Softwarica College of IT & E-Commerce
STW210CT: Programming,
Algorithms and Data Structures

in collaboration with

Assignment Brief 2021

Module Name STW210CT: Programming, Algorithms and Data Structures	Ind/Group **Individual**	Cohort **Sept 2020**	Module Code: **STW210CT**
Coursework Title (e.g. CWK1)			Hand out date: TBD
Lecturer: Hikmat Saud			Due date: TBD
Estimated Time (hrs): Word Limit*: n/a	Coursework type: Individual / Practical		% of Module Mark 70%
Submission arrangement online via Softwarica Moodle: Upload through Assignment links File types and method of recording: URLs (source code repositories) Mark and Feedback date: Within 3 weeks of assignment submission Mark and Feedback method: Rubric marks and comments via Softwarica LMS			

Module Learning Outcomes Assessed:

1. Write software to solve a range of problems.
2. Implement and use simple searching and sorting algorithms.
3. Use libraries to extend the functionality of the base language.
4. Use basic design and testing strategies

Notes:

1. You are expected to use the CUHarvard referencing format. For support and advice on how this students can contact Centre for Academic Writing (CAW).
2. Please notify your registry course support team and module leader for disability support.
3. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computer. Students should therefore regularly back-up any work and are advised to save it on the University system.
4. If there are technical or performance issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours or the next working day if the deadline falls on a Friday or over the weekend period. This will be communicated via email and as a Softwarica Moodle announcement.

Week 1

a) provided set of binary numbers in a linked list, find all unique permutations and return each integer value of generated permutations as linked list. [3 Marks]

Input: 1 $\longrightarrow$ 0 $\longrightarrow$ 1

Permutation:

[1, 0, 1] $\longrightarrow$ 5

[1, 1, 0] $\longrightarrow$ 6

[0, 1, 1] $\longrightarrow$ 3

Output: 5 $\longrightarrow$ 6 $\longrightarrow$ 3

b) Given array of contiguous prime number, return array with missing element in between. [3 Marks]

Input [2, 3, 5, 11, 17, 23]

Output: [7, 13, 19]

Week2

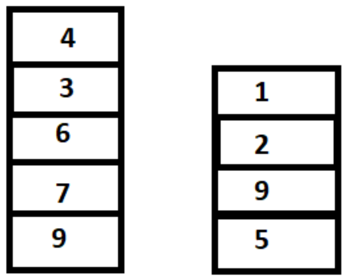
a) Using stack check for balanced parenthesis within a string. [2 Marks]

Input "{()}{}"

Output true

b) Given two stack a and b, if allowed to pop from each stack, all popped values are added together, return maximum pop one can make without sum exceeding some given integer k. [3 Marks]

Given k=11



Output: 4

Week3

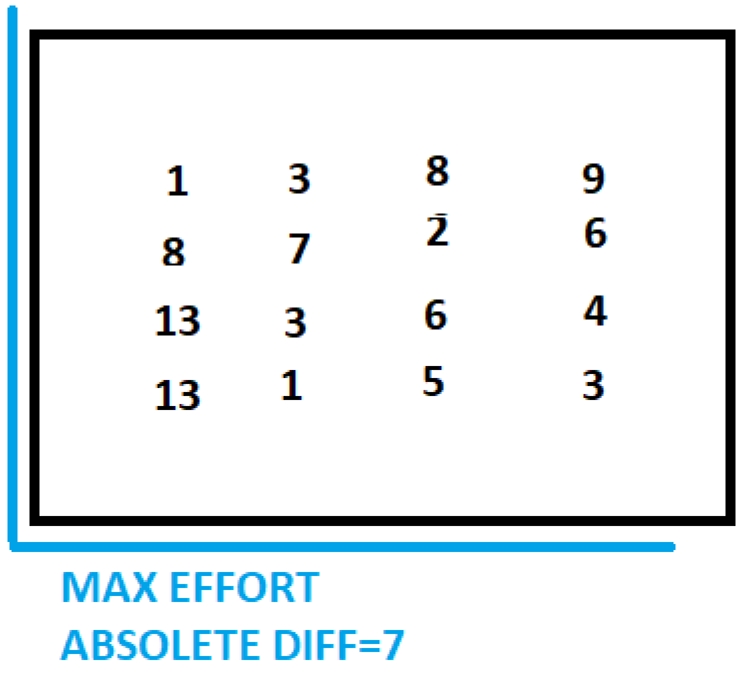
Convert Infix expression to Reverse Polish Notation (RPN) and Evaluate Reverse polish notation using stack. [5 Marks]

