Insertion at beginning

(2.) Video - 02 : Singly Linked list

* A singly linked list is one in which all the nodes are linked together in some sequential manner.
* First node is accessed by an external pointer list (head.
* Pointer points to next node
* Last node contains NULL value.

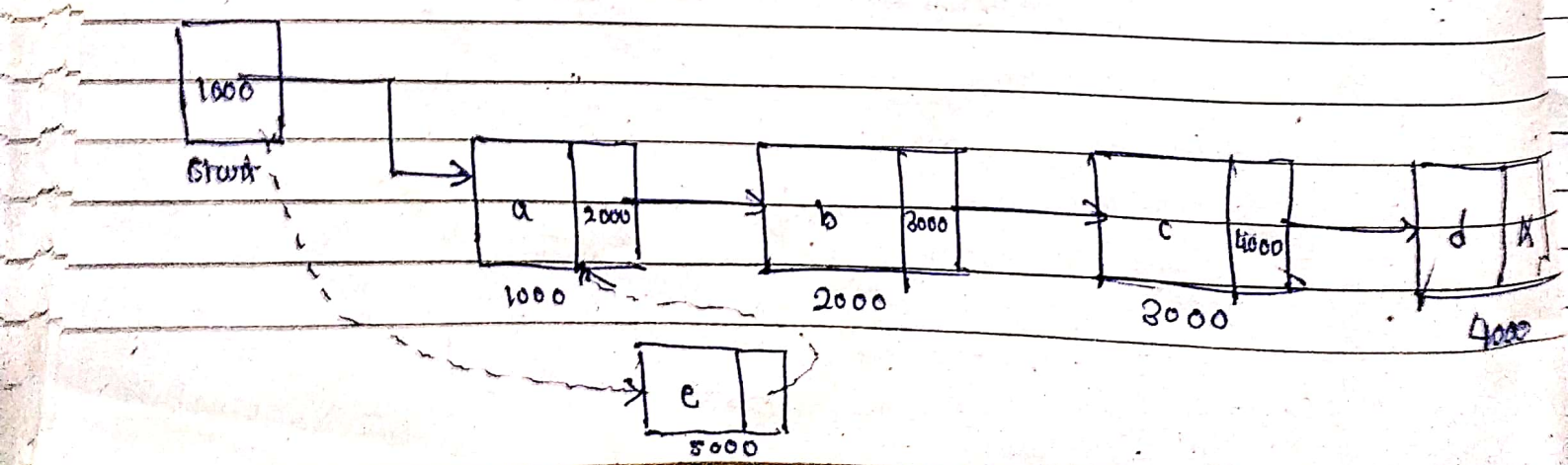
Operations in singly linked list:

→ Insertion at beginning
→ Insertion at end
→ Insertion at any position
→ Program for implementation

→ Delete at beginning
→ Deletion at end
→ Deltion at any pos
→ Program

① Insertion at begginning:

Algorithm:



Insertatfirst(start, data, link, avail, item)

start: pointer pointing to first node
data: data part of the node
link: address part of the node
avail:
item: item to be inserted

Step 1: [To check overflow]
if (avail = -1) then
write "Overflow"
return

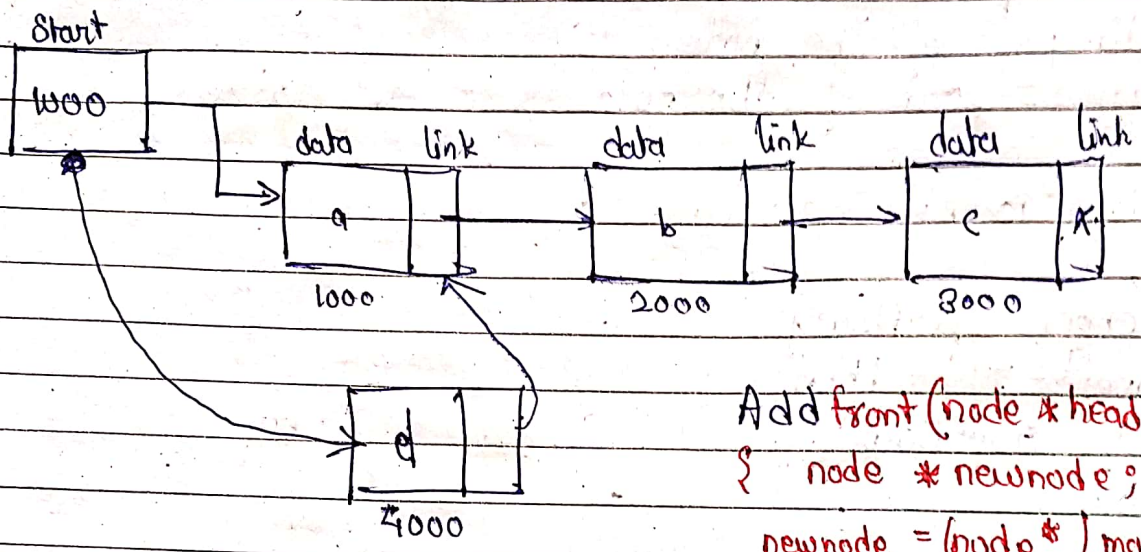
Step 2: [To break the node from avail list]
newnode = avail
avail = link[newnode]

Step 3: [To add new node into data list]
link[newnode] = start
start = newnode

Step 4: [To add item]
data[newnode] = item

Step 5: Exit

C - programming Implementation:




```
Struct node
{   char data;
    struct node *link;
};
typedef structnode node;

node* createnode (char info)
    {   node *p;
        p= (node*) malloc (size of (node));

        p-> data = 'a'
        p -> link = NULL;
        return p;
    }
```

```
Addfront (node *head, char info)
{   node *newnode;
newnode = (node*) malloc (sizeof(node));
newnode -> data = info;
newnode -> link = head;
 head = newnode;
return head;
}
```