Code :

```
#include <stdio.h>
 #include <stdlib.h>
Struct Node
{
      int data;
      struct Node * next;
};

struct Node *head = NULL;
void    Createlist (int n)
 {
    Struct  Node * newNode , *temp ;
    int data;
     if (n <= 0)
     {
        printf("number of nodes should be greater than 0\n");
        return;
     }
     for(int i=1 ; i<=n; i++)
     {
        newNode = (struct Node*)malloc(sizeof (struct Node));
        if ( newNode == NULL)
        { printf(" memory allocation failed\n");
           return;
        }
        printf(" enter data:\n");
        scanf("%d", &data);
        newNode->data = data;
        newNode-> next = NULL;
        if ( head == NULL)
        {   head = newNode;
            ~~return;~~
        }
        else
            temp->next = newNode;

        temp = newNode;
   }
   printf(" created linked list \n");
```

```
void delete First ( )
{  struct Node * temp;
   if(head == NULL)
   { printf("Linked list not present\n");
     return;
   }
   temp = head;
   head = head->next;
   printf("deleted value is %d\n", temp->data);
   free(temp);
}
void delete End()
{  struct Node *temp, *prev;
   if (head == NULL)
   { printf ("Linked list not present\n");
     return;
   }
   if(head ->next==NULL)
   {
     printf ("deleted value is %d \n", head->data);
     free(head);
     head = NULL;
     return;
  }
   temp=head;
   while (temp->next != NULL)
   { prev = temp;
       temp = temp->next;
   }
   prev->next = NULL;
   printf("deleted value is %d\n", temp->data);
   free(temp);
}
void delete Specific (int val)
{   struct Node * temp = head;
    struct Node * prev = NULL;
    if(head == NULL)
    { printf("Linked list not present\n");
      return;
    }
    if (head->data == val)
    { head = head->next;
      printf ("deleted value is %d\n", temp->data);
      free(temp);
      return;
    }
    while (temp! = NULL && temp->data! = val)
    { prev = temp;
      temp = temp->next;
    }
    if (temp == NULL)
       printf(" specified value not present in linked list\n");
    prev->next = temp->next;
    printf ("deleted value is %d\n", temp->data);
    free(temp);
}
void display()
{  Struct Node * temp=head;
   if (head == NULL)
   { printf (" List is empty\n");
     return;
   }
   while (temp! = NULL)
   { printf ("%d->", temp->data);
     temp = temp->next;
   }
}
int main()
{  int ch, n, val;
   do {
     printf ("1. Create Linked List , 2.delete First, 3.delete End,
              4. delete at specific ., 5.display \n");
     printf ("enter choice: \n");
     scanf ("%d", &ch);
     switch(ch) {
       case 1 : { printf ("enter number of nodes: ");
                  scanf(" %d", &n);
                  createList(n);
                  break; }
       case 2: { deleteFirst( );
                 break; }
       case 3: { deleteEnd( );
                 break; }
       case 4: { printf ("enter specific value: \n");
                 scanf ("%d", &val);
                 deleteSpecific(val);
                 break; }
       case 5: { display( );
                 break; }
       default: printf(" invalid choice\n");
     }
   }while (ch! = 0);
   return 0;
}
```

Output:

```
1. create linked list , 2.deleteFirst , 3.delete End , 4.delete at
 specific , 5.display .
enter choice:
1
enter number of nodes : 4
enter data:
10
enter data:
20
enter data:
30
enter data:
40
created linked list
1. create linked list, 2.delete First , 3.delete End , 4.delete at specific,
                                                                 5.display
enter choice : 2
deleted value is 10
1. create linked list 2. delete First, 3.delete End, 4.delete at specific, 5.display
enter choice: 3
deleted value is 40
1. Create linked list . . . .
enter choice : 4
enter specific value : 20
deleted value is 20
1. Create . . . .
enter choice: 5
30 ->
```

O/P