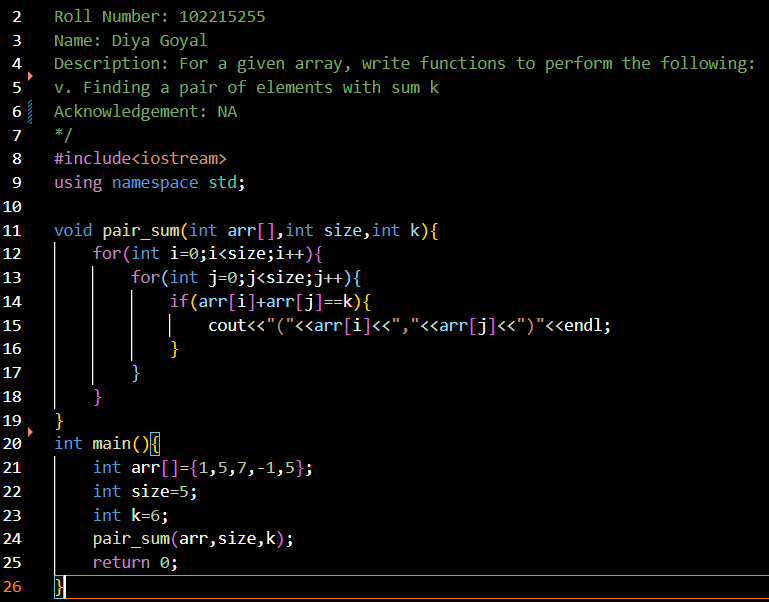


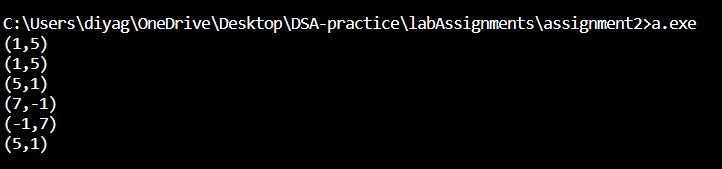
1. Finding duplicates in an unsorted array



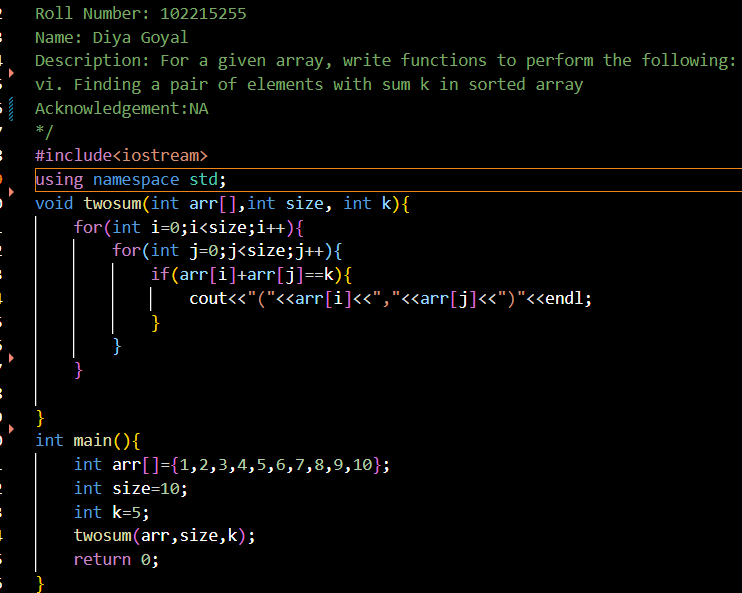


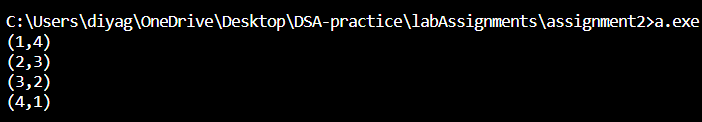
1. Finding a pair of elements with sum k



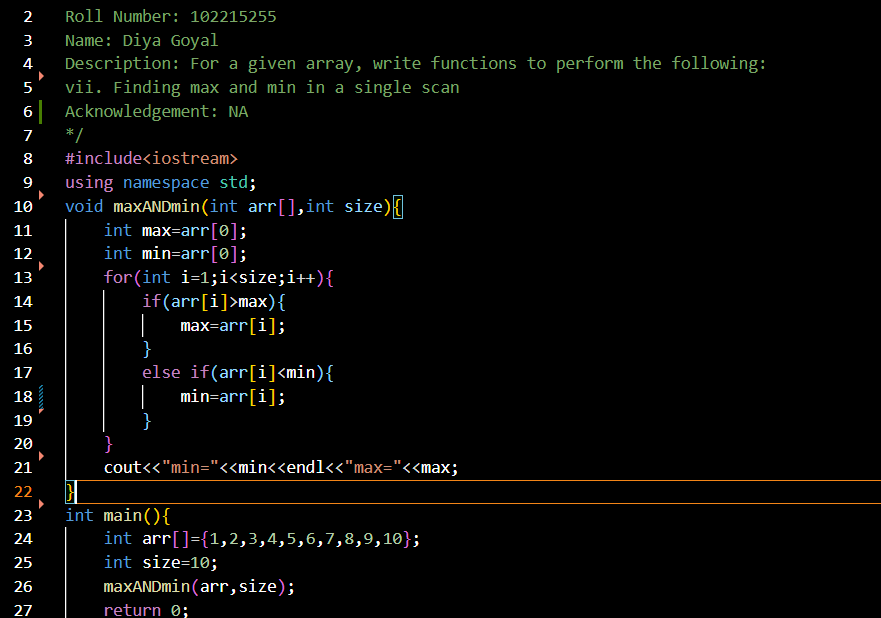


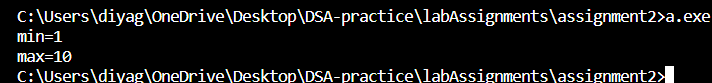
1. Finding a pair of elements with sum k in sorted array





