

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. dequeue\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", dequeue(&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;
}

```

queue.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:

```

ADClabwork git:(main) elogccn-Wallw-Coqueue!cs queue.h?main.c Templates
Deslabwork git:(main) umeccs main?oA queue.o Po queue Public Videos
→ labwork git:(main) _ ./queue 1, -x 2.0 -x notebook 1 (0.2.3)
Enter choice: ready satisfied pure eval in ./local/lib/python3.12/site
1. init -x jupyterlab 4.3, -x 2.0 -x notebook 1 (0.2.3)
2. enqueue already satisfied types python-dateutil 2.8.1 in ./local
3. dqueue on-x jsonschema format non-gpl 4.18.0 -x jupyter-events 0.9.0-
4. Is empty
5. Display PATH HOME
6. Exit
1 -x jupyterlab notebook
Enter choice: ready satisfied jupyter lsp | extension was success
1. init -x jupyterlab jupyter server terminals | extensi
2. enqueue -x jupyterlab jupyterlab | extension was success
3. dqueue -x jupyterlab notebook | extension was successful
4. Is empty -x jupyterlab Writing Jupyter server cookie sec
5. Display is secret
6. Exit -x jupyterlab notebook shim | extension was suc
5 -x jupyterlab notebook shim | extension was suc
[]
Enter choice: ready satisfied jupyter server terminals | extensi
1. init -x jupyterlab JupyterLab extension loaded from /ho
2. enqueue
3. dqueue -x jupyterlab JupyterLab application directory is
4. Is empty -x jupyterlab Extension Manager is 'pypi'
5. Display -x jupyterlab jupyterlab | extension was success
6. Exit -x jupyterlab notebook | extension was successful
2 -x jupyterlab Serving notebooks from local dire
Enter element: -x jupyterlab Jupyter Server 2.14.2 is running
56

```

```

56 ~$ ls -l | grep already satisfied
Enter choice: ion Development
1. sinit -x already Documents
2. enqueue -x jupyterlab
3. dqueue already satisfied
4. Is empty jupyterlab 4.3.0
5. Display already satisfied
6. Exit on-x jsonschema format
5
[56 ] PATH HOME
Enter choice:
1. init notebook
2. enqueue
3. dqueue
4. Is empty
5. Display
6. Exit
2 is secret
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 already satisf
6. Exitimulation Development
5 desktop -x already Docume
[56~34 ]
Enter choice: ready satisf
1. init jupyterlab
2. enqueue already satisf
3. dqueue on-x jsonschema
4. Is empty
5. Display PATH HOME
6. Exit
3 -x jupyterlab notebook
56
Enter choice:
1. init
2. enqueue
3. dqueue
4. Is empty type secret
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

```