

Student Name: \_\_\_\_\_  
 Program: BS (CS/SE)  
 Semester: Spring 2023  
 Time Allowed : 1 hour  
 Course: CS2001 - Data Structures

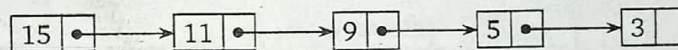
Roll No: \_\_\_\_\_  
 Examination: Sessional-I  
 Marks: 45, Weightage: 15  
 Date: 13-03-2023  
 Instructor: Waqas Ali

**NOTE:** Attempt all questions.

1. Suppose that an array-based list of integers exists in memory i.e., the ArrayList class is already present. Write a function (named **resizeList**) that resizes the list to a new size, without losing any elements in the list. The new size should be the parameter of the function. If the new size is smaller than the current size, reduce the list to the new size. If the new size is larger than the current size, add the elements of the original list to the start of the new list. Your function should return **true** if the resize operation was successful, and **false** otherwise (e.g., if the new size is less than zero). [CLO 3, 10 points]

**Note:** Use of index is not allowed, you can only use pointers (ZERO credit for index based solution).

2. Assume you are given a linked list of sorted integers. Create a function (called **sortedInsert**, has one integer parameter) that inserts a value into the linked list at its proper position while keeping the list sorted. The list can be in ascending or descending order, and your function must work for both. A separate function can be written to determine whether the linked list is sorted ascending or descending order. Consider that the linked list class is already present. [CLO 3, 10 points]

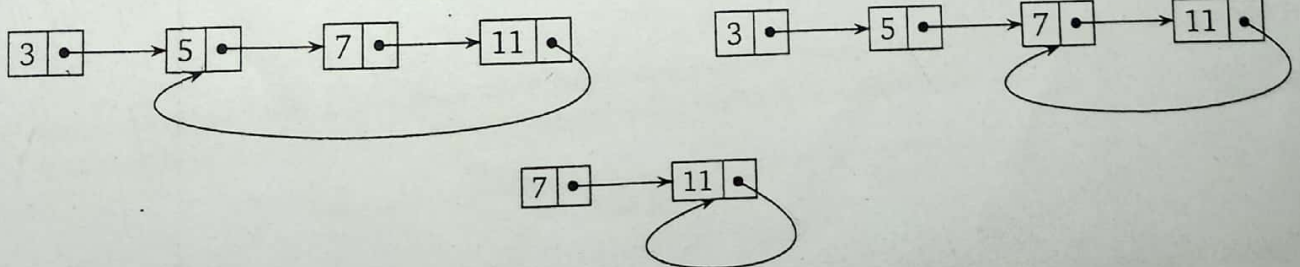


If we want to insert 8 in to the above linked list, the correct position of 8 is between 9 and 5.

**Note:** You are not allowed to sort the list after insertion, you have to insert the value in its proper position.

3. Consider that the same operation (**sortedInsert** in question 2) needs to be performed on an array based list and the fixed size of the list is not an issue. Comment on whether a linked list or an array-based list is better in this situation. Show, using an example, which one will require the least number of operations. [CLO 2, 7 points]
4. In an array based implementation of a stack, is it a good idea to insert (push) and delete (pop) only at index 0 of the array? Provide arguments and use an example to demonstrate (diagrammatically) whether it is a good or a bad idea. [CLO 2, 8 points]
5. Create a function (named **hasLoop**) that determines whether or not a circular linked list contains a loop. If the circular linked list contains a loop, the function should return true; otherwise, it should return false. Every circular linked list contains a loop, which means that the last node points to the first node (head). This function should not detect such a loop; instead, it should detect whether the final node's next pointer is pointing to the final node-itself or any other node other than the head node. [CLO 4, 10 points]

Following are examples of a loop in a circular linked list.