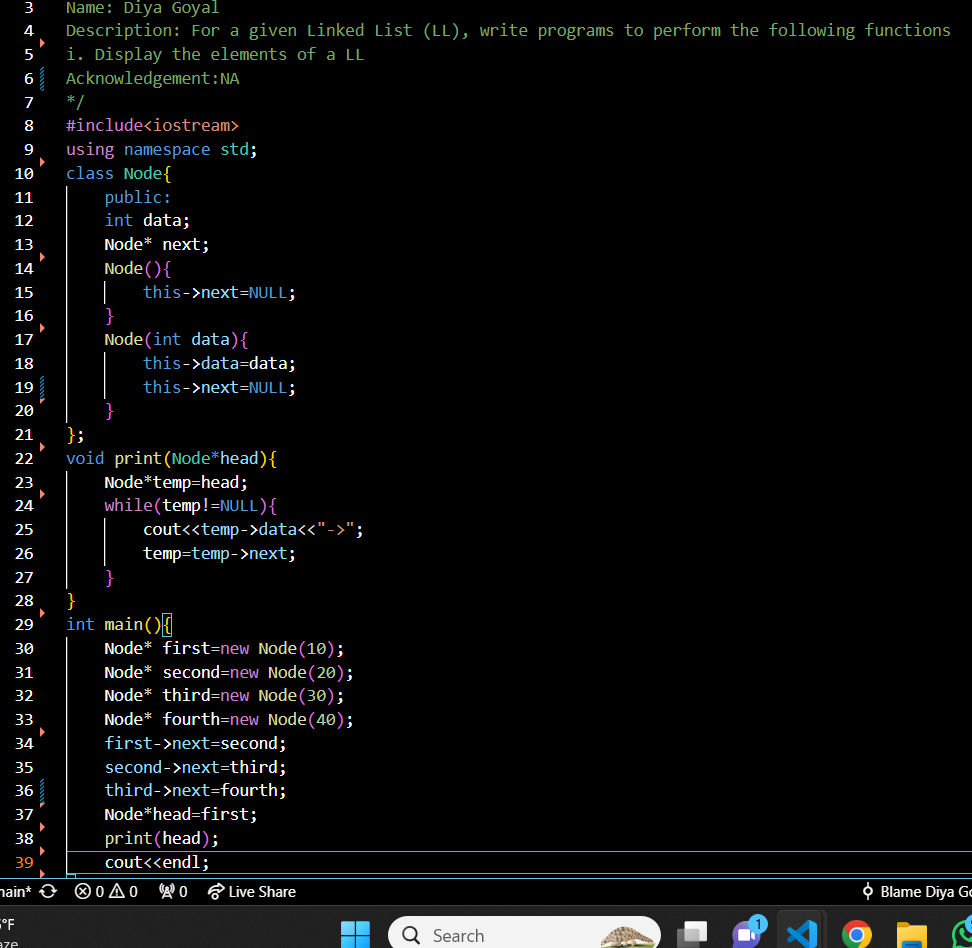
1. Finding max and min in a single scan

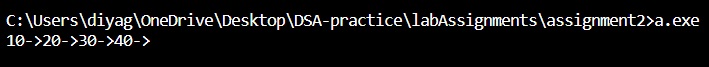




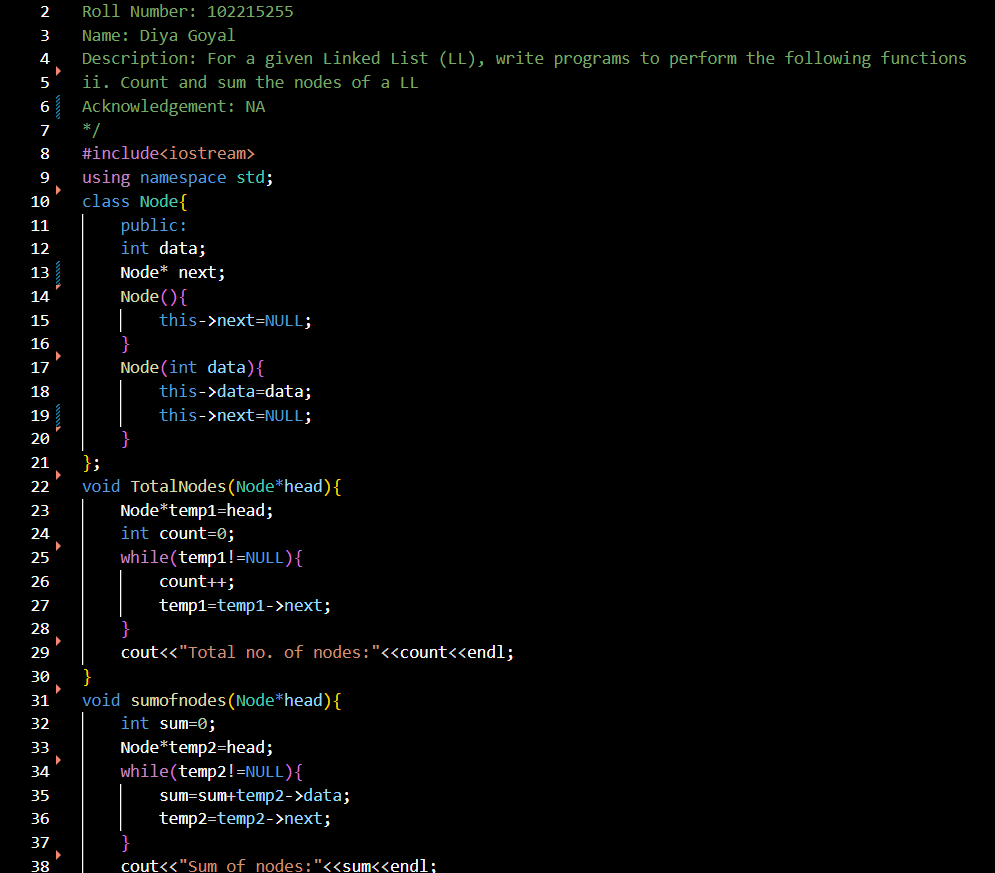
Q4. For a given Linked List (LL), write programs to perform the following functions

