

**CS2001 Data Structures**

**Assignment #01**

| **Course**  **Instructor** | **Mr. Muhammad Usman Joyia** |
| --- | --- |
| **Session** | **Fall 2022** |
| **Section** | **BSC 3A – BSE 3A** |

**General Guidelines**

1. Write neat and clean code. Avoid any memory leaks and dangling pointers while implementing the scenarios required in this assignment.

2. You can lose the marks if conventions are not strictly followed.

3. Peer plagiarism and the late submissions are strictly not allowed. In case, zero marks will be awarded for whole assignment

4. Total Marks: 50

**Submission Guidelines**

1. You will upload the assignment on CLASSROOM in given timeline. 2. Don’t email your solution to instructor or TA for submission. Submit your assignment in given deadline said LMS.

3. You have already given one extra day for submission. No submission will be accepted later than said deadline.

4. Set file name as *Roll-no\_Section\_Assignment#*

**Deadline: September 27, 2022 4:00 PM**

**Assignment # 1**

**Task # 1**

Take inputs in a user defined array. After that if the input is even, place it at an even index and if the input is odd, place it an odd index. If the user puts in an even integer and all even indexes are occupied, ask user to enter an odd integer and vice versa. If all the even and odd entries are filled, then notify the user that the program has ended.

**Inputs:** 9 8 5 7 4 2 1

**Output:** 8 9 4 5 2 7 1

**Index:** 0 1 2 3 4 5 6 7

**Task # 2**

Create a user defined array and then take inputs. Print all the Divisors of every number user gave as input.

**Inputs:** Size: 3

**Elements:** 4 5 6

**Output:**

4: 1 2

5: 1

6: 1 2 3

**Task # 3**

Create a menu driven program implementing the following class.

LinkedList

{

public:

Node\* head;

private:

void insertNodeAtBeginning(int data);

void insertNodeInMiddle(int data, int key); //will search for key and insert node after the node where a node’s data==key

void insertNodeAtEnd(int data);

bool deleteFirstNode();

bool deleteNode(int key);

bool deleteLastNode();

void display();

void displayReverse();

bool search(int data);

Node\* sortList(Node\*)

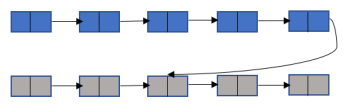
Node\* mergeList(Node\*, Node\*)

};

**Note:** *you are not allowed to use any built-in libraries.*

**Task # 4**

Your task is to take two singly linked lists let suppose A and B. The last index of A is connected to 3rd index of the B. Find the common elements in both linked lists. For more idea see the figure below.



**Task # 5**

Create a menu driven program implementing the following class.

DoublyLinkedList

{

public:

Node\* head;

Node\* tail;

private:

void insertNodeAtBeginning(int data);

void insertNodeInMiddle(int data, int key); //will search for key and insert node after the node where a node’s data==key

void insertNodeAtEnd(int data);

bool deleteFirstNode();

bool delete\Node(int key);

bool deleteLastNode();

void display();

void displayReverse();

bool search(int data);

};

*Note: you are not allowed to use any built-in libraries.*

**Task # 6**

Write a program that prompts the user to input a string and then outputs the string in the pig Latin form. Input the string in a **doubly circular link-list**. The rules for converting a string into pig Latin form are as follows:

If the string begins with a vowel, add the string **"-way"** at the end of the string. Forexample, the pig Latin form of the string **"eye"** is **"eye-way"**.

If the string does not begin with a vowel, first add **"-"** at the end of the string. Then rotate the string one character at a time; that is, move the first character of the string tothe end of the string until the first character of the string becomes a vowel. Then add the string **"ay"** at the end. For example, the pig Latin form of the string **"There"** is **"ere-Thay**

Strings such as **"by"** contain no vowels. In cases like this, the letter **y** can be considered a vowel. So, for this program the vowels are **a, e, i, o, u, y, A, E, I, O, U,and Y**. Therefore, the pig Latin form of **"by"** is **"y-bay"**.

Strings such as **"1234"** contain no vowels. The pig Latin form of the string **"1234"** is**"1234- way"**. That is, the pig Latin form of a string that has no vowels in it is the string followed by the string **"-way"**.

*Note: Your program must store the characters of a string into a linked list and use thefunction rotate, to rotate the string.*

**Task # 7:**

Round-Robin algorithm is an algorithm used in CPU Scheduling. In this algorithm, all processes take turns and run for a specific interval, until they are all completely executed.You are required to make a process that will consist of two things,

ID *(must be unique)*

Execution time

Each process will take a certain amount of time to execute completely, this time will bestored in the “execution time” variable.

You are required to implement a **circular linked list** that will store *user-entered* number of processes (say *n*). You will also input a *time slice* from the user, this will be the time every process will get in one turn.

You will then create a function that will execute every process in the circular linked list using Round-Robin algorithm and store the total time taken by each process to execute. Your final output will be the time required by each process to completely execute. **Sample Output:**

Enter number of processes: 4

Enter Process #1 Execution Time: 2

Enter Process #2 Execution Time: 3

Enter Process #3 Execution Time: 9

Enter Process #4 Execution Time: 5

Enter Time Slice: 3

Processes in Linked List: 1 2 3 4 - Process #1 executed for 2 seconds.

Process #1 has been executed. Time Taken: 2

Processes in Linked List: 2 3 4 - Process #2 executed for 3 seconds.

Process #2 has been executed. Time Taken: 5

Processes in Linked List: 3 4 - Process #3 executed for 3 seconds.

Processes in Linked List: 3 4 - Process #4 executed for 3 seconds.

Processes in Linked List: 3 4 - Process #3 executed for 3 seconds.

Processes in Linked List: 3 4 - Process #4 executed for 2 seconds.

Process #4 has been executed. Time Taken: 16

Processes in Linked List: 3 - Process #3 executed for 3 seconds.

Process #3 has been executed. Time Taken: 19

Execution Completed