struct node

{

int data;

struct node \*next;

};

struct queue

{

struct node \*front;

struct node \*rear;

};

struct queue \*q;

void create\_queue(struct queue \*);

struct queue \*insert(struct queue \*,int);

struct queue \*delete\_element(struct queue \*);

struct queue \*display(struct queue \*);

int peek(struct queue \*);

void create\_queue(struct queue \*q)

{

q -> rear = NULL;

q -> front = NULL;

}

struct queue \* insert(struct queue \*q,int val)

{

struct node \*ptr;

ptr = (struct node\*)malloc(sizeof(struct node));

ptr -> data = val;

if(q -> front == NULL)

{

q -> front = ptr;

q -> rear = ptr;

q -> front -> next = q -> rear -> next = NULL;

}

else

{

q -> rear -> next = ptr;

q -> rear = ptr;

q -> rear -> next = NULL;

}

return q;

}

struct queue \* display(struct queue \*q)

{

struct node \*ptr;

ptr = q -> front;

if(ptr == NULL)

printf("\n QUEUE IS EMPTY");

else

{

printf("\n");

while(ptr!=q -> rear)

{

printf("%d\t", ptr -> data);

ptr = ptr -> next;

}

printf("%d\t", ptr -> data);

}

return q;

}

struct queue \* delete\_element(struct queue \*q)

{

struct node \*ptr;

ptr = q -> front;

if(q -> front == NULL)

printf("\n UNDERFLOW");

else

{

q -> front = q -> front -> next;

printf("\n The value being deleted is : %d", ptr -> data);

free(ptr);

}

return q;

}

int peek(struct queue \*q)

{

if(q->front==NULL)

{

printf("\n QUEUE IS EMPTY");

return –1;

}

else

return q->front->data;

}