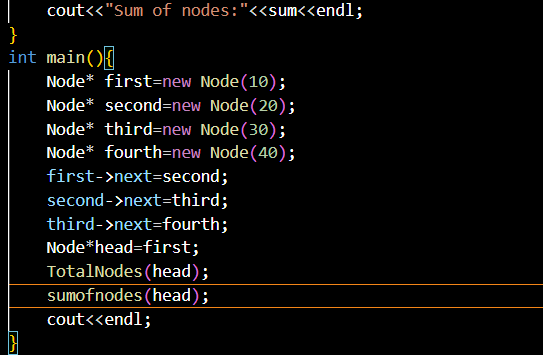
1. Display the elements of a LL

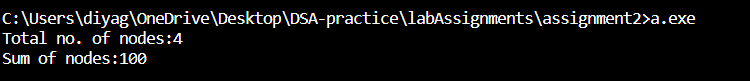




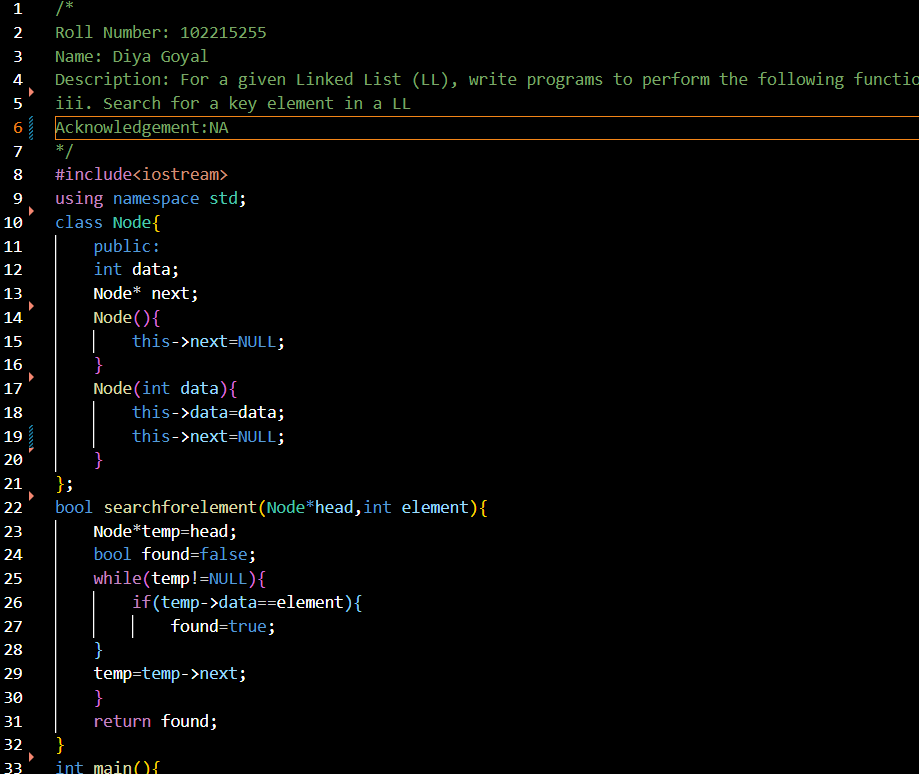
1. Count and sum the nodes of a LL

