

## Lab#10:

### Array of pointers, multiple indirection, functions with pointers

---

Total marks: 20 (10 marks each)

**Submission deadline:** 10<sup>th</sup> May, 2020, Sunday till 11:59pm

**Submission guidelines:**

1. Put all your .cpp and relevant files in a folder. Zip that folder and name it with your roll number.
  2. Email the zipped folder at [natalia@pucit.edu.pk](mailto:natalia@pucit.edu.pk). You must put the **subject** of your email as:  
**"Lab#10"**
  3. Students who are facing issues with their laptop or compiler can write the code on paper and send the screenshots, following the above guideline. Since your issue will be genuine therefore relaxation will be given accordingly. Don't worry!
- 

### Task#1:

---

You need to design two network topologies- ring and star. Let us explore what these topologies are.

#### Network topology

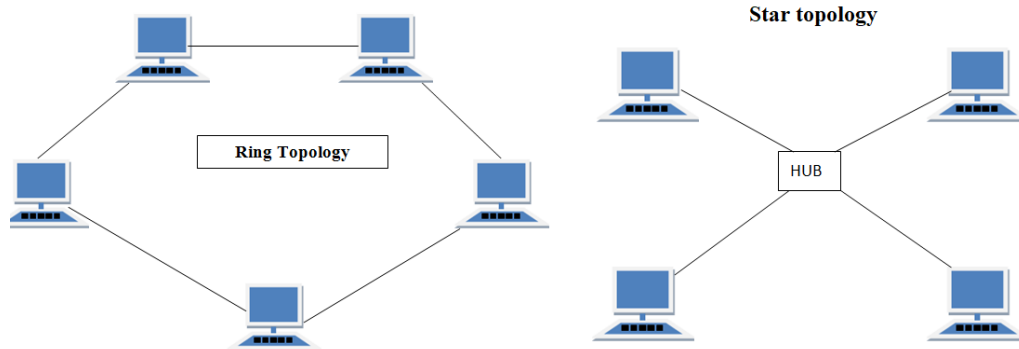
Network Topology is the schematic description of a network arrangement, connecting various nodes(sender and receiver) through lines of connection.

#### RING Topology

It is called ring topology because it forms a ring as each computer is connected to another computer, with the last one connected to the first. Exactly two neighbours for each device.

#### STAR Topology

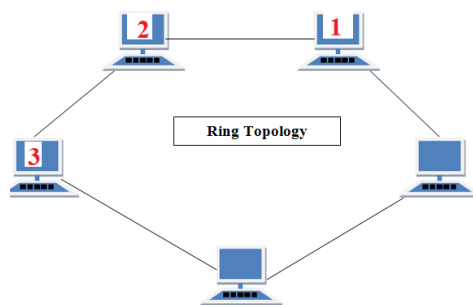
In this type of topology all the computers are connected to a single **hub** through a cable. This hub is the central node and all others nodes are connected to the central node.



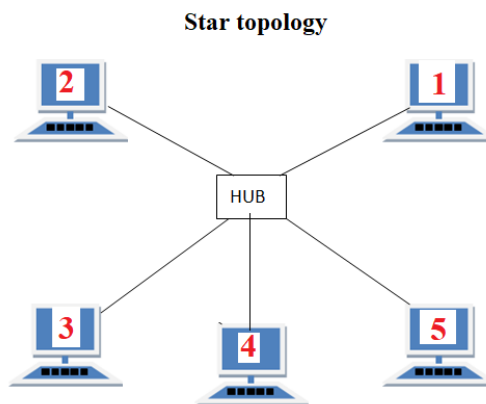
<https://techdifferences.com/wp-content/uploads/2018/03/star-vs-ring-topology.jpg>

In this lab task, **you need to design these two topologies**. Suppose you are given 5 quad core systems and 4 octa core systems. These systems are identified by the unique addresses allotted to them. Thus, at the start you need to store these 10 addresses somewhere in an array (of size 10 obviously; use static array). Then, take the systems present at even indexes (0,2,4...) of an array and design **ring topology** using them. Take the systems at odd indexes (1,3,5..) and design a **star topology** using them. In each design, you can connect any two systems by creating a *pointing* relationship between them. You can point any of the two systems to the other one. See example below. You can connect the systems in any of the two ways:

1. Point 1 to 2, 2 to 3, 3 to 4, 4 to 5, 5 to 1
2. Point 1 to 5, 5 to 4, 4 to 3, 3 to 2, 2 to 1



In star topology, you also need to simulate (design) hub. Hub is basically acting as a switch, to which you can connect various systems. Consider the example below:



In this scenario, all the five systems are connected to one hub. You need to use the knowledge that you got from pointers to simulate this scenario. Think and decide how you can connect hub with the other devices/systems.

## Guidelines

- After designing the topologies, you need to print the following things on the console:
  1. Address and contents of all the systems (for both the topologies)
  2. Address and contents and hub.

- Use as many *array of pointers, array of pointers to pointers, or pointers* as you want, accordingly to the needs of your logic.
- Make sure to access quadcore and octa core systems using a loop and meaningful expression in an if statement. Similarly do the same when you access systems while designing topology.

## Task#1:

---

Declare an array of 100 integers and initialize the contents with 0. Then, implement the following functions:

1. void setValues(int \*)
  - This function takes a pointer to an integer array as an argument. It
    - places value **1** at indexes that are multiple of 5 (i.e. 5,10,15,...)
    - places value **2** at indexes that are multiple of 2 (i.e. 7,14,28,...)
2. void AddtoMid(int \*, int \*)
  - This function takes two pointers as an argument. Pointer 1 is set to point at array's 0<sup>th</sup> index. Pointer 2 points at the last cell i.e. 99<sup>th</sup> index. This setting should automatically be done when you make function call. Your task is to declare another pointer **Pointer 3** and make it point to the mid of an array i.e. cell with index 49. Use meaningful expression to reach to the mid point of an array. Then, using pointer 3's notation, store the following at the mid of an array:  
 Sum of the contents of integer locations pointed by pointers **Pointer 1** and **Pointer 2** (use pointer 1 and 2 notation respectively to access the locations). That is you need to add up things: `array[49]= array[0]+ array[99]`. You are not allowed to use array notation! Use pointer notation to access the specific cell of an array directly.

## Guidelines

**Main function should only declare an array and call these two functions. Before and after calling these functions, contents of arrays should be printed to the console (in main function obviously).**

**Your functions should not have any print statement in it. All the printing logic should be there in main() only!**