**BIT208 Assignment 1**

**\_\_\_\_\_**

Release Date: 28/09/2021

Due Date: 19/10/2021

Value: 20% of total assessment

Marks: Marked out of 100

Assessment Mode: Individual Assessment

**\_\_\_\_\_**

#### Expected Learning Outcomes Assessed

This assignment assesses that you are able to:

CLO1: Use standard and custom data structures to solve computing problems

CLO2: Apply key algorithms for sorting, searching and graph traversal

CLO3: Evaluate program performance in terms of space and time complexity

**Question 1 [Singly Linked List, 40 marks]**

i)In java, the largest integer (long 64 bits) that can be stored is 263 – 1 (9223372036854775807). You are required to write a program that will add and print the sum of two integers with arbitrary number of digits. The program should allow the user to enter two positive integers (as string) and will then add the two integers, stored and print the result. You will have to develop the **singly linked list** as described below,and use it to store each number (*N1* and *N2*) and perform the addition of digits one at a time. The operation of adding the numbers is like that used with pencil and paper, adding the individual digits along the numbers.

As an example output only, consider the following:

Enter first number: 73467390890216198527

Enter second number: 42965230048857126238

The sum is: 116432620939073324765

Using the fragment code from the text as a guide, complete the singly linked list class with **both head and tail**.

You are given the following **Node** class to be used in this question. Accordingly, you will define your LinkedList class.

public class Node<Type>

{

// Instance variables:

private Type element;

private Node<Type> next;

/\*\*

\* Creates a node with the given element and next node.

\*/

public Node(Type e, Node<Type> n)

{

element = e;

next = n;

}

/\*\*

\* Create a node with the given element, with next node

\* sets to null

\*/

public Node(Type e)

{

this(e, null);

}

/\*\*

\* Creates a node with null references to its element

\* and next node.

\*/

public Node()

{

this(null, null);

}

// Accessor methods:

public TypegetElement()

{

return element;

}

public Node<Type > getNext()

{

return next;

}

// Modifier methods:

public void setElement(TypenewElem)

{

element = newElem;

}

public void setNext(Node<Type> newNext)

{

next = newNext;

}

}

ii. Implement a method that will concatenate two singly linked lists N1and N2, into a

single list *C*that contains all the nodes of *N1*followed by all the nodes of *N2*.

iii. Discuss about the complexity of operations - addition in (i) and concatenation in (ii) in terms of Big-Oh (O) notation.

[ **NOTE** : Create driver program to test the implementation and attach the output]

**Question 2 [ Recursion30 marks]**

Consider the power function *power(m, n)* where *m* is a double and *n* is a integer, *n*>= 0. The function should return *mn*.

1. Implement this function using a for loop.
2. Implement this function using recursion (looping):  
   
3. Implement this function using an alternative recursion algorithm:  
   
4. Compare the actual running times for the three implementations and fill in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **M** | **n** | **Implementation (a)** | **Implementation (b)** | **Implementation (c)** |
| 3 | 5 |  |  |  |
| 3 | 50 |  |  |  |
| 3 | 500 |  |  |  |

1. Perform a **recursive trace** of the two recursive functions in implementation (b) and (c) for *power*(3, 5)
2. Based on your results in (d) and (e), discuss reasons for using recursion and which recursion algorithm is better, and why.

**Question 3 [Use of stacks, and queues30 marks]**

A *palindrome* is a sequence of characters that reads the same backward and forward. The word noon, and the number 123454321 are palindromes.

Write Java program to test lines of input text to see whether they are palindromes. In this program we are going to ignore spaces and punctuation, and lowercase and uppercase letter are considered equivalent. Therefore, the following two input lines are palindromes:

Madam, I’m Adam

Able was I ere I saw Elba

Here is the output when this program is run with a few lines of text:

>Noon

is a palindrome

>Good

is not a palindrome

>able was I ere I saw elba

is a palindrome

>Able was I ere I saw Elba

is a palindrome

>123454321

is a palindrome

>12345

is not a palindrome

Using a combination of **stack** and **queue y**ou are to implement the program to check for palindrome:

**IMPORTANT NOTE:**

Students are **NOT** allowed to use the Java predefined collection classes such as Stack, ArrayList, etc to solve the programming problems in this assignment. The students are required to develop the data structures class based on what they have learned in class.

