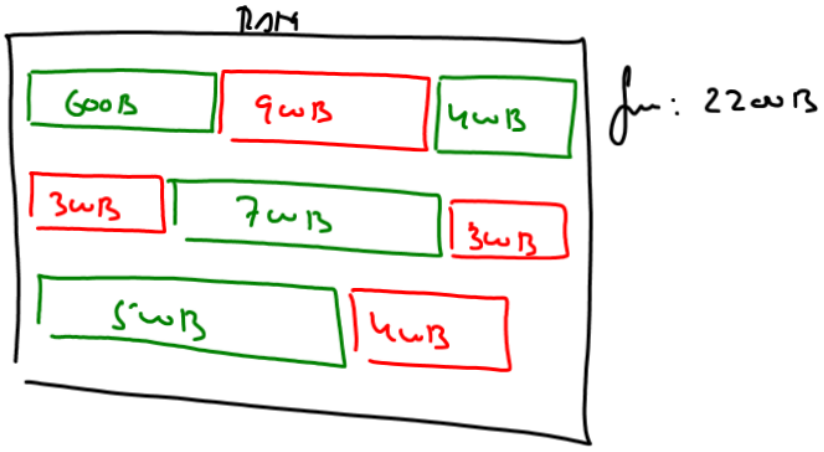
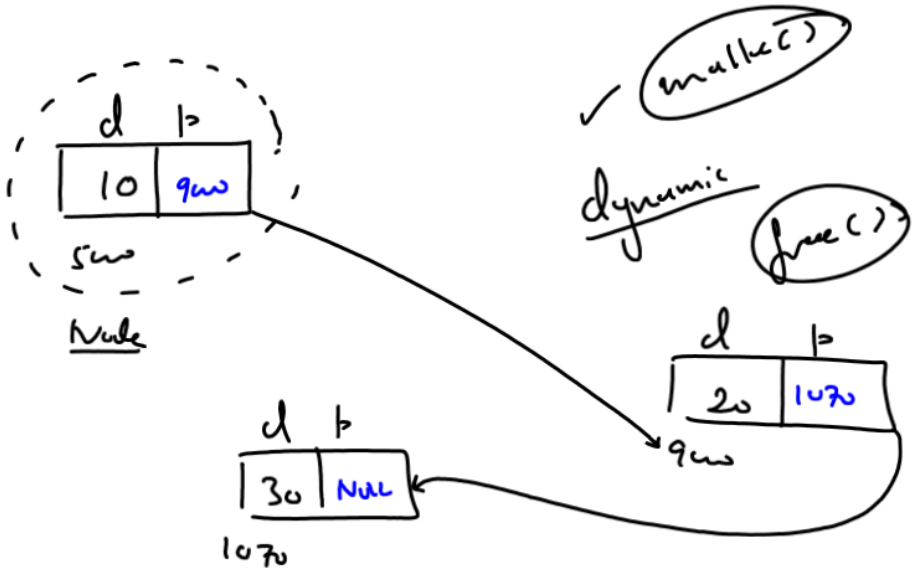
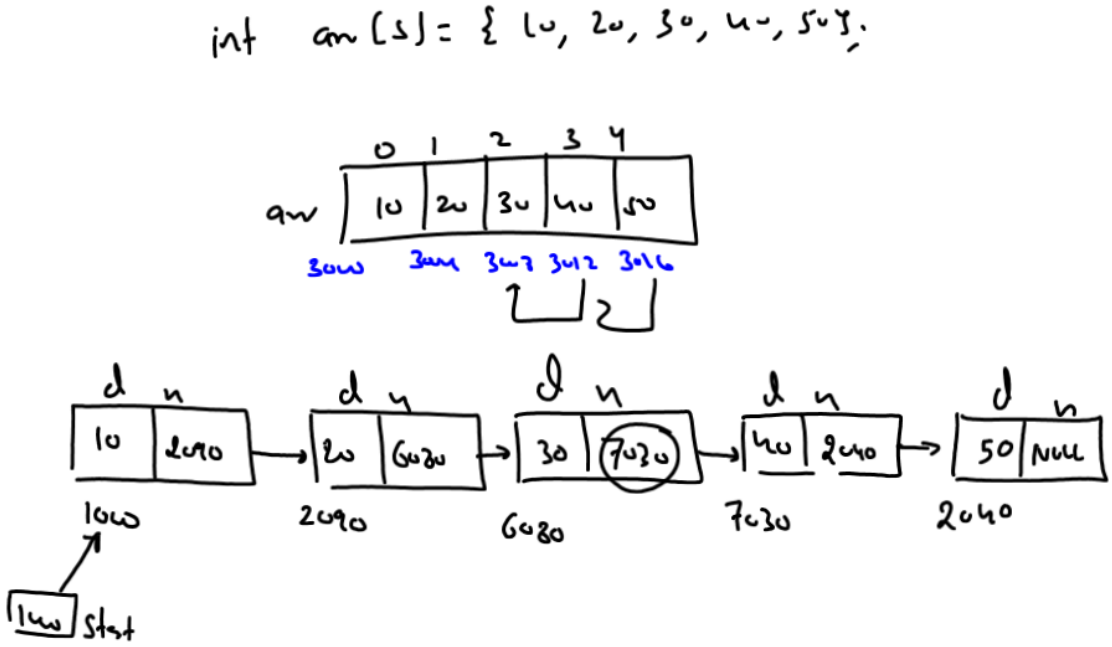
LINKED LIST

int arr[200];





