PAGE No.
DATE 20/11/20

LAB 7

CHIRAG
1BM19CS039

```
#include<stdlib.h>
#include<stdio.h>

void create();
void display();
void insert_begin();
void insert_end()
void insert_pos()
void delete_begin();
void delete_end()
void delete_pos();

struct node
{
    int info;
    struct node *next;
};
struct node *start = NULL;
int main()
{
  int choice;
  while(1){
        printf(" MENU\n");
        printf("1.Create\n");
        printf("2.Display\n");
        printf("3. Insert at begining\n");
        printf("4. Insert at end\n");
        printf("5. Insert at specific position\n");
```

```
printf("\n 6. Delete from beginning");
printf("\n 7. Delete from end \n");
printf(" 8. Delete from specified position \n");
printf("\n 9. Exit \n");
printf("\n ---------- \n");
switch(choice
printf("Enter your choice :\t");
scanf("%d", &choice);
switch(choice)
{
  case 1:
        create();
        break;
  case 2:
        display()
        break;
  case 3:
        insert_begin();
        break;
  case 4:
        insert_end();
        break;
  case 5:
        insert_pos();
        break;
  case 6:
        delete_begin();
        break;
  case 7:
        delete_end();
        break;
```

CHIRAG

```
        case 8:
                delete_pos();
                break;
        case 9:
                exit(0);
                break;
        default
                printf("\n Wrong Choice:");
                break;
    }
    return 0;
}

void create()
{
    struct node *temp, *ptr;
    temp = (struct node *) malloc(sizeof(struct node));
    if (temp == NULL)
    {
        printf("\n Out of memory space:\n");
        exit(0);
    }
    printf("\n Enter the data value for the node:");
    scanf("%d", &temp->info);
    temp->next = NULL;
    if (start == NULL)
    {
        start = temp;
    }
    else
    {
        ptr = start;
```

```
        while( ptr -> next != NULL)
        {
            ptr = ptr -> next;
        }
        ptr -> next = temp;
    }
}

void display()
{
    struct node *ptr;
    if (start = NULL)
    {
        printf("\n List is Empty");
        return;
    }
    else
    {
        ptr = start
        printf("\n The list elements are:");
        while (ptr != NULL)
        {
            printf("\n%d", ptr->info);
            ptr = ptr->next;
        }
    }
}
```

CHIRAG

```
void insert_begin()
{
    struct node *temp;
    temp=(struct node *)malloc(sizeof(struct node));
    if(temp==NULL)
    {
        printf("\n Out of Memory Space");
        return
    }
    printf("\n Enter the data value for node: ");
    scanf("%d", &temp->info);
    temp->next=NULL;
    if(start==NULL)
    {
        start=temp;
    }
```

```
    else
    {
        temp -> next = start;
        start = temp;
    }

void insert_end()
{
    struct node *temp, *ptr;
    temp = (struct node *) malloc (sizeof (struct node));
    if (temp == NULL)
    {
        printf ("\n Out of memory Space:");
        return;
    }
    printf (" Enter value for node ");
    scanf ("%d", &temp -> info);
    temp -> next = NULL;
    if (start == NULL)
    {
        start = temp;
    }
    else
    {
        ptr = start;
        while (ptr -> next != NULL)
        {
            ptr = ptr -> next;
        }
        ptr -> next = temp;
    }
}
```

CHIRAG

```
void insert_pos()
{
  struct node *ptr, *temp;
  int i, pos;
  temp = (struct node *)malloc(sizeof(struct node));
  if (temp == NULL)
  {
    printf("\n Out of memory space");
    return;
  }
  printf("\n Enter position for new node :");
  scanf("%d", &pos);
  printf("\n Enter data value :");
  scanf("%d", &temp->info);
  temp->next = NULL;
  if (pos == 0)
  {
    temp->next = start;
    start = temp;
  }
  else
  {
    for (i=0, ptr = start; i<pos-1; i++)
    {
      ptr = ptr->next;
      if (ptr == NULL)
      {
        printf("Position not found:");
        return;
      }
    }
}
```

```
        temp->next = ptr->next;
        ptr->next = temp;
    }
}

void delete_begin()
{
    struct node *ptr;
    if (ptr == NULL)
    {
        printf("List is empty");
        return;
    }
    else
    {
        ptr = start;
        start = start->next;
        printf("The deleted element is %d", ptr->info);
        free(ptr);
    }
}

void delete_end()
{
    struct node *temp, *ptr;
    if (start == NULL)
    {
        printf("List is Empty:");
        exit(0);
    }
```

```
elsif (start -> next == NULL)
{
    ptr = start;
    start = NULL;
    printf (" The deleted element is %d", ptr->info);
    free(ptr);
}
else
{
    ptr = start;
    while (ptr -> next != NULL)
    {
        temp = ptr;
        ptr = ptr -> next;
    }
    temp -> next = NULL;
    printf (" The deleted element is %d", ptr -> info);
    free (ptr);
}
}

Void delete_pos()
{
  int i, pos;
  struct node *temp, *ptr;
  if (start == NULL)
  {
    printf ("The list is empty");
    exit(0);
  }
```

```
else
{
    printf("Enter the position to be deleted : ");
    scanf("%d", &pos);
    if (pos == 0)
    {
        ptr = start;
        start = start -> next;
        printf("The deleted element is %d", ptr -> info);
        free(ptr);
    }
    else
    {
        ptr = start;
        for(i=0; i<pos; i++)
        {
            temp = ptr; ptr = ptr -> next;
            if (ptr == NULL)
            {
                printf("Position not found.");
                return;
            }
        }
        temp -> next = ptr -> next;
        printf("The deleted element is %d", ptr -> info);
        free(ptr);
    }
}
```

CHIRAG
1BM19CS039