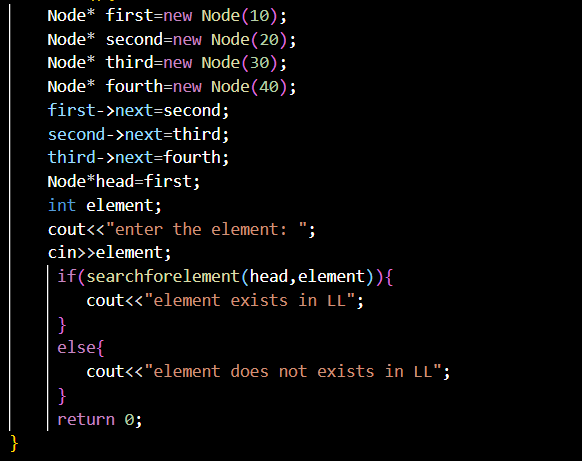


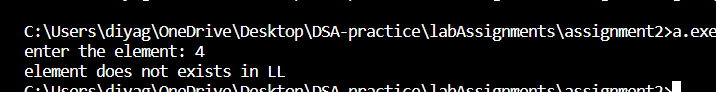




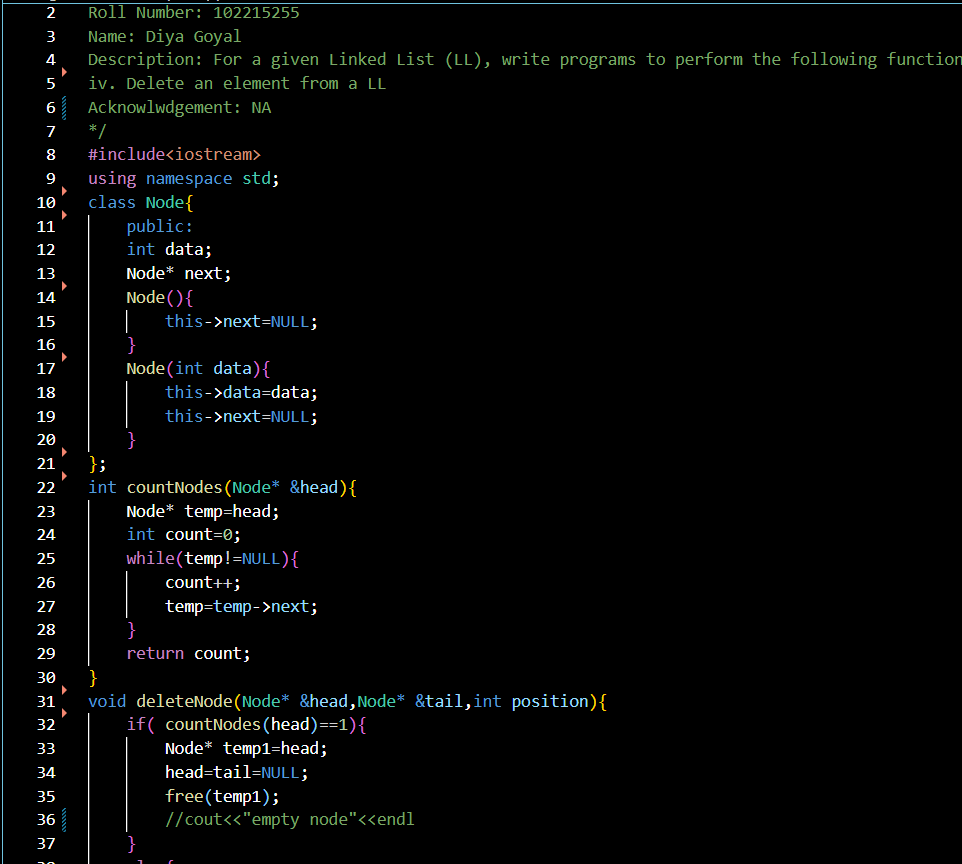
1. Search for a key element in a LL

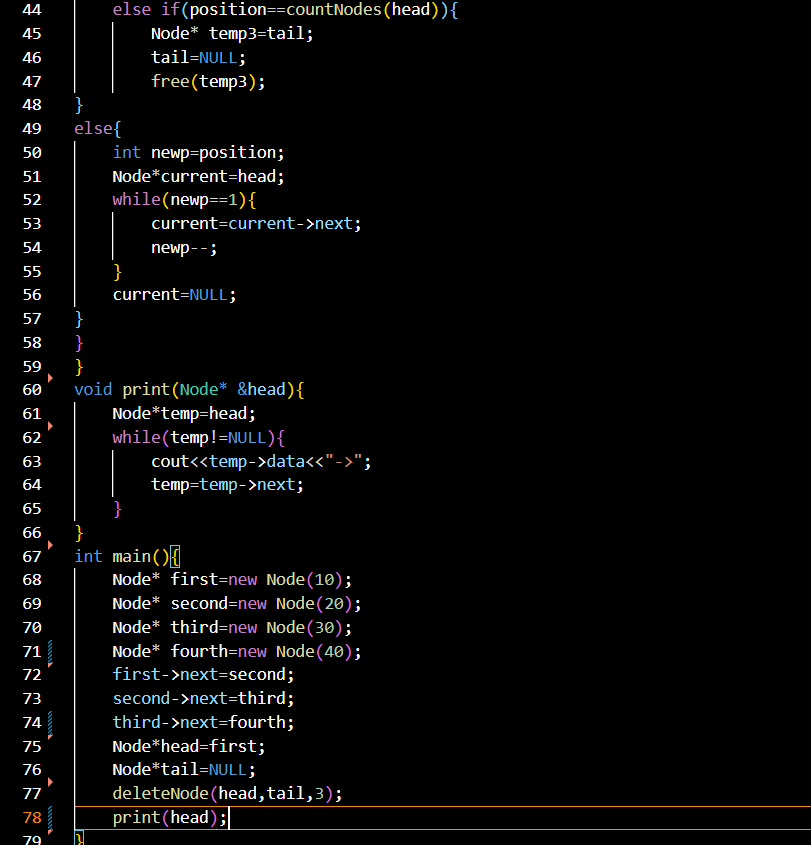


