Vaishwanth Kiran S
1BM19CS187

# LAB - 8

## Linked - Stack

```
Struct Stack
{
  int data;
  Struct Stack *next;
};
Struct Stack *top = NULL;

void push() {
Struct stack *ptr;
ptr = (Struct Stack *) malloc (Size of (Struct Stack));
Ptr -> data = val; // val is from user
Ptr -> next = NULL;
if (top == NULL)
{
  top = ptr;
}
else
{
  ptr -> next = top;
  top = ptr;
}
}

void display() {
  Struct stack *ptr;
  ptr = top;
  if (top == NULL)
    Printf("stack is empty");
```

```
else
{
    while(ptr != NULL)
    {
        printf("%d \t", ptr->data);
        ptr = ptr->next;
    }
}
}

void pop(){
    Struct Stack *ptr;
    ptr = top;
    if (top == NULL)
        printf("Stack underflow");
    else {
        top = top->next;
        printf("The value being deleted is: %d",
        free(ptr);                    ptr->data);
    }
}

int peep()
{
    return top->data;
}
```

## linked queue

```
struct node {
    int data;
    struct node *next;
};
node *root = NULL
```

```
void insert() {
    struct node *ptr;
    ptr = (struct node *) malloc (sizeof (struct node));
    ptr -> data = val;

    ptr -> next = NULL;
    if (root == NULL)
    {
        root = ptr;
    }
    else
    {
    struct node *p = root;
        while (p -> next != NULL)
        {
            p = p -> next;
        }
} } p -> next = ptr;
```

```
void dequeue()
{
struct node *temp;
  if (root == NULL)
   {
    printf("Queue is empty");
   }
  else
  {
    temp = root;
    root = temp->next;
    temp->next = Null;
     free(temp);
  }
}
void display() { struct node *temp=ro
   while (temp != NULL)
    {
     printf("%d \t", temp->data);
      temp = temp->next;
    }
  }
}
```