Example
Infix: 4(5+6) (input)
RPN conversion: 456+* (output)
Evaluated value: 44 (output)

Week4

You have provided height of wall represented in 2D array where Each cell represent height of wall example a[i][j]=2,.suppose you are at position a[0][0], return maximum effort and minimum effort as array in sorter order required to travel to bottom right Cell of the given 2D array.

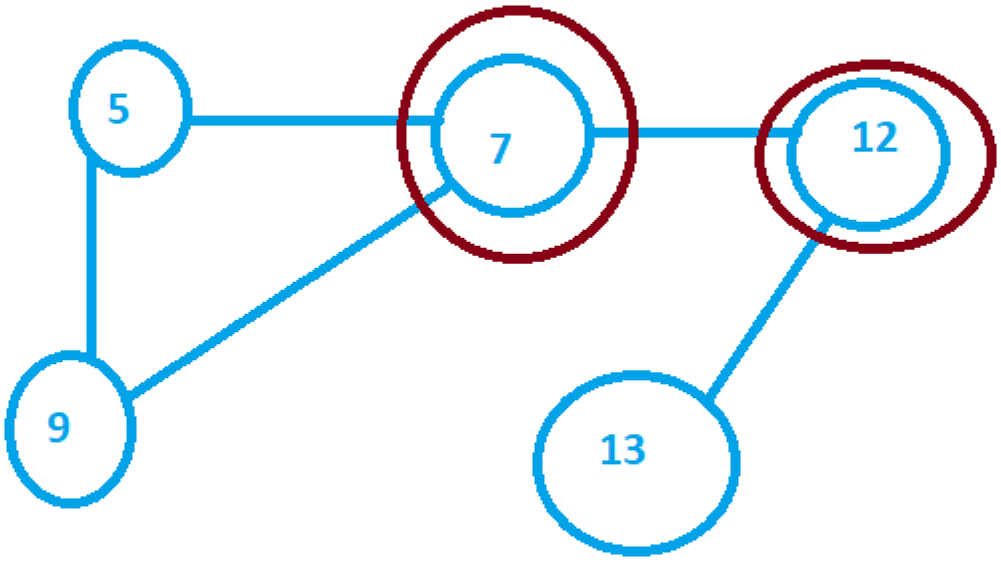
An effort is maximum or minimum absolute difference in height of two consecutive wall. NOTE: A person is allowed to go through rows and column. [5 Marks]



INPUT = {{1,3,8,9},{8,7,2,6},{13,3,6,4},{13,1,5,3}}

Week 5

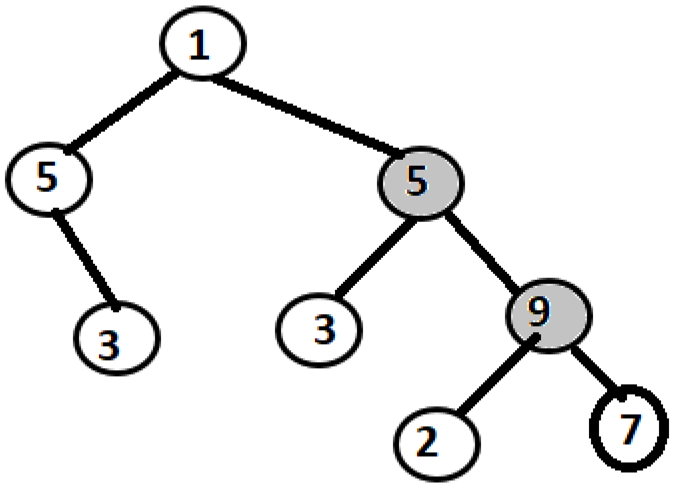
You are hired as a network engineer, given an undirected graph representing a local area network of organization, design an algorithm to find vulnerable points in a connected network. Where each vulnerable point is the vertex within a graph whose failure will split the network and fails the whole communication system. [7 Marks]