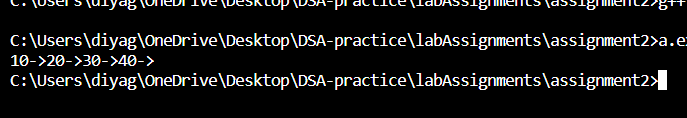




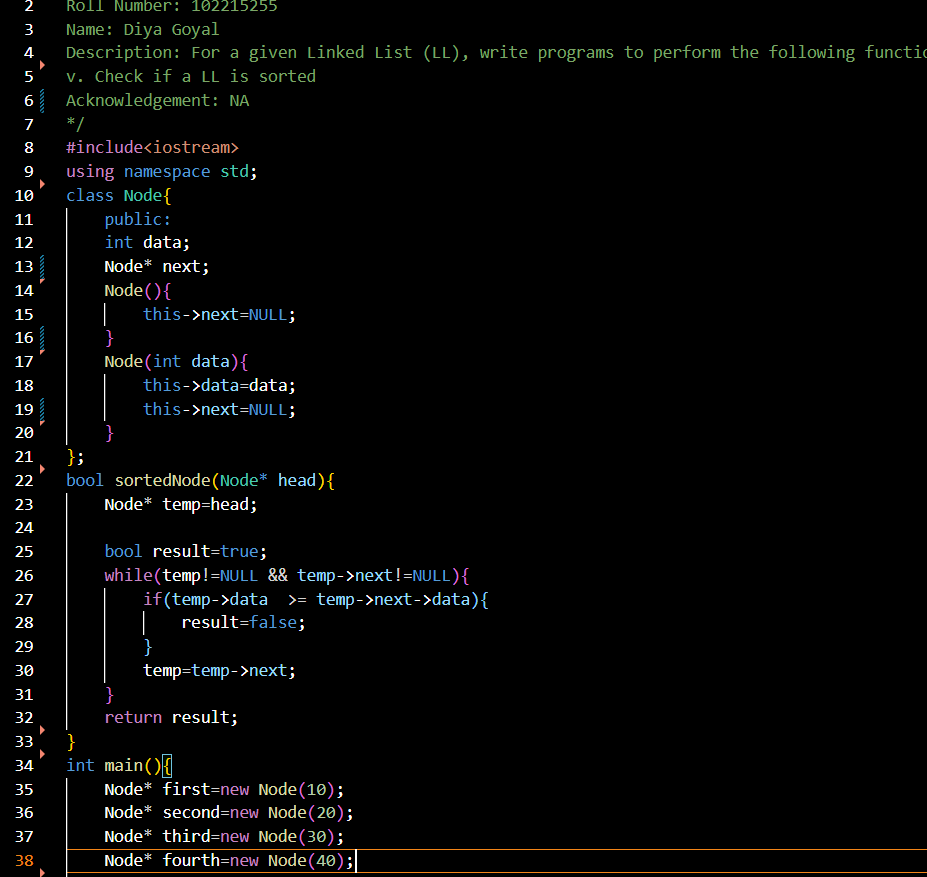
1. Delete an element from a LL

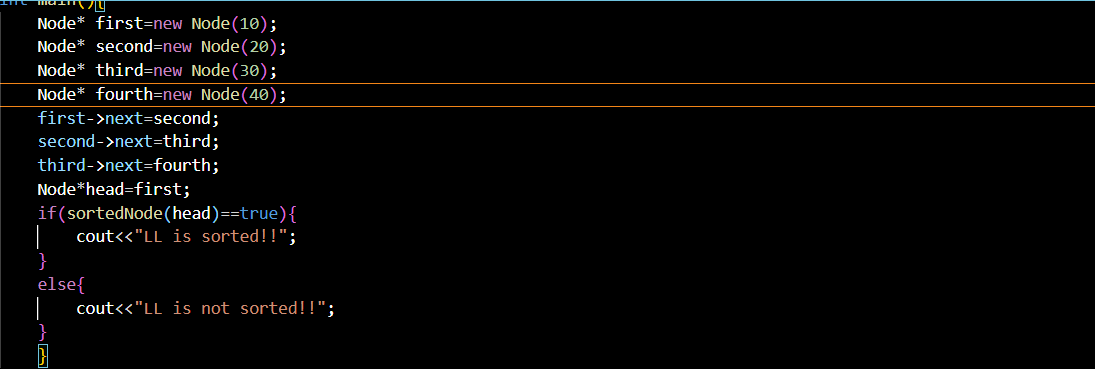


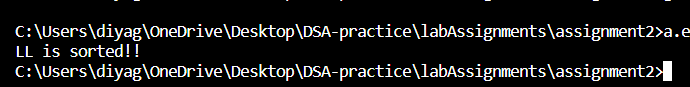




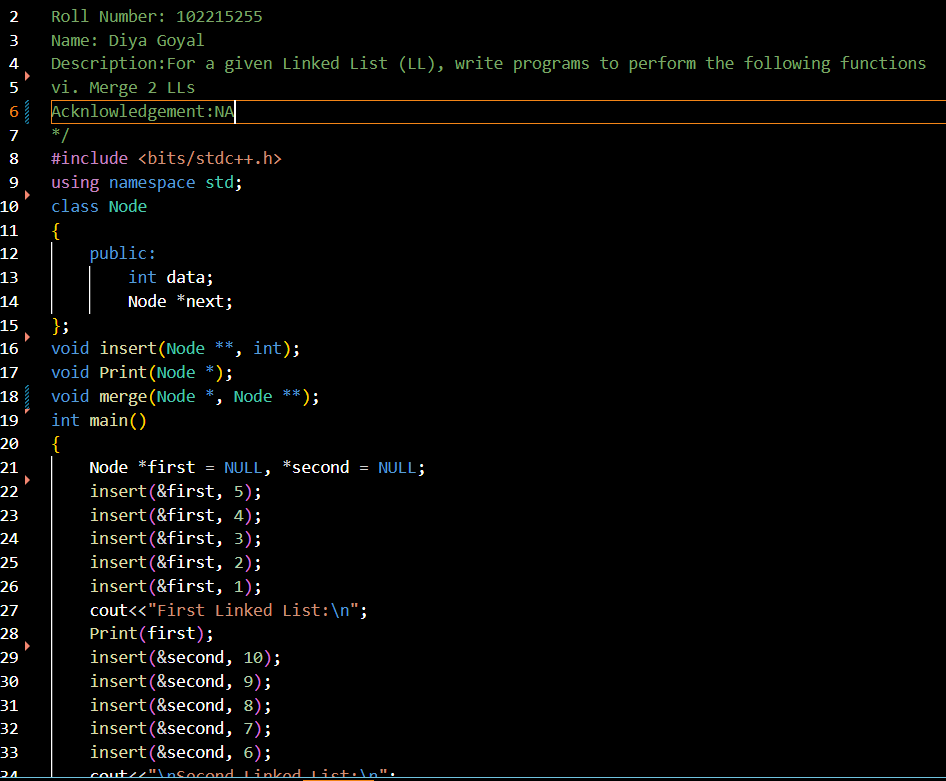
