Doubly Linked List

Derek Stanley Kannath
1BM19CS041

```
struct node
{ int data;
  struct node *next;
  struct node *prev;
};

void insert_left()
{ struct node *new_node;
  new_node = (struct node *) malloc (sizeof (struct node));
  printf("Enter the item:\n");
  scanf("%d", &new_node -> data);
  new_node -> next = NULL;
  new_node -> prev = NULL;
  if (head == NULL)
  { head = new_node; }
  else
  { new_node -> next = head;
    head -> prev = new_node
    head = new_node;
  }
}

void display()
{ struct node *ptr;
  ptr = head;
  while (ptr != NULL)
  { printf("%d\t", ptr -> data);
    ptr = ptr -> next; }
  printf("\n");
}
```


Derek

Derek Stanley Kannathu
1BM19CE045

```
void del()
{ struct node *temp;
  int ele;
  if (head == NULL)
  { printf("List is Empty.\n");
    return; }
  printf("\n Enter the element to be deleted \n");
  scanf("%d", &ele);
  temp = head;
  while (temp -> data != ele)
  { temp = temp -> next
    if(temp == NULL)
    { printf("\n Element is not in the list.\n");
      break;
    }
  }
  if (temp == head)
  { head = head -> next; }
  else if (temp -> next == NULL)
  { temp = temp -> prev;
    temp -> next = NULL;
  }
  else
  { temp -> prev -> next = temp -> next;
    temp -> next -> prev = temp -> prev;
  }
}
```


Derek

Derek Stanley Kanwar
1BM19CS044

```
void insert_right()
{ struct node *new_node, *temp;
  new_node = (struct node *) malloc (sizeof(struct node));
  printf("\n Enter the element : \n");
  scanf("%od", &new_node -> data);
  new_node -> next = NULL;
  new_node -> prev = NULL;
  if (head == NULL)
   { head = new_node; }
  else
  { temp = head;
    while (temp -> next != NULL)
      temp = temp -> next;
    temp -> next = new_node;
    new_node -> prev = temp;
  }
}
```