Vulnerable points: 7, 12

Week6

Provided a Binary Tree and an array of nodes from tree. Return true if each node from array has connected corresponding balanced leaf node, else return false. [5 Marks]



Input: head=[5,9] root=[1,5,5,null,3,3,9,2 ,7]

Output: True

Week7

a) Write a java program to solve convex hull problem. [4 Marks]

b) Design a circular queue with implementation of following operations

front(), Rear(), enQueue(int val), deQueue(int val), IsEmpty(), isFUll(). [3 Marks]

Week 8 to 11

Suppose you are assigned with project to develop an application to store information about network architecture of your organization. This application will helps to find information about connected network devices. This application also allow users to find optimal route between provided source and destination device, so that network engineer can define optimal path for data transfer. Developed system should meet following requirements.

- System should allow Registration and login [12 marks]
- Add/update/delete Network devices [12 marks]
- Display network architecture [12 marks]
- Recommend optimal path between provided source and destination device [12 marks]
- View details of each device including ports, product name, product type etc. [12 marks]

Note: For optimal path, cost of each hop is 1, use graph data structure and files to store information.

Total [60 marks]

Marking Notes

1. All submitted coursework will be assessed via VIVA conducted at the end of this semester.
2. Each VIVA will last 20 minutes.
3. You will submit on the deadline a document (PDF or Word) on Moodle containing all the coursework tasks solved and including a link to your GitHub Classroom repository shared via Softwarica LMS.
4. During the VIVA you will be assessed with few relevant random questions.
5. If you submit only some of the task, your mark will be proportional to that.
6. The marking criteria valid for week 8-11 is presented below.

Criteria	0	2	4	6	8	10
Feature complete	Not submitted	Only few features implemented and are not executing	Many of the features are implemented but are not executing correctly	Many of the features are implemented and are executing correctly	Most of the features are implemented and are executing correctly	All features implemented and are executing correctly
Code aesthetic	Not submitted	Assignment submitted but not commented and formatted. variable's/classes/ function are defined but meaningless	Lack of comments, formatted in Source code. Only few classes and functions are define but hard to read	Lack of comments, formatted in Source code, but meaningful variable/class/ function names are used few function are define.	Lack of comments, formatted in Source code, but meaningful variable/class/ function names are used. Code is easy to read	Source code is well commented, properly formatted, meaningful variable/function/class names are used. Code is easy read and understand, having many pure functions.
GUI	Not submitted	Hard to use. Only some component are used and unmanaged	Few frame are difficult to use. UI component are used but unmanaged.	Some frame are difficult to use. UI component are used but unmanaged.	Easy to use, Proper use of various UI components. User Interaction is low	Easy to use, Proper use of various UI components, Clean and interactive UI
I/P Validation	Not submitted	Only few input fields are validated. Error message are not shown	Only few inputs field are validated. Error messages are shown in code format	Most input fields are properly validated. Error messages are shown in code format	Most input fields are properly validated. Error messages are properly shown in natural language	All input fields are properly validated. Error messages are properly shown in natural language.
Unit Testing	Not submitted	Only few features are tested without using framework and many of them are fail	Many of the modules are tested and many of them are fail	Many of the modules are tested using suitable unit testing framework.	Most of the modules are tested using suitable unit testing framework. Should have partial coverage.	All modules are unit tested using suitable unit testing framework. Should have full testing coverage.

Viva	Not present (Assignment submitted but absent in viva)	Could not explain reasoning behind the code. But answered only one viva question	Could explain basic terms but not about algorithm. But answered only two viva question	Could explain reasoning behind the code, including use of loops, conditions, algorithms. answered only three viva question	Could explain reasoning behind the code, including use of loops, conditions, algorithms. answered only four viva question	Could explain reasoning behind the code, including use of loops, conditions, algorithms. Answered all five question