Yashwanth Kiran.S
1BM19CS187 '3D' Batch-2

Date: 23/11/2020

LAB 5 & 6

Linked list implementation

```
// for insertion
void insert_at_beginning()
{
  struct node * ptr
  ptr -> data = new_item
  ptr -> next = head
  head = ptr
  print "node inserted at beginning"
}

insert_at_last()
{
  struct node *ptr, *temp
  ptr = (struct node*) malloc (size of (structnode))
  ptr -> data = new_item
    if (head == NULL)
  {
    ptr -> next = NULL
     head = ptr
  print "node inserted"

  }
```

```
else
{
    temp = head
  while (temp -> next != NULL)
  { temp = temp -> next }
    temp -> next = ptr
    ptr -> next = NULL
  print " node inserted at last "
  }

}
  insert_at_pos()
{ struct node * ptr, * temp
   ptr -> data = new_item
    temp = head
   if (pos == 1)
{
   ptr -> next = temp
     head = ptr
     return
   }
   for (i=1; i<pos-1; i++)
{ temp = temp -> next }
   ptr -> next = temp -> next
     temp -> next = ptr
   }
```

Date: 23/ 11 /202

```
// for deletion
 delete_at_beginning()
{
 struct node * ptr
 if (head ==NULL)
 print " list is empty"

 else
  {
    ptr = head
    head = ptr ->next
    free (ptr)
  print " node deleted from beginning"
  }
}
  delete_at_end()
{ struct node *ptr, *ptr1
  if (head ==NULL)
  print " list is empty"

 else if (head -> next == NULL)
  {
    head = NULL
    free (head)
   print " node is deleted."
  }
```

Date: 23/11/201

```
else
 { ptr = head
  while (ptr -> next != NULL)
   { ptr1 = ptr
     ptr = ptr -> next }
    ptr1 -> next = NULL
     free (ptr)
    print "node deleted from last" }
}

delete_specified_data()
{ struct node *ptr, *ptr1
   ptr = head
  while (ptr != NULL && ptr -> data != item)
  {   ptr1 = ptr
      ptr = ptr -> next }
   print ptr -> data
     ptr1 -> next = NULL
       free (ptr)
   print " is deleted from the list"
}
```

Date: 23 / 11 / 2020

```
display ()
{ struct node *temp
    temp = head
  if (head == NULL)
   print " List is empty "

  else
  { while (temp -> next != NULL)
     { print temp -> data
          temp -> temp -> next }

     }

}
```