1. Check if a LL is sorted

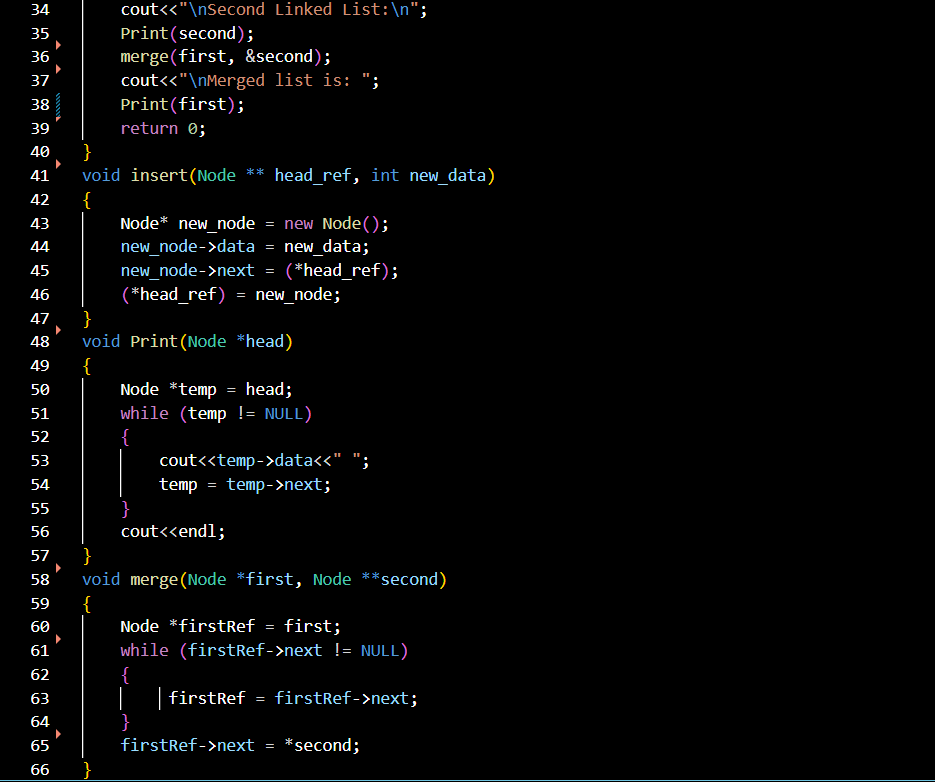


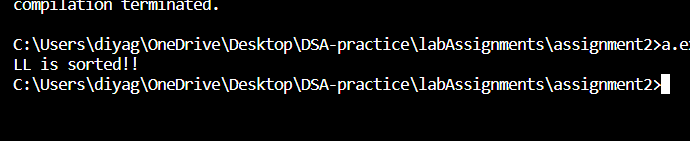


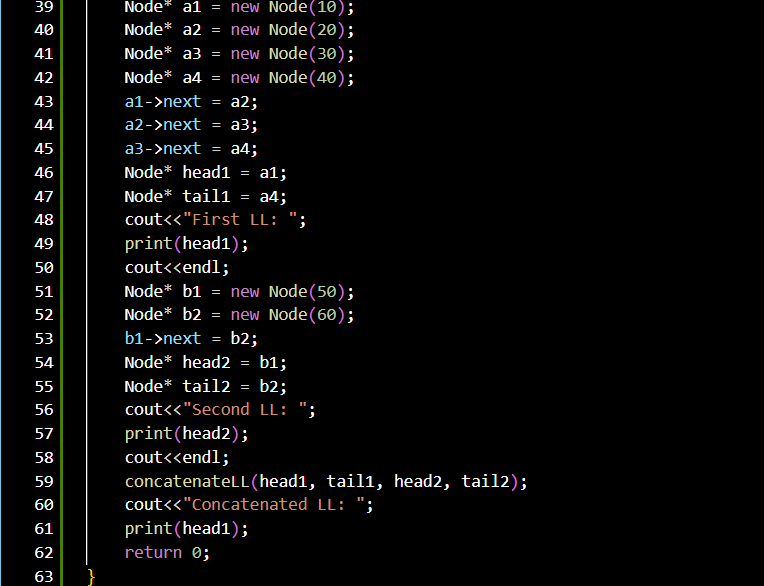
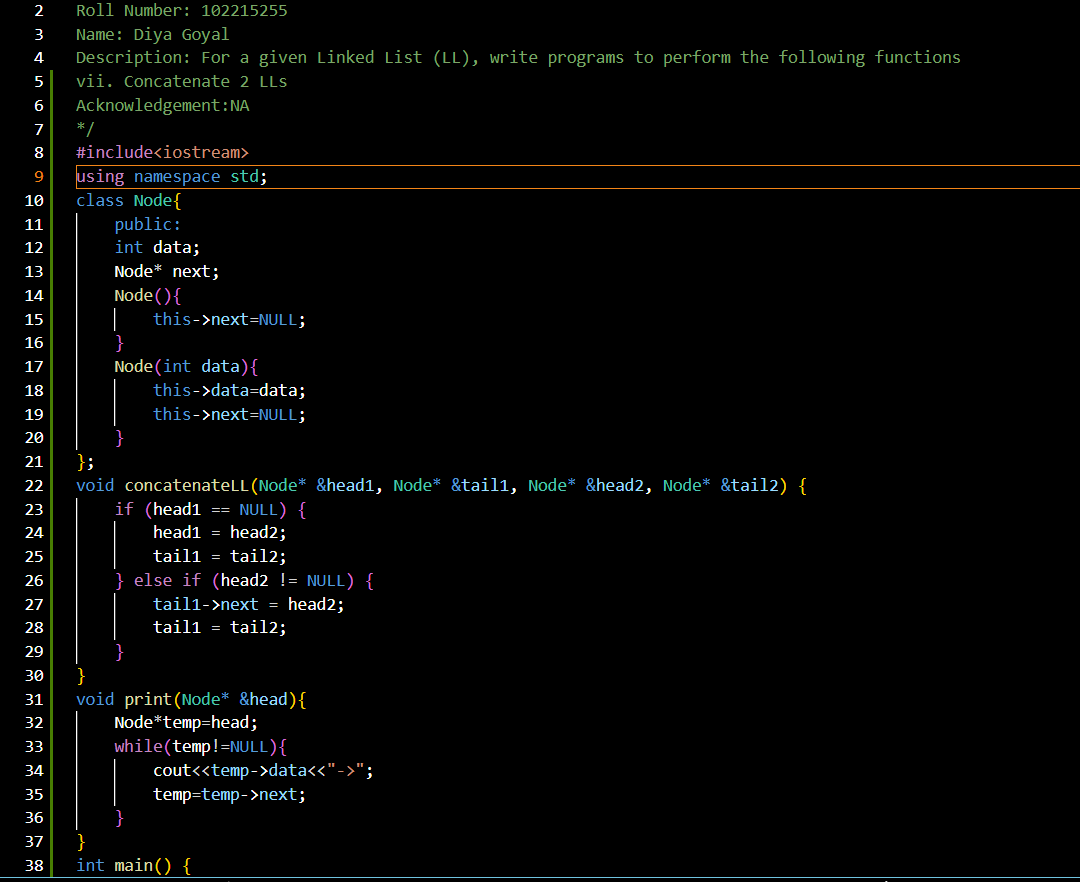