\*\*\*\*\* The End \*\*\*\*\*



Student Name: \_\_\_\_\_  
Program: BS (CS/SE)  
Semester: Spring 2022  
Time Allowed : 1 hour  
Course: CS2001 - Data Structures

Examination: Sessional-II  
Marks: 25, Weightage: 15  
Date: 10-05-2022  
Instructor: Waqas Ali

**NOTE:** Attempt all questions.

1. Write a function **reverseOdd** that takes a queue of integers as a parameter, and the function modifies that queue, reversing the order of the odd integers in the queue while leaving the even integers in place. For example given this queue (rear to front):

14	13	17	8	4	10	11	4	15	18	19
----	----	----	---	---	----	----	---	----	----	----

calling the function would change it to:

14	19	15	8	4	10	11	4	17	18	13
----	----	----	---	---	----	----	---	----	----	----

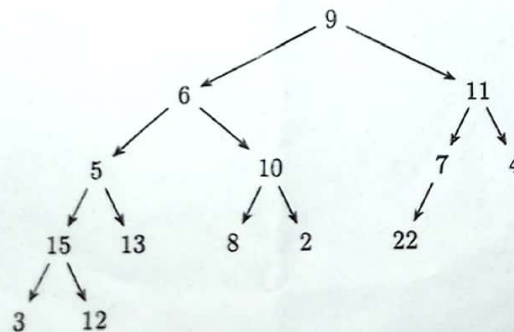
**Note:** You are not allowed to create a new queue to store the result (the items should be reordered in the same queue) and are only allowed to perform the enqueue and dequeue operations on the queue.

2. Write recursive functions for the following (ZERO credit for non-recursive solution):

- To check whether a given binary tree is a binary search tree or not (the function should return true if it is, it should return false otherwise).
- To count the number of nodes in a binary search tree.

3. Assume that a binary tree with integer info exists in memory. Write a recursive function that **counts** the number of nodes whose info is equal to the sum of its children's info.

For Example, in the following binary tree the nodes with info 10, 11 and 15 fulfill the requirement mentioned in the question:



4. Write a recursive function that finds the **balance** of a node in a binary search tree. The info of the node whose balance is to be found should be passed to the function as a parameter.

The End

Student Name: \_\_\_\_\_  
Program: BS (CS/SE)  
Semester: Spring 2022  
Time Allowed : 1 hour  
Course: CS2001 - Data Structures

Roll No: \_\_\_\_\_  
Examination: Sessional-II  
Marks: 25, Weightage: 15  
Date: 10-05-2022  
Instructor: Waqas Ali

**NOTE:** Attempt all questions.

1. Write a function **reverseOdd** that takes a queue of integers as a parameter, and the function modifies that queue, reversing the order of the odd integers in the queue while leaving the even integers in place. For example given this queue (rear to front):

[7]

14	13	17	8	4	10	11	4	15	18	19
----	----	----	---	---	----	----	---	----	----	----

calling the function would change it to:

14	19	15	8	4	10	11	4	17	18	13
----	----	----	---	---	----	----	---	----	----	----

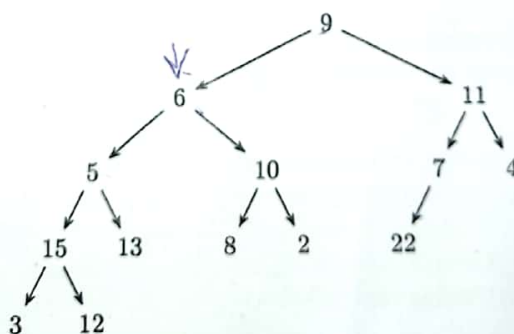
**Note:** You are not allowed to create a new queue to store the result (the items should be reordered in the same queue) and are only allowed to perform the enqueue and dequeue operations on the queue.

2. Write recursive functions for the following (ZERO credit for non-recursive solution):
- To check whether a given binary tree is a binary search tree or not (the function should return true if it is, it should return false otherwise).
  - To count the number of nodes in a binary search tree.
3. Assume that a binary tree with integer info exists in memory. Write a recursive function that counts the number of nodes whose info is equal to the sum of its children's info.

[4 + 3]

[7]

For Example, in the following binary tree the nodes with info 10, 11 and 15 fulfill the requirement mentioned in the question:



4. Write a recursive function that finds the **balance** of a node in a binary search tree. The info of the node whose balance is to be found should be passed to the function as a parameter.

