UNIVERSITY OF LAGOS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

B.Sc. (Hon.) Computer Engineering Degree Examinations

First Semester Test 2023/2024 Session

CPE 415/405: Computer Programming Languages

Instructions:

- **Time Allowed: 1 Hour 30 Minutes Attempt all questions**
- **Create a new project with the format: MatricNo_CPE415_QNumber and save all your source and header files into the appropriate folders.**
- **When you complete a question, zip the project folder and upload it to the server.**
- **Use only Visual Studio 2010 unless otherwise stated.**
- **Use multi-file technique.**

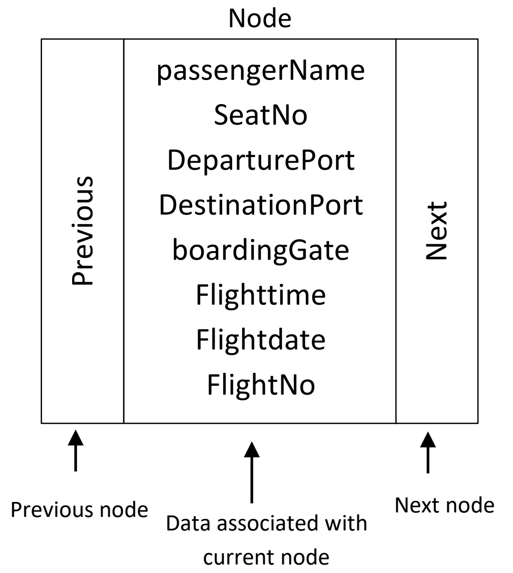
QUESTION 1

Write a complete C++ program that implements a boarding pass for ARIK Airline using a **Linked List** data structure which contains entries called **Node**. A Node has subentries: **passengerName, SeatNo, DeparturePort, DestinationPort, boardingGate, Flighttime, Flightdate, FlightNo** (all string data type) and two pointers (**previous and next**). When you enter a new item on a linked list, you allocate the new node and then set the pointers to the previous and next nodes. The node structure definition in C++ has the form shown in Fig. Q1. Copy and complete the structure by initializing the variables within the node structure appropriately and noting that the stored values of the linkedlist are provided by the user when you create a new node. Given that the public member functions of the LinkedList class are LinkedList(), ~LinkedList(), appendNode(), displayNodes(), destroyList().

Define completely the

(a) implementations of the member functions of the LinkedList class LinkedList(), ~LinkedList(), appendNode(), displayNodes(), destroyList() and its private data members in addition to the ones provided in Fig Q1.

(b) Write a main function as given in Table Q1 that displays the results on the screen as shown in Fig Q2.

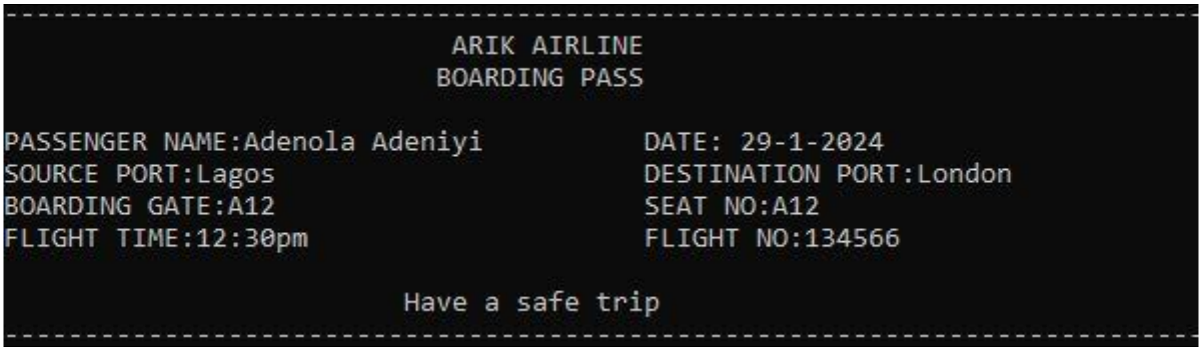


```
struct Node
{
  Node(string passengerName, string SeatNo,string
DeparturePort,string DestinationPort,string boardingGate,string
Flighttime,string Flightdate,string FlightNo)
            {
            .
            .
            }
            .
};
```

