Date 23 NOV, 2020
Page

Name - Vedika Dalmia, USN - 1BM19CS181

WAP to create singly linked list with following operations

a) Create

b) Insert a node at 1st pos, at any pos and at end of list

c) Deletion of first element, specified element and last element in the list

d) Display the contents of linked list

```
void create()
{
   struct node *newnode, *temp;
   int item;
   newnode = (struct node*) malloc (sizeof (struct node));
   printf ("Enter the data");
   scanf ("%d" & item);
   newnode → data = item;
   if (head == NULL)
   {
        newnode → next = Null;
        head = newnode;
        printf (" Your Node is successfully
              created \n");
   }
   else
   {
      temp = head;
      while (temp → next = NULL)
      {
           temp = temp → next;
      }
      temp → next = newnode;
      newnode → next = NULL;
     printf (" Node created successfully at the
            end \n");
   }
}
```

Date
Page

```
void insert_first ()
{
        if (head == NULL)
          {
            create ();
            return;
          }
        structnode * newnode ;
        int item;
        printf ("Enter the element to insert at
                   first position\n");
         scanf ("%d", item);
        newnode = (structnode* )malloc (sizeof (struct node)));
        newnode → data = item;
        newnode → next = head;
         head = newnode ;
}
void insert_between (int pos)
{
        if (head == NULL)
          {
            insert_first ();
            return;
          }
      elseif ( pos> length )
          {
            create ();
            return;
          }
     struct node *newnode, *temp;
      temp = head;
      int item;
      printf ("enter data to be inserted \n");
       scanf ("%d", &item);
```

```
        int count = 1;
        while (count < (pos-1))
         {
             temp = temp -> next;
             count ++;
         }
    newnode = (structnode *) malloc (size-of (struct node));
    newnode -> data = item;
    newnode -> next = temp -> next;
    temp -> next = newnode;
  }
    int length ()
  {
      structnode * temp = head;
      int c = 0;
      while (temp -> next != NULL)
      {
          c++;
        temp = temp -> next;
      }
      return c+1;
  }
void delete_first ()
{
   if (head == NULL)
    {
       printf ("No elements present in list\n");
       return;
    }
   struct node *temp = head;
   head = head -> next;
   free (temp);
   printf ("element from first node deleted");
}
```

Date
Page

```
void delete_end (:)
{
    if (head = NULL)
    {
        printf (" Elements not present in list \n");
        return;
    }
    struct node *temp = head;
    while (temp → next → next != NULL)
    {
        temp=temp →next;
    }
    free (temp → next);
    temp → next = NULL;
    printf ("Element at the end deleted \n");
}
void delete_between (int pos)
{
    if (head = = NULL)
    {
        printf ("No element in list \n");
        return;
    }
    if (pos = = 1)
    {
        delete_first ();
        return;
    }
    if (pos > length ())
    {
        delete_end ();
        return;
    }
    struct node * prev = head;
    int count = 1;
```

```
        while (count < pos-1)
        {
            prev = prev → next;
            count ++;
        }
      struct node * temp = prev → next;
      prev → next = temp → next;
      free (temp);
      printf (" element at %d deleted \n", pos);
   }
   void display ()
   {
      struct node * ptr = NULL;
      ptr = head;
      if (ptr == NULL)
         printf (" No elements to print \n");
      else
      {
         printf ("List contents are : \n");
         while (ptr != NULL)
         {
            printf ("%d", ptr → data);
            ptr = ptr → next;
         }
         printf ("\n");
         }
      }
 }
```