[4]

The End

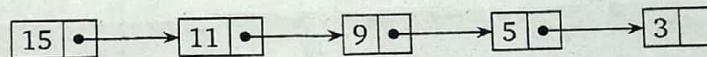


Student Name: \_\_\_\_\_  
Program: BS (CS/SE)  
Semester: Spring 2023  
Time Allowed : 1 hour  
Course: CS2001 - Data Structures

Roll No: \_\_\_\_\_  
Examination: Sessional-I  
Marks: 45, Weightage: 15  
Date: 13-03-2023  
Instructor: Waqas Ali

**NOTE:** Attempt all questions.

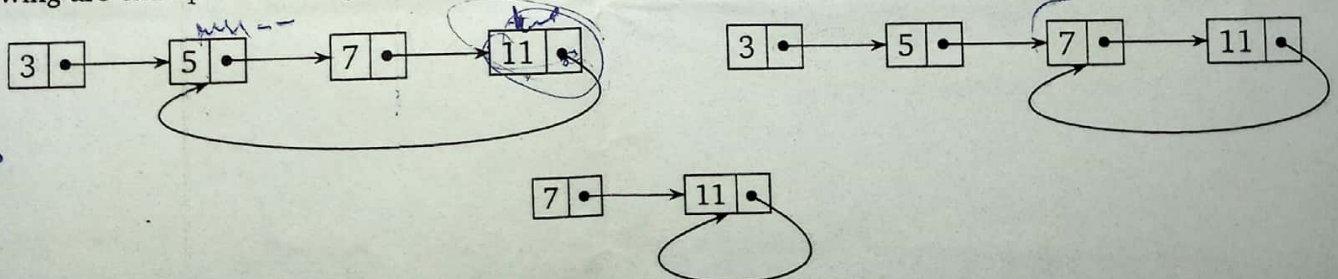
1. Suppose that an array-based list of integers exists in memory i.e., the ArrayList class is already present. Write a function (named **resizeList**) that resizes the list to a new size, without losing any elements in the list. The new size should be the parameter of the function. If the new size is smaller than the current size, reduce the list to the new size. If the new size is larger than the current size, add the elements of the original list to the start of the new list. Your function should return **true** if the resize operation was successful, and **false** otherwise (e.g., if the new size is less than zero). [CLO 3, 10 points]
- Note:** Use of index is not allowed, you can only use pointers (ZERO credit for index based solution).
2. Assume you are given a linked list of sorted integers. Create a function (called **sortedInsert**, has one integer parameter) that inserts a value into the linked list at its proper position while keeping the list sorted. The list can be in ascending or descending order, and your function must work for both. A separate function can be written to determine whether the linked list is sorted ascending or descending order. Consider that the linked list class is already present. [CLO 3, 10 points]



If we want to insert 8 in to the above linked list, the correct position of 8 is between 9 and 5.

**Note:** You are not allowed to sort the list after insertion, you have to insert the value in its proper position.

3. Consider that the same operation (**sortedInsert** in question 2) needs to be performed on an array based list and the fixed size of the list is not an issue. Comment on whether a linked list or an array-based list is better in this situation. Show, using an example, which one will require the least number of operations. [CLO 2, 7 points]
4. In an array based implementation of a stack, is it a good idea to insert (push) and delete (pop) only at index 0 of the array? Provide arguments and use an example to demonstrate (diagrammatically) whether it is a good or a bad idea. [CLO 2, 8 points]
5. Create a function (named **hasLoop**) that determines whether or not a circular linked list contains a loop. If the circular linked list contains a loop, the function should return true; otherwise, it should return false. Every circular linked list contains a loop, which means that the last node points to the first node (head). This function should not detect such a loop; instead, it should detect whether the final node's next pointer is pointing to the final node itself or any other node other than the head node. Following are examples of a loop in a circular linked list. [CLO 4, 10 points]



\*\*\*\*\* The End \*\*\*\*\*



NOTE: Attempt all questions.

1. Write a function named *sortedIntersect*. The function should take two arrays sorted in ascending order as parameters. It should create and return a new array representing the intersection (common elements) of the two arrays. The original arrays should not be changed. The new array should be in descending order.

**Note:** Use of index is not allowed, you can only use pointers (ZERO credit for index based solution).

For example,

A = [1 5 6 7 9]

B = [1 3 4 7 9]

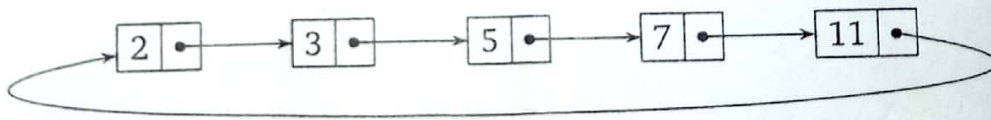
$A \cap B = [9 7 1]$

2. Write a function named *reverse* that reverses a singly linked list by rearranging all the next pointers and the head pointer. Ideally, the function should only need to make one pass of the list.