Fig. Q1

```
void main()
{
        LinkedList * list = new LinkedList();
        list->appendNode("Adenola Adeniyi","A12" ,"Lagos","London","A12","12:30pm","29-1-2024","134566");
        list->appendNode("Bukola Adenola ","A13" ,"Abuja","Paris","B45","13:30pm","30-1-2024","134577");
        list->displayNodes();
        delete list;
        system("pause");
}
```

Table Q1



```
--------------------------------------------------------------------------
                           ARIK AIRLINE
                          BOARDING PASS

PASSENGER NAME:Adenola Adeniyi          DATE: 29-1-2024
SOURCE PORT:Lagos                       DESTINATION PORT:London
BOARDING GATE:A12                       SEAT NO:A12
FLIGHT TIME:12:30pm                     FLIGHT NO:134566

                         Have a safe trip
--------------------------------------------------------------------------
```

Fig Q2

QUESTION 2

A pictorial description of the implementation of queue using an array is shown in Fig. Q2. Data organized by a queue may be stored in an array. The queue determines the array element that is at the front and back of the queue. The array is pictured as a block of elements while queue is pictured as a circle. The empty boxes are where values are stored in the queue, and the numbers correspond to the index of the array that is associated with the queue. To the right of the circle are three values. The front and back values store the index of the front and back of the queue. The size value is the number of elements in the queue. Implement the Queue class in C++ by declaring the appropriate data members as described above and depicted in Fig. Q2 and the following public member function: Queue(), ~Queue(), isFull(), isEmpty(), enqueue(), dequeue(). Hence, write the complete C++ definitions of the Queue(), enqueue(), dequeue() and utilizing the other member functions defined in Table Q2, set up a main function that:

(i) enqueue the values 10, 20 and 30 to the queue.

(ii) dequeue and display the values in the queue

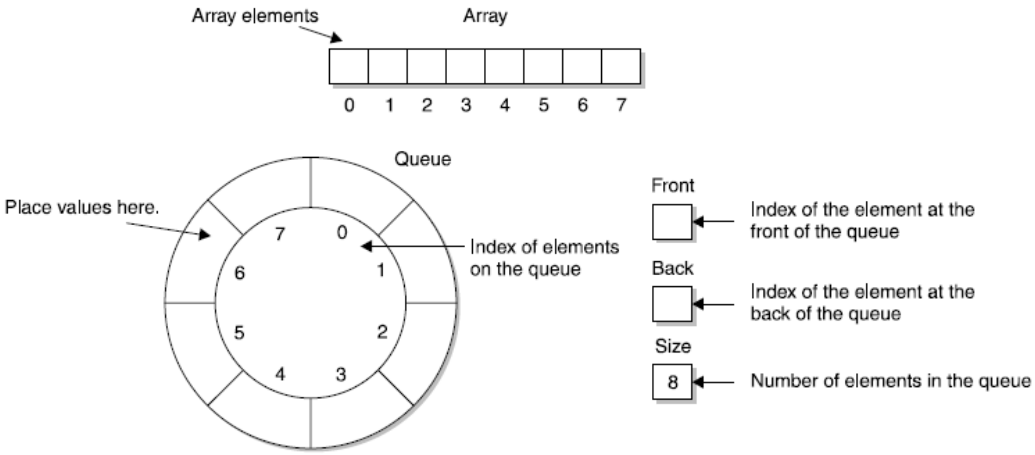

Array elements
Array
0
1
2
3
4
5
6
7
Queue
Place values here.
7
0
6
1
5
2
4
3
Index of elements on the queue
Front
Index of the element at the front of the queue
Back
Index of the element at the back of the queue
Size
8
Number of elements in the queue

Fig. Q2

Table Q2

Queue::~Queue()
{
    delete[] values;
}
	bool Queue::isFull()
	{
	    if((back+1) % size == front)
	    {
	        return true;
	    }
	    else
	    {
	        return false;
	    }
	}
		bool Queue::isEmpty()
		{
		    if(back == front)
		    {
		        return true;
		    }
		    else
		    {
		        return false;
		    }
		}