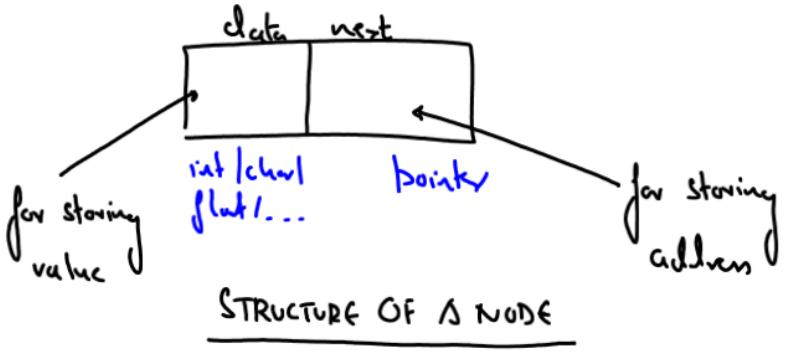


What is a Linked List?
==========

1. A linked list is a linear Data Structure which is used to remove the drawbacks associated with an Array.

2. A linked list is a collection of **NODES** and every node itself contains two members:

   a) Data : For holding value
   b) Next : For holding address of the next node

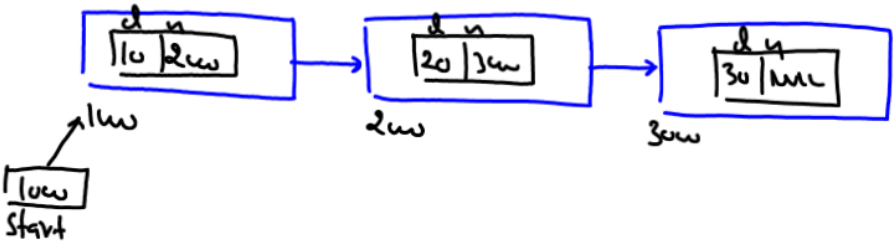


3. Every node in a linked list is created at runtime so we say that linked list is a **dynamic data structure**

4. These nodes are created in **heap area** of RAM at random locations

5. Whenever we create a linked list we actually do 3 things:

a) We allocate space for a node dynamically using **malloc()** function
b) We set the **data** part of the new node to the desired value and the **next** part to **NULL**
c) We **connect** the newly created node with the previous node by storing its address in the **next** part of previous node.



Advantages of Linked List
==========
1. Allows us to use memory even if no **continuous** space is free.

2. Since nodes are created dynamically so we can add or remove nodes from the list as per our requirement.

3. This means that linked list gives a better utilization of RAM as compared to an Array.

4. It is very easy to add or remove nodes in b/w the list since only two operations are needed irrespective of the location of the node in the list.

5. So, linked list is a very good choice when we have more insert and delete operations and less amount of searching or retreival operation.

Drawbacks of Linked List
==========
1. Per node linked list consume 'x' bytes extra where 'x' is size of pointer on the given compiler.

2. Linked list takes more time to access a particular node as compare to an array because to access nth node in the linked list we have to traverse all the previous n-1 nodes.

3. It is a bit difficult to implement a linked list as compare to an array because pointers are used heavily to implement a linked list.

4. A broken link (a pointer with wrong address) makes the further nodes of the list inaccessible.

Benefits of an Array
==========
1. Per element and array consumes only those many number of bytes which are required as per the data type and size. For example considering that an integer is of 4 bytes, an array of 10 integers will require 40 bytes only. While if we create a linked list of 10 nodes then it will require 120 bytes, 4 bytes for data and 8 bytes for pointer per node. This means array consumes less memory compared to linked list.

2. Array provides random access so to access any element in the array we can directly access it using its index.

3. It is bit easy to implement an array as compared to a linked list since no pointers are involved.

Drawbacks of an array
==========
1. Arrays require continuous memory for all the elements.

2. Array is not growable by nature i.e. once we have created an array neither we have increase its size nor we can decrease its size.

3. It is very difficult to add or remove any data in b/w the array because we need to shift all the elements either towards right (in case of insertion) or towards left (in case of deletion) .

4. Thus array is good if frequent operations are searching or retreival and it is a bad choice if frequent operations are adding and removing.