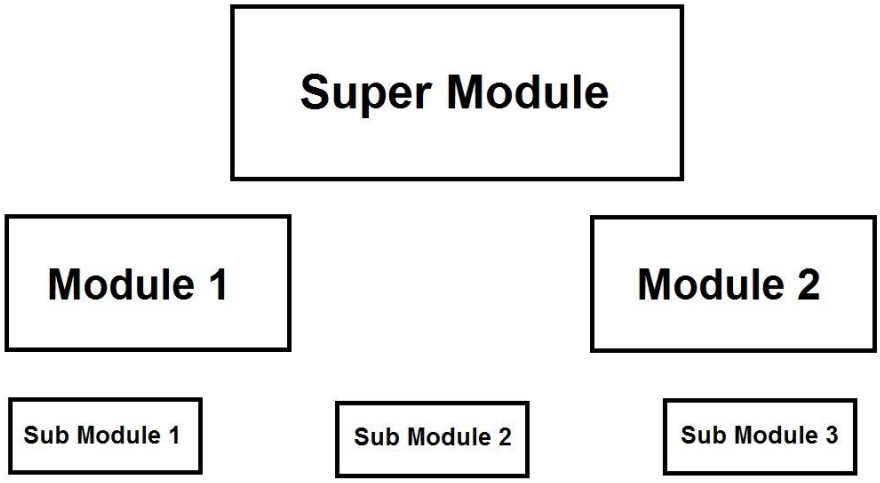
National University of Computer & Emerging Sciences, Karachi
Fall-2018 CS-Department
Quiz 0

Course Code: EL-213	**Course Name:** Computer Organization & Assembly Language	
Student IDs		
Instructor: Syed Zain Ul Hassan	**Time Allowed:** 45 min	**Marks:** None

Challenge# 1:

Consider an imaginary architecture as shown in the diagram:



Read these conditions carefully:

i. The simplified architecture consists of 1 super module, 2 modules and 3 sub-modules.
ii. Each of the super module, modules & sub-modules can only be in one of the two possible states: ***Active(represented by 1)*** or ***Inactive (represented by 0).***
iii. If super module becomes **Active** & any of the two modules are also **Active**; then sub-module 2 becomes **Active** (but sub-module 1 & 3 remain **Inactive**).
iv. If super module becomes **Active** but none of the modules are **Active** then all the sub-modules become **Active**.

v. If super module is **Inactive**, then all the modules & sub-modules instantly become **Active**.

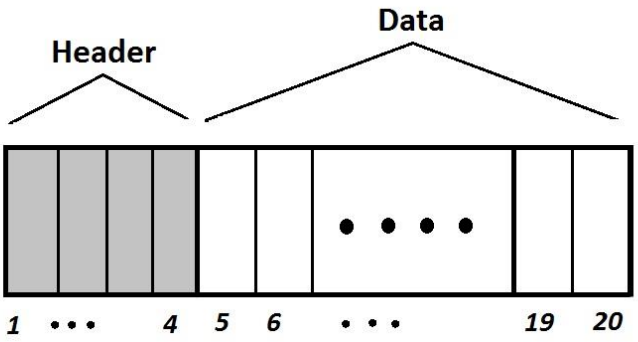
Implement this architecture using any of the techniques you have learned thus far in Assembly Language. To test your program, take input state of the Super module & the two modules from the user and based on that input, display the values of all sub-modules.

Challenge# 2

i) Write a program to implement Binary Search Tree (you can choose to make the tree for any given depth *d*).

Challenge# 3:

Consider the following array of size 20 divided into two ***logical*** sections:



Read these conditions very carefully:

i. The first 4 elements make up header section, while rest of the array contains data.

ii. Each index in the data section can store a value in the range 3 to 233.

iii. In the data section, if a value outside the specified range is stored at an index then it is regarded as a 'corrupted index'.

iv. In the header section:

a) The 1st element must be the count of all corrupted values in the data section.

b) The 2nd element must be the count of all legal (non-corrupt) values in the data section.

c) The 3rd elements must be the equal to the index where largest value is stored in data section.

d) The 4th element must be equal to index where smallest value of data section is stored.

Implement this array and test your program, take input values from the user to fill up the memory unit and compute (and also display) contents of the header section based on that input data.

~ May the odds be with you