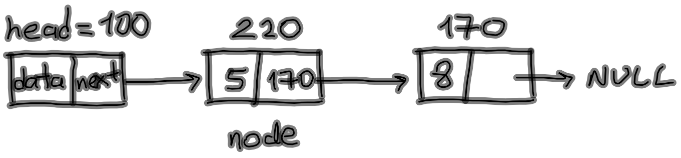
Linked Lists(Linear)

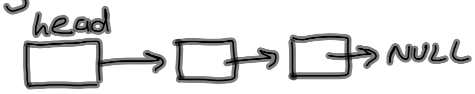


logical view

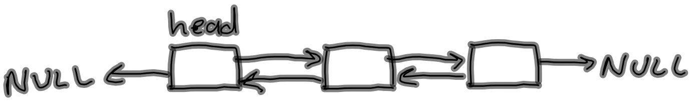
```
struct node{
        int data;
        struct node *next;
};
```

Linked List Types

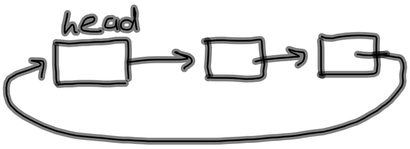
1. Single linked lists(next)



2. Double linked lists (next+prev)



3. Circular Linked Lists (next)

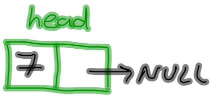


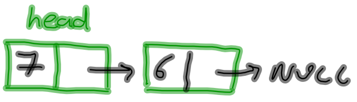
Linear Linked List Operations

1. Creating a list
2. Inserting elements into list → front → end → inside
3. Deleting an element
4. Searching
5. Counting
6. Printing
7. Sorting

Creating a List

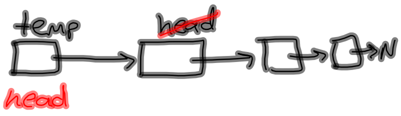
```
main(){
        struct node *head;
        head= (struct node *)malloc(sizeof(struct node));
        head->data= 7;
        head -> next= NULL;
        head -> next= .... malloc....
        head->next->data=6;
        head->next->next= NULL;
}
```

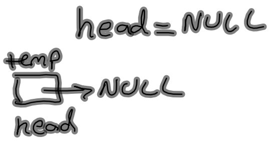




Inserting an element in front of the list

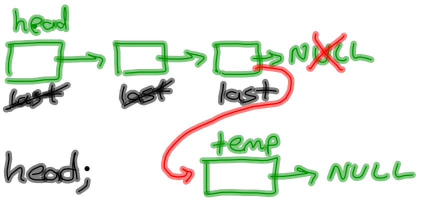
```
struct node *addfront(struct node *head,int key){
        struct node *temp= .... malloc.....
        temp->data= key;
        temp->next= head;
        head= temp;
        return head;
}
```



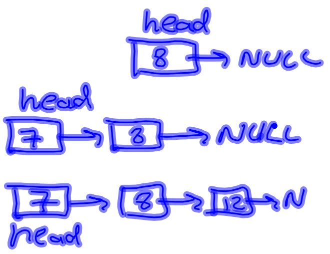


Inserting an element to the end of the list

```
struct node *addlast (struct node *head,int key){
        struct node *temp= .... malloc.....
        temp->data= key;
        temp->next= NULL;
        if (head==NULL)
                head=temp;
        else{
            struct node *last=head;
traversal { while( last->next!=NULL)
                   last= last->next;
            last->next=temp
        }
        return head;
}
```



```
main(){
     struct node *head= NULL;
     head= addfront(head,8);
     head= addfront(head,7);
     head= addlast (head,12);
}
```

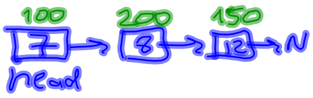


Printing the elements in a list

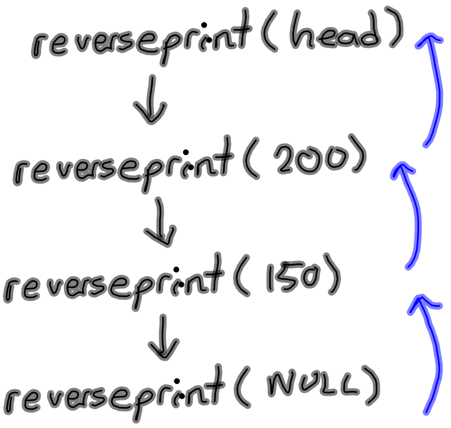
```
void print (struct node *head){
     if ( head==NULL)
          printf (" List is empty");
     else {
          while ( head != NULL){
               printf("%d\t", head->data);
               head= head->next;
          }
     }
}
```

Printing the elements in a list in reversed order

```
void reverseprint(struct node *head){
     if ( head==NULL)
          return;
     reverseprint ( head->next);
     printf("%d\t", head->data);
}
```



12 8 7



reverseprint(head)

reverseprint(200)

reverseprint(150)

reverseprint(NULL)

Counting the elements in a list

```
int count(struct node *head){
      int counter=0;
      while (head!=NULL){
            counter++;
            head=head->next;
      }
      return counter;
}
```

HW: Write the fonction above recursively.