Queue - Stack (SLL) Operations

1BM19CS048
Derek Stanley Kanrath

```
void qinsert()
{ struct node *newnode;
  newnode = (struct node *) malloc (sizeof(struct node));
  printf(" Enter the element : \n ");
  scanf(" %od", &newnode -> data);
  newnode -> next = NULL;
  if (rear == NULL)
  { rear = newnode;
    front = newnode; }
  else
  { rear -> next = newnode;
    rear = newnode; }
}
void qdel;
{ if ( front == NULL)
  { printf(" Queue is empty \n");
    return ; }
  else
  { printf (" Deleted ele is %od", front -> data );
  ; if ( front == rear ) {
    { printf(" Queue is empty \n");
      front = NULL ; rear = NULL;
    }
    else
      front = front -> next;
  }
}
```

Derek Stanley Kannathe
1BM19CS045

```
void qdisplay()
{   struct node *temp;
    if(front==NULL)
    {  printf("Queue Empty\n");
       return; }
    temp = front;
    while(temp != NULL)
    { printf("%d", temp->data);
      temp = temp->next;
     }
}

void push()
{  int ele;
   struct node *newnode;
   printf("Enter the element\n");
   scanf("%d",&ele);
   newnode = (struct node *)malloc(sizeof(struct node));
   newnode->data = item;
   newnode->next = NULL;
   if(top==NULL)
       top= newnode;
    else
     { newnode->next = top;
       top = newnode; }
}
```


Derek

Derek Stanley
Kannathe
IBM19CS045

```
void spop()
{  if(top==NULL)
      printf("\n Stack is empty");
   else
   { printf("\n Element removed is %od", top->data);
     top = top->next; }
}
void sdisplay()
{  struct node *temp;
   temp = top;
   if(top==NULL)
      printf("\n Stack is empty");
   while(temp!=NULL)
   { printf("\n %od", temp->data);
     temp = temp->next;
   }
}
```


Derek