Coding Exercise: 2- Going in Circles [1]

Chapter 2: Arrays and Linked Lists[2]

Jarrian Vince G. Gojar[3]

[1] A coding exercise for Chapter 2 of the Study Guide on the course Data Structures and Algorithms.

[2] This chapter introduces the concepts of arrays and linked lists.

[3] https://github.com/godkingjay

Sorsogon State University - Bulan Campus

Coding Exercises

> *"C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do, it blows away your whole leg."*
>
> – Bjarne Stroustrup

Instructions: Write a program that solves the following problems. Submit your code to the Google Drive folder provided by the instructor.

1. **Reverse an Array:** Write a C++ program to reverse an array of integers. The program should take the size of the array and the elements of the array as input and output the reversed array.

   (a) Declare an array of integers with a fixed size.

   ```
   int arr[6];
   ```

   (b) Initialize the array with the input elements.

   ```
   arr = {19, 10, 8, 17, 9, 15};
   ```

   (c) Reverse the array using a loop.

Original Array	19	10	8	17	9	15

Reversed Array	15	9	17	8	10	19

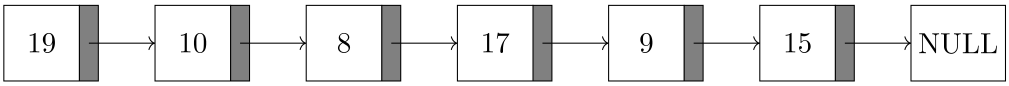
   (d) Print the reversed array to the console.

   ```
   Reversed Array: 15, 9, 17, 8, 10, 19
   ```

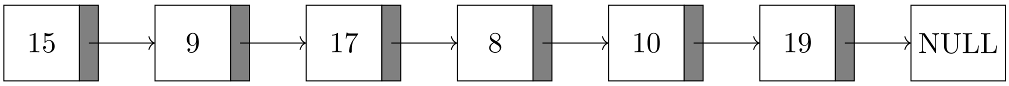
   (e) Determine the **time complexity** and **space complexity** of the program.

2. **Reverse a Linked List:** Write a C++ program to reverse a singly linked list. The program should take the elements of the linked list as input and output the reversed linked list.

   (a) Create a singly linked list with nodes containing the elements



   (b) Reverse the linked list using a loop.



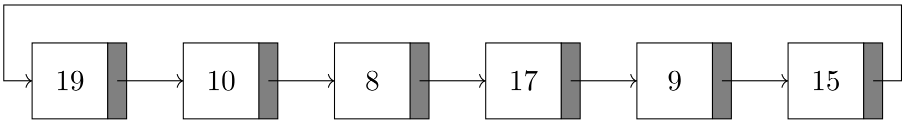
(c) Print the reversed linked list to the console.

Reversed Linked List: 15, 9, 17, 8, 10, 19

(d) Determine the **time complexity** and **space complexity** of the program.

3. **Search a Circular Linked List:** Write a C++ program to search for an element in a circular linked list. The program should take the elements of the circular linked list as input and output the position of the element in the list. If the current node's next pointer points to the head node, the list is circular. If the element is not found in the list, output "Element not found." Else, output the position of the element in the list.

(a) Create a circular linked list with nodes containing the elements



(b) Using *cin*, take the input element to search for

Search Element: 21

(c) Search for the element in the circular linked list

Output: Element not found.

(d) Determine the **time complexity** and **space complexity** of the program.

Submission of Coding Exercises

Instructions:

1. Go to the Google Drive folder provided by the instructor:

   BSIT 2-4:
   https://drive.google.com/drive/folders/1uc3ehhK4Mv84KXPe8oV3l3AGI6czVhBx?usp=sharing

   BSIT 2-5:
   https://drive.google.com/drive/folders/1eIkUp3t2cAKIpd9KZGbQlZEGU516S_sE?usp=sharing

2. Inside the folder, create another folder for your group with the following format:

   Group Number - LastName1_FirstName1, LastName2_FirstName2
   Example: **Group 1 - Doe_John, Smith_Jane**

3. Inside the sub-folder, create another folder with the name:

   Chapter 2- Coding Exercise 2- Going in Circles

4. Inside the folder, upload the file of your submission.

Fill in the template provided in the following link and upload it inside the folder: https://docs.google.com/document/d/1zZf3W0Hj6NfCGU7sKAaz_cztSER5e7Vx/edit?usp=sharing&ouid=112709378145681657270&rtpof=true&sd=true

5. The activity must be submitted **on or before October 18, 2024**.

6. Late submissions will not be accepted.