Date

Expt. No. 1BM19CS007 Aditi Akarsh Page No. 1

```
#include<stdlib.h>
#include<stdio.h>
void create();
void display();
void insertbegin();
void deletebegin();
void deleteend();
void deletepos();

struct node{ int id,sem;
             char name[20];
             int info;
             struct node* next; };
struct node* start=NULL;
int main()
{ int choice;
  while(1){ printf("1.Create\n2.Display\n3.Inse
      \n4.Delete from beginning 5.Delete from
      the end 6.Delete from position\n7.exit
  printf("Enter your choice");
  scanf("%d",&choice);
  switch(choice)
  { case 1: create();
            break;
  case 2; display(); break;
  case 3: insertbegin(); break;
  case 4: deletebegin(); break;
  case 5: deleteend(); break;
  case 6: deletepos(); break;
```

```
case 7: init(0);
        break;
default:
        printf("Wrong choice\n");
        break;
}
}
return 0;
}
void create()
{ struct node *temp, *ptr;
  temp=(struct node *)malloc(sizeof(struct
  if (temp==NULL)
{
  printf("Out of Memory Space\n");
    exit(0);
}
  printf("Enter input");
scanf("%d",&temp->id);
  temp->next=NULL;
  if(start==NULL)
  { start=temp; }
  else { ptr=start;
         while(ptr->next!=NULL)
         { ptr=ptr->next;
         }
       ptr->next=temp;
  }
}
```

Teacher's Signature

Expt. No. _____ Date _____

Page No. _____

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

Expt. No. 5 Aditi Alwarh 1BM19CS007 Date

Page No. 4

```
void insertbegin()
{ struct node *temp;
  temp = (struct node *) malloc (sizeof(struct node));
  if (temp == NULL)
  { printf ("Out of memory");
    return;
  }
  printf ("Enter id");
  scanf ("%d", &temp->id);
  printf ("Enter sem");
  scanf ("%d", &temp->sem);
  temp->next = NULL;
  if (start == NULL)
  { start = temp;
  }
  else
  { temp->next = start;
    start = temp;
  }
}
void insertend()
{
  struct node *temp, *ptr;
  temp = (struct node *) malloc (sizeof(struct
  if (temp == NULL)
  {
    printf ("Memory more");
    return;
  }
```

Teacher's Signature

```
else
{
    ptr = start;
    while (ptr -> next != NULL)
    { ptr = ptr -> next;
    }
    ptr -> next = temp;
}
}
void insertpos()
{
 struct node *ptr, *temp;
  int i, pos;
temp = (struct node *) malloc (sizeof (struct node));
 if (temp == NULL)
 { printf ("no memory");
   return;
 }
printf ("Enter position");
 scanf ("%d", &pos);
 printf ("Enter data");
scanf ("%d", &temp -> id);
if (pos == 0)
    { temp -> next = start; start = temp; }
else
 { for (i = 0, ptr = start; i < pos - 1; i++)
   { ptr = ptr -> next; }
   temp -> next = ptr -> next;
   ptr -> next = temp;
 }
```

Teacher's Signature