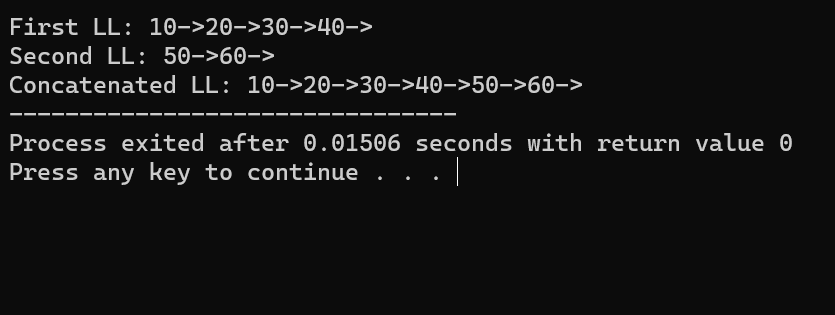
1. Merge 2 LLs

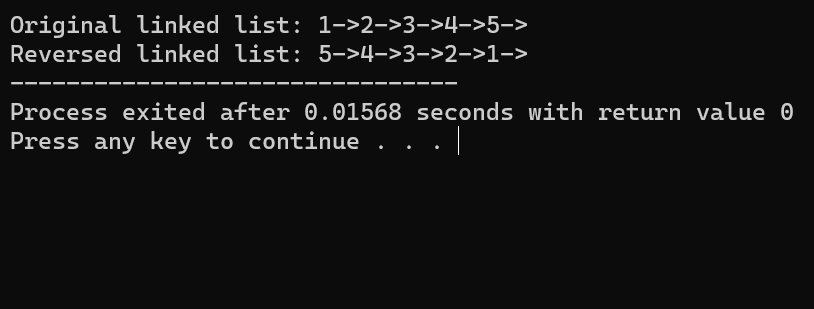
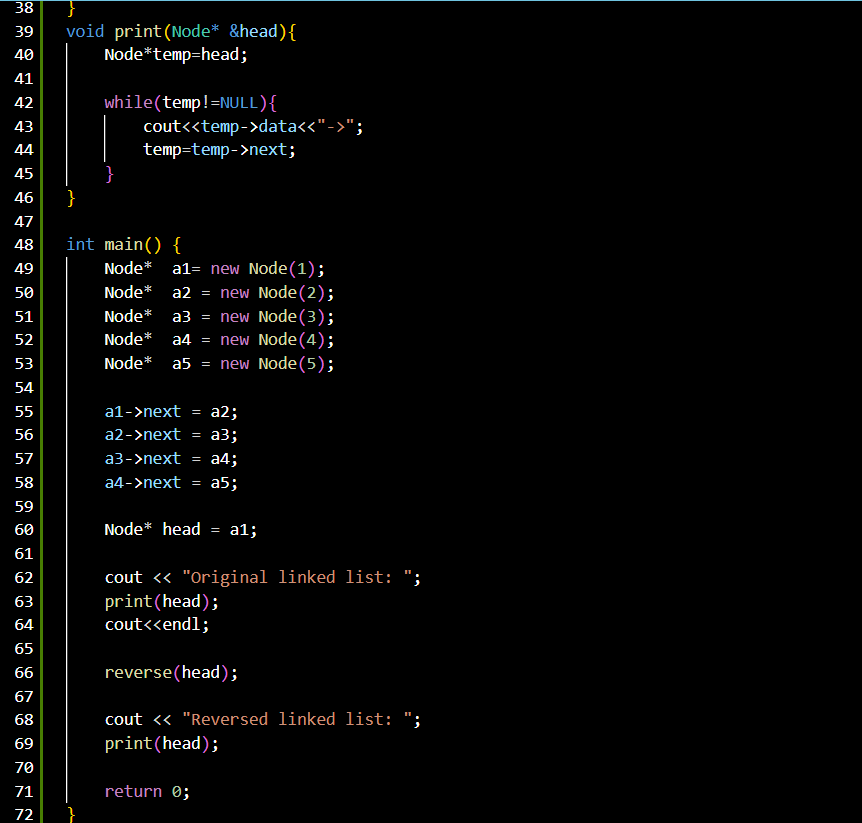
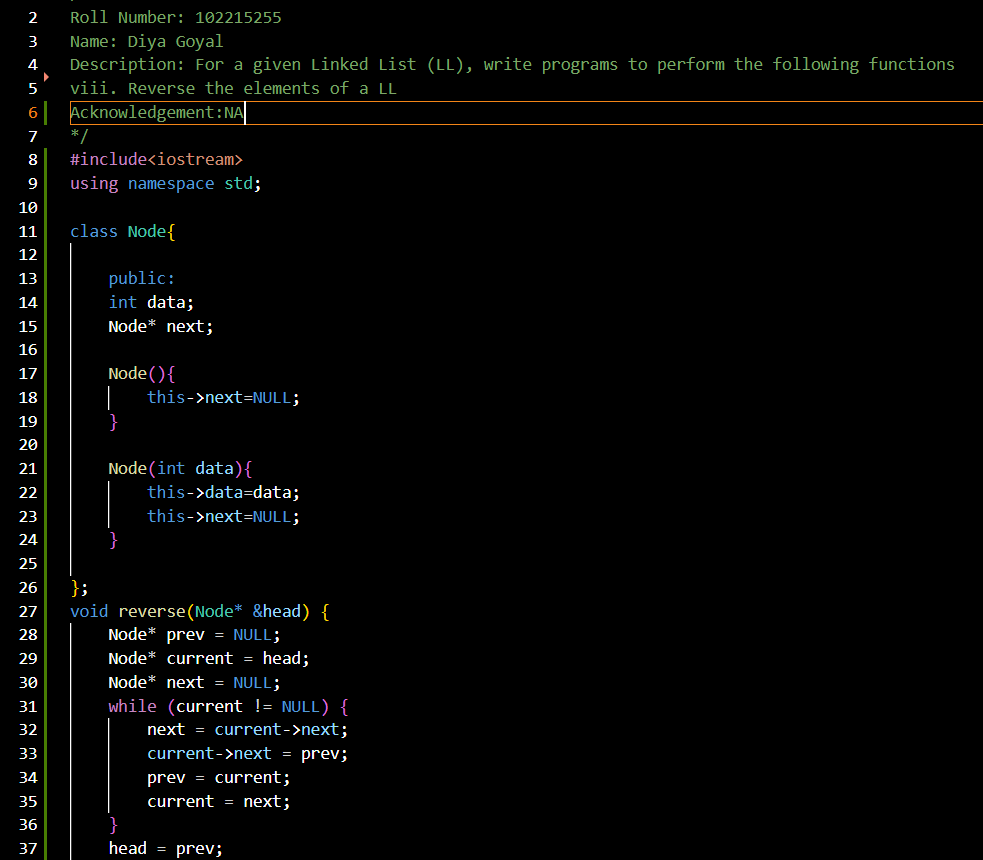