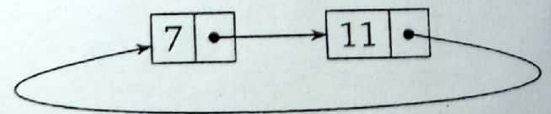
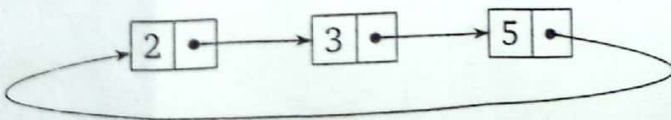
**Note:** You need to change the position of the entire node in the list, ZERO credit if only the values are swapped.

3. Given a circular linked list, write a function named *splitList* that splits the list into two circular sublists. One for the front half, and one for the back half. If the number of elements is odd, the extra element should go in the front list.

For Example, Splitting the following circular linked list:



should yield the following two circular linked lists:




5 → 9



4. A function that deletes all the nodes in a circular linked list is given below:

[5]



```
void deleteList()
{
    if(head == NULL)
        return;

    node* temp = head;
    do
    {
        temp = temp->next;
        delete temp;
    }while(temp != head);

    head = NULL;
    length = 0;
}
```

What is the problem in the code? Identify the problem, mention it and then write the correct code.

---

The End

Program: CS & SE

Semester: Spring-2023

Time Allowed: 02 hours 15 minutes

Course: Data Structure Lab

Examination: Final

Total Marks: 45 Weightage: 45

Instructor: Muhammad Usman

Date: 15<sup>th</sup> May, 2023

35 ✓

**NOTE:**

- Don't rush, carefully understand the problem and then go for solution.
- Understanding the question is the part of Exam.
- An individual may be assigned a straight-forward 0 if the submitted assessed task is copied/cheated from another individual/internet/found using internet.
- Create a folder named as Roll number followed by your, name and section (20P-8721\_Zain\_4B), copy all the scripts only (Q1.cpp, Q2.cpp, Q3.cpp), in that folder and submit the folder. Don't submit the executable files

**Question # 01**

12 ✓  
[15 Marks]

Create a function that takes an linked list of integers that represent the amount in dollars that a single stock is worth, and return the maximum profit that could have been made by buying stock on day x and selling stock on day y where  $y > x$ .

**Example 1**

Given: 44->30->24->32->35->30->40->38->15->NULL

your program should return 16 because at position 3 the stock was worth \$24 and at the position 7 the stock was then worth \$40, so if you bought the stock at 24 and sold it on 40, you would have made a profit of \$16, which is the maximum profit that could have been made with this list of stock prices. If there is no profit that could have been made with the stock prices, then your program should return -1

**Example 2**

Given: 10->9->8->2->NULL

return -1

**Example 3**

Given: 3->5->6->8->10->15->16->25->11->NULL

return 22

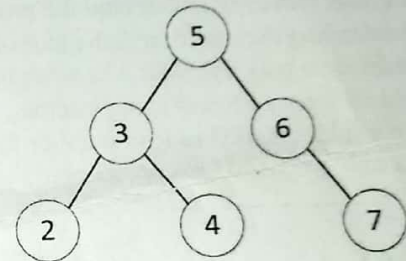


12

### Examples:

Input: K=24  
Output: true  
Explanation: The pair (6,4) product equal to 24

Input: K=2  
Output: false  
Explanation: There are no two nodes in the BST that product up to 2

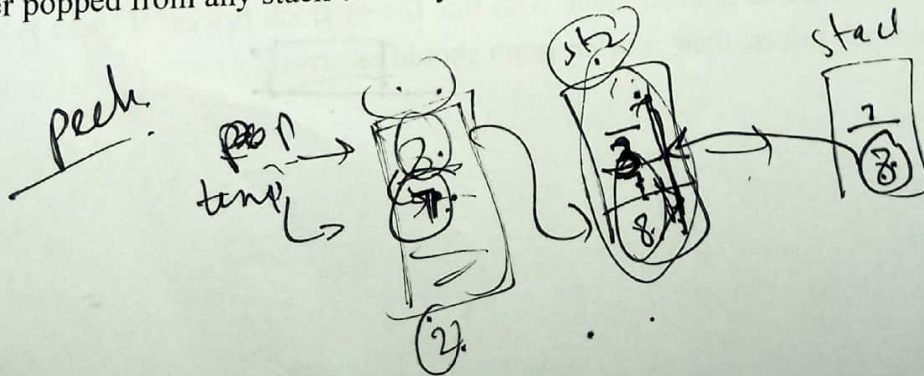


Input: K= -4  
Output: false

Input: K=15  
Output: true

12:

