Date
Page

23|11|20 lab6, SUDESHNA BHUSHAN, 1BM19CS189

a) # Node creation in linked list

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

b) # Add a node at the front

```
void push(struct Node ** head_ref, int new_data)
{
    struct Node* new_node = (struct Node*)
                             malloc(sizeof
                                  (struct Node));
    new_node -> data = new_data;
    new_node -> next = (* head_ref);
    (* head_ref ) = new_node;
}
```

Add a node at the end

```
void append(struct Node ** head_ref,
            int new_data)
{
 struct Node * new_node = (struct Node*)
                    malloc (sizeof(struct Node));
 struct Node * last = * head_ref;
 new_node -> data = new_data;
new-node -> next = NULL;
if (*head-ref == NULL)
{
  *headref = new-node;
  return;
}
while (last -> next != NULL)
   last = last -> next;

last -> next = new-node;
return;
}
```

Add a node at specified position

```
void ins-at-pos-n(int data, int position)
{
  struct node *ptr = (struct node *)
                      malloc(sizeof(struct node));
  ptr-> data= data;
  int i;
  struct  node * temp = head;
  if (position == 1)
  {
    ptr -> next = temp;
       head = ptr;
       return;
  }
```

Date
Page

```
for (i=1; i< position-1; i++)
{
  temp = temp -> next;
}
ptr -> next = temp -> next;
temp -> next = ptr;
}
```

c) # Delete a node at the front

```
void pop()
{
  struct node *ptr;
  if (head==NULL)
  {
   printf("\n List is empty");
  }
  else
  {
    ptr = head;
    head = ptr -> next;
    free (ptr)
    printf ("\n Node deleted from the beginning
               ...");
  }
}
```

Data ______
Page ______

# <u>Delete a node at the end</u>

```
void end-delete ()
{
  struct node *ptr, *ptr1;
  if (head == NULL)
  {
    printf ("\n list is empty");
  }
  else if (head -> next == NULL)
  {
    head = NULL;
    free (head);
    printf ("\n Only ~~no~~ node of the list deleted...");
  }
  else
  {
    ptr = head;
    while (ptr -> next != NULL)
    {
      ptr1 = ptr;
      ptr = ptr -> next;
    }
    ptr1 -> next = NULL;
    free (ptr);
    printf ("\n Deleted Node from the last...");
  }
}
```

Date
Page

Delete a node at specified position

```
void deletespecified()
{
  struct node *ptr, *ptr1;
  int loc, i;
  scanf("%d", &loc);
  ptr=head;
  for (i=0; i<loc; i++)
  {
    ptrI = ptr;
    ptr = ptr -> next;
   if (ptr == NULL)
   {
    printf("\n There are less than %d elements
                in the list.. \n", loc);
   return;
   }
  }
 ptr1->next = ptr->next;
 free(ptr);
 print ("\n Deleted %d node", loc);
}
```

d) # Display the contents of Linked List

```
void printlist (struct Node * node)
{
  while (node != NULL)
  {
    printf ("%d", node -> data);
   node = node -> next;
  }
}
```

— × —— × —— × —