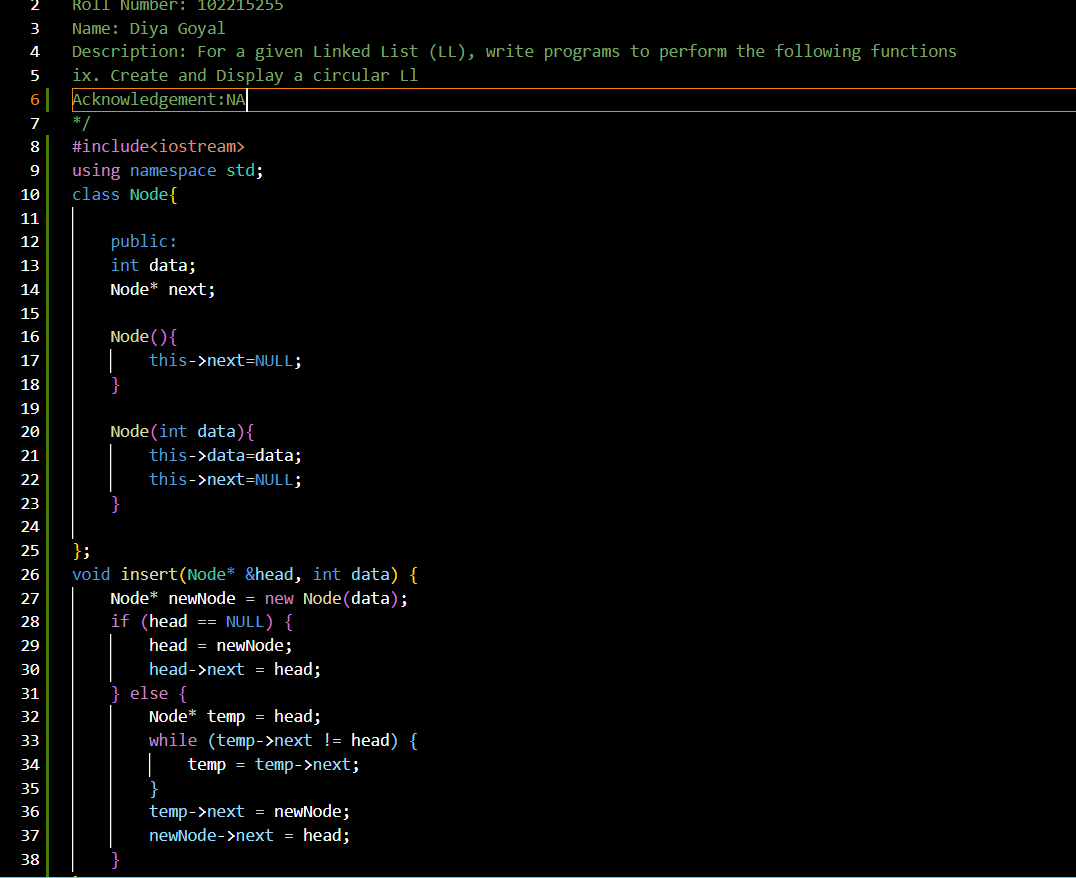


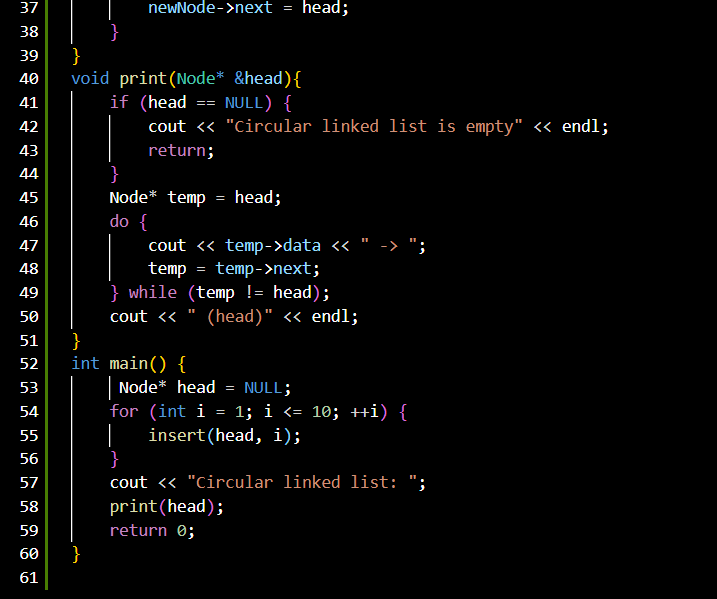


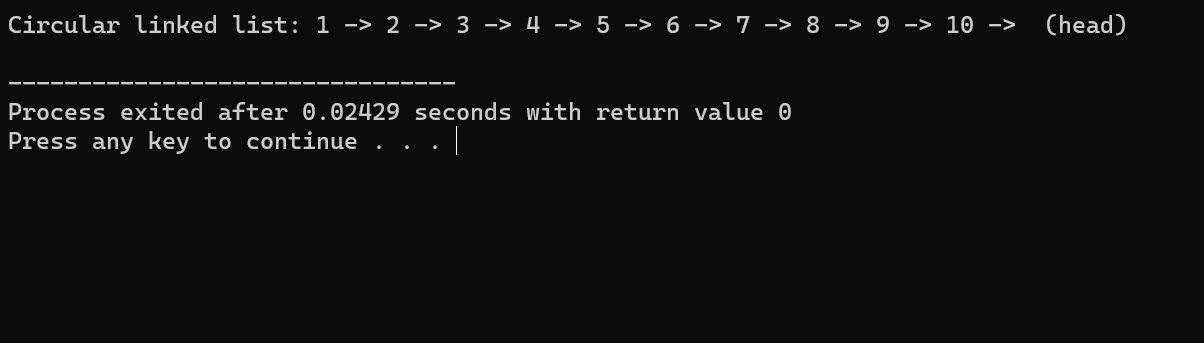
1. Concatenate 2 Ll



1. Reverse the elements of a LL
2. Create and Display a circular Ll







1. Create a doubly LL, insert in a doubly LL and reverse a doubly