Write a program to construct 3 stack objects i-e (stack1, stack2, stack3). Push any five random numbers into stack 1. Now write a function to sort the stack 1 in ascending order such that the smallest number is on the top and largest is at the bottom of the stack1. Your function can only utilize stack2 and stack3 to sort stack1. Any number popped from any stack can only be stored in another stack. You cannot use any other variables.



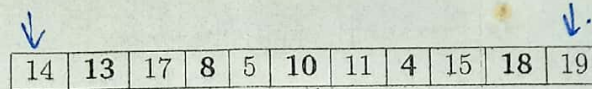


Student Name: \_\_\_\_\_  
Program: BS (CS/SE)  
Semester: Spring 2023  
Time Allowed : 1 hour  
Course: CS2001 - Data Structures

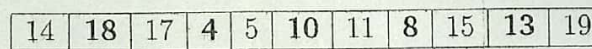
Roll No: \_\_\_\_\_  
Examination: Sessional-II  
Marks: 40, Weightage: 15  
Date: 07-04-2023  
Instructor: Waqas Ali

**NOTE:** Attempt all questions.

1. Write a function **reverseEven** that takes a queue of integers as a parameter, and the function modifies that queue, reversing the order of the integers at even positions (not indexes) in the queue while leaving the integers at odd positions in place. For example given this queue (rear to front): [CLO 1, 10 points]

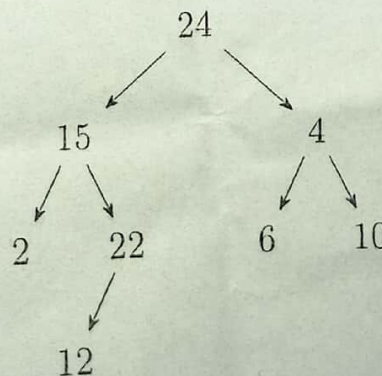


calling the function would change it to:



**Note:** You are not allowed to create a new queue to store the result (the items should be reordered in the same queue) and are only allowed to perform the enqueue and dequeue operations on the queue.

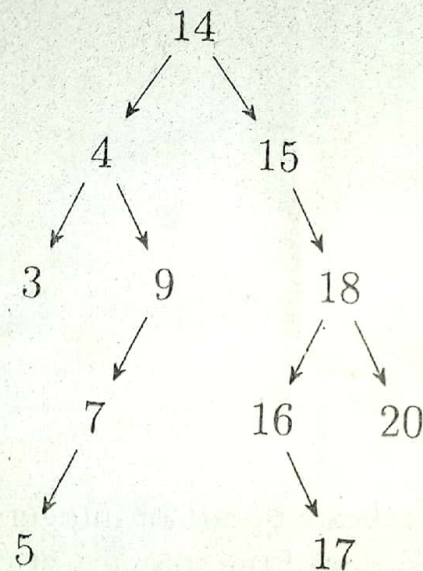
2. Write a recursive function that returns true if every node in a binary tree except the leaf nodes has only one child. It should return false otherwise. **ZERO** credit for non-recursive solution. [CLO 3, 10 points]
3. Write a function that takes an existing binary tree as a parameter. This function should create a new binary search tree from the existing binary tree. The nodes from the binary tree should be traversed and inserted into the binary search tree in an inorder fashion. (You have to write the code for insertion yourself). For example, if following binary tree is passed to the function, the nodes should be inserted into the binary search tree in the following order (inorder): 2, 15, 12, 22, 24, 6, 4, 10. [CLO 3, 10 points]





4. Consider the following binary search tree. Calculate the balance of each node in the tree, show full calculation for each node. Also state whether this tree is an AVL tree or not, if it is not an AVL tree then also mention which nodes are not balanced. [CLO 4, 10 points]

Note: You do not have write code for this question.