**If the students are found to use predefined Java collections classes to solve the problem, they will not be awarded any marks at all.**

**Documentation**

* You should include comments in your code stating what each method does and explaining any complex sections of code.
* You should also include your name and student ID as comments within the code.
* You should use meaningful variable names so that your code is to some extent self-documenting.

**What to Submit**

You should submit the following:

* A cover-sheet stating your student number. NOTE: Do not put your name on the cover-sheet.
* The source files, with extension .java, (Project folder) and the word document.

[ **Submit on LMS**]

 **Assignment Cover Sheet**

Assignment No.: \_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Student Information (For group assignment, please state names of all members)** | | **Grade/Marks** |
| **Name** | **ID** |  |
|  | B1901898 |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Module/Subject Information** | | **Office Acknowledgement** |
| **Module/Subject Code** | BIT208 |  |
| **Module/Subject Name** | Data Structures and Algorithms |  |
| **Lecturer/Tutor/Facilitator** |  |  |
| **Due Date** | 19 October 2021 |  |
| **Assignment Title/Topic** | Assignment 1 |  |
| **Intake (where applicable)** |  |  |
| **Word Count** | N/A | **Date/Time** |

**Declaration**

* I/We have read and understood the Programme Handbook that explains on **plagiarism**, and I/we testify that, unless otherwise acknowledged, the work submitted herein is entirely my/our own.
* I/We declare that no part of this assignment has been written for me/us by any other person(s) except where such collaboration has been authorized by the lecturer concerned.
* I/We authorize the University to test any work submitted by me/us, using text comparison software, for instances of plagiarism. I/We understand this will involve the University or its contractors copying my/our work and storing it on a database to be used in future to test work submitted by others.

Note: 1) The attachment of this statement on any electronically submitted assignments will be deemed to have the same authority as a signed statement.

2) The Group Leader signs the declaration on behalf of all members.

|  |  |
| --- | --- |
| Signature: | Date: |
| E-mail: |  |

*ACA-F-020(010611:01)*

Page **1** of **2**

|  |
| --- |
| **Feedback/Comments\*** |
| **Main Strengths** |
|  |
|  |
|  |
|  |
|  |
|  |
| **Main Weaknesses** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **Suggestions for improvement** |
|  |
|  |
|  |

|  |  |
| --- | --- |
|  | **Student acknowledge feedback/comments** |
|  |
| Grader’s signature | Student’s signature: |
| Date: | Date: |

Note:

1. A soft and hard copy of the assignment shall be submitted.
2. The signed copy of the assignment cover sheet shall be retained by the marker.
3. If the Turnitin report is required, students have to submit it with the assignment. However, departments may allow students up to **THREE** (3) working days after submission of the assignment to submit the Turnitin report. The assignment shall only be marked upon the submission of the Turnitin report.

\*Use additional sheets if required.

*ACA-F-020(010611:01)*

Page **2** of **2**

Name: StudentID:

**Marking Scheme**

|  |  |  |  |
| --- | --- | --- | --- |
| **Question** | **Criteria** | **Allocated Marks** | **Awarded Marks** |
|  |  |  |  |
| 1.a | Data structures based on Singly Linked List set up with required functionality implemented.  Driver class to Test the working.  Documentation including the output | 20 |  |
| 1.b | Concatenation algorithm correctly implemented.  Driver class to Test the working.  Documentation including the output. | 10 |  |
| 1.c | Comprehensive discussion on analysis of algorithms | 10 |  |
| 2a | Power() function using for loop | 5 |  |
| 2b | Power() function using recursion | 5 |  |
| 2c | Power() function using recursion | 5 |  |
| 2d | Analysis of time taken | 5 |  |
| 2e | Recursive Trace | 5 |  |
| 2f | Discussion | 5 |  |
| 3 | Read input data + Sanitation  Correct implementation of Stack to read/store the input string.  Correct implementation of Queue to read/store the input string.  Driver class to Test the working.  Documentation including the output | 30 |  |
|  | **TOTAL** | **100** |  |
| **Deductions** | Assignment Cover Sheet not attached/filled (5) |  |  |
|  | Late submission (5 / day) |  |  |
|  | Cover Sheet / Marking Scheme not attached (5) |  |  |
|  | **DEDUCTIONS** |  |  |
|  | **TOTAL MARKS** |  |  |