Name: Bankar Krushna Lahanubhau
Batch : S2
Division : 1
MIS : 612303031

Main.c

```
#include <stdio.h>
#include "queue.h"


void printMenu() {
   printf("Enter choice:\n");
   printf("1. init\n");
   printf("2. enqueue\n");
   printf("3. dqueue\n");
   printf("4. Is empty\n");
   printf("5. Display\n");
   printf("6. Exit\n");
   return;
}

int main() {

   queue q;
   int choice;
   while(1) {
      printMenu();
      scanf("%d", &choice);

      switch(choice) {
         case 1:
            initq(&q);
            break;
         case 2:
            int data;
            printf("Enter element: \n");
            scanf("%d", &data);
            enqueue(&q, data);
            break;
         case 3:
            printf("%d\n", dqueue(&q));
            break;
         case 4:
            if(is_empty(q)) {
               printf("Queue is Empty!\n");
            }
            else {
               printf("Queue is not Empty!\n");
            }
            break;
         case 5:
```

```
            traverse(q);
            break;
        case 6:
            return 0;
        default:
            printf("Enter valid choice\n");
      }
   }


   return 0;
}
```

queueu.c

```
#include <stdio.h>
#include <stdlib.h>
#include "queue.h"
void initq(queue *q) {
   q->head = NULL;
   q->tail = NULL;
   return;
}

int dqueue(queue *q) {

   if(is_empty(*q)) return -1;

   int data = q->head->data;

   node *p = q->head;
   q->head = q->head->next;
   free(p);

   return data;
}

void enqueue(queue *q, int data) {

   node *nn = (node *)malloc(sizeof(node));

   if(nn) {
      nn->data = data;
      nn->next = NULL;
   }

   if(is_empty(*q)) {
      q->head = nn;
      q->tail = nn;
      return;
```

```
    }

    q->tail->next = nn;
    q->tail = nn;

    return;
}

void traverse(queue q) {

    if(is_empty(q)) {
        printf("[]\n");
        return;
    }

    node *p;
    p = q.head;
    printf("[");
    while(p) {
        printf("%d ", p->data);
        p = p->next;
    }

    printf("]\n");
    return;
}
int is_empty(queue q) {
    return q.head == NULL;
}
```

quueue.h

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

typedef struct queue {
    node *head, *tail;
}queue;

void initq(queue *q);
int dqueue(queue *q);
void enqueue(queue *q, int data);
void traverse(queue q);
int is_empty(queue q);
```

output:

```
labwork git:(main) _ gcc -Wall -c queue.c queue.h main.c
labwork git:(main) _ cc main.o queue.o -o queue
labwork git:(main) _ ./queue
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
1
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
5
[]
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
2
Enter element:
56
```

```
56
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
5
[56 ]
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
2
Enter element:
34
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
5
[56 34 ]
Enter choice:
1. init
```

```
5. Display
6. Exit
5
[56 34 ]
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
3
56
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
4
Queue is not Empty!
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
6
```