\*\*\*\*\* The End \*\*\*\*\*



Student Name: \_\_\_\_\_  
Program: BS (CS/SE)  
Semester: Spring 2023  
Time Allowed : 3 hour  
Course: CS2001 - Data Structures

Roll No: \_\_\_\_\_  
Examination: Final  
Marks: 90, Weightage: 50  
Date: 22-05-2023  
Instructor: Waqas Ali

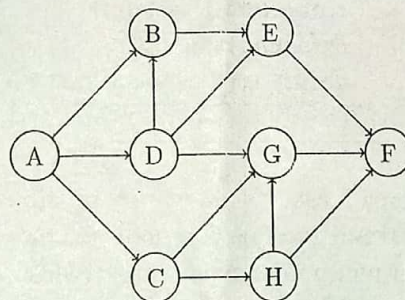
**NOTE:** Attempt all questions.

- ✓ 1. Consider that an array list of integers. The elements in the list are in no particular order. Write a function that rearranges the elements in the list such that all the even numbers appear before the odd numbers. (CLO 1) [10]
- Note:** Use of index is not allowed, you can use pointers to access the elements of the array. You should not create another array, your implementation should only modify the original list to achieve the required result.

- ✓ 2. Encode the following phrase using Huffman Codes and show each step of the process: (CLO 1) [8]

**The best of both worlds**

- ✓ 3. Consider the following graph: (CLO 2) [4 + 4 + 3 + 4 = 15]



- a. Provide adjacency matrix representation of the graph.  
b. Provide adjacency list representation of the graph.  
c. State which of the above representation is better and why.  
d. Mention how the in-degree and out-degree can be calculated for each representation.
- ✓ 4. Consider two sorted linked lists named: **list1** and **list2**, which contain integer values and are in ascending order. Write a function to merge the two lists into a single sorted linked list, without creating any new nodes. The resulting merged list should have the same nodes as the original lists but rearranged in sorted order. The function should return the merged list. The prototype of the function is given below for reference: (CLO 3) [10]

**List mergeLinkedLists(List list1, List list2);**

**Note:** You are not allowed to sort the list after merging them, you have to insert each node in its correct position in the resultant list at each step which will result in a sorted list.



5. Write a function that takes an arithmetic expression containing the three types of brackets (i.e., parenthesis "()", square brackets "[]" and curly brackets "{}") as a parameter and return if the brackets in the expression are balanced or not. The brackets will be balanced if for each type of opening bracket there is a closing bracket and vice versa. (CLO 3) [10]

For example, If the function should return **true** for the expression:  $8 \times \{5 - 2 \times [4 - (-3 - 1)]\}$  but **false** for the expression:  $5 + 4) \times [2 - (3 + 1)]$  because the closing parenthesis after 4 has no opening parenthesis.

6. Write a recursive function that counts and returns the number of nodes in a binary search tree that are not balanced (i.e., balance other than -1, 0, or 1). (CLO 3) [7]

**Note:** The function should not be dependent on variables that are external to the function e.g., global variables. Write the entire code yourself, do not assume that some of the functions are already present. ZERO credit for non-recursive solution.

7. Consider the following words and its meanings (Dictionary of words). Do the following:  $[6 + 7 = 13]$  (CLO 3)

- Build a Binary Search Tree of words and its meanings i.e., you have to write a function for insertion in a binary tree.
- Write a function to search for a word, if the word is present in the tree it should return the corresponding meaning. Otherwise, it should return an empty string.

Words	Meanings
consider	deem to be
accord	concurrence of opinion
evident	clearly revealed to the mind
utter	without qualification
obtain	come into possession of
scarce	deficient in quantity
apparent	clearly revealed to the mind

8. If we want a hash table that stores a set of strings, one possible hash function is the string's length i.e.,  $h(x) = x.length()$ . Is this a good hash function? Provide your arguments. A correct answer without a valid argument will not be considered. (CLO 4) [7]

9. Consider an array based implementation of heap. Write a function named **updateValue** that takes two parameters, one is the position (index) of a node in the heap and the other is the value to be placed at that position. Once the value of a node is updated in a heap, it might result in violation of the heap order (min or maxheap), the heap must be re-organized. Write code in the same function to re-organize the values to make it a heap. Write the entire code yourself, do not assume that some of the functions are already present. (CLO 4) [10]

\*\*\*\*\* The End \*\*\*\*\*