**USN:- 1NT21IS039**

## Aim:- ORDINARY QUEUE IMPLEMENTATION USING ARRAY

[GitHub-Link](https://github.com/logan-14/Queue-Using-LinkedList)

**ALGORITHM:-**

**enqueue(data)**

* Create a struct node type node.
* Insert the given data in the new node data section and NULL in address section.
* If Queue is empty then initialize front and rear from new node.
* Queue is not empty then initialize rear next and rear from new node.
* New node next initialize from front

**dequeue()**

* + Check if queue is empty or not.
  + If queue is empty then dequeue is not possible.
  + Else Initialize temp from front.
  + If front is equal to the rear then initialize front and rear from null.
  + Print data of temp and free temp memory.
  + If there is more than one node in Queue then make front next to front then initialize rear next from front.
  + Print temp and free temp.

**print()**

* Check if there is some data in the queue or not.
* If the queue is empty print “No data in the queue.”
* Else define a node pointer and initialize it with front.
* Print data of node pointer until the next of node pointer becomes NULL.

**Code:-**

#include <stdio.h>

#include <stdlib.h>

struct node

{

    int data;

    struct node \*next;

};

struct node \*f = NULL;

struct node \*r = NULL;

void enqueue(int d) // Insert elements in Queue

{

    struct node \*n;

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

    n->data = d;

    n->next = NULL;

    if ((r == NULL) && (f == NULL))

    {

        f = r = n;

        r->next = f;

    }

    else

    {

        r->next = n;

        r = n;

        n->next = f;

    }

}

void dequeue() // Delete an element from Queue

{

    struct node \*t;

    t = f;

    if ((f == NULL) && (r == NULL))

        printf("\nQueue is Empty");

    else if (f == r)

    {

        f = r = NULL;

        free(t);

    }

    else

    {

        f = f->next;

        r->next = f;

        free(t);

    }

}

void print()

{ // Print the elements of Queue

    struct node \*t;

    t = f;

    if ((f == NULL) && (r == NULL))

        printf("\nQueue is Empty");

    else

    {

        do

        {

            printf("\n%d", t->data);

            t = t->next;

        } while (t != f);

    }

}

int main()

{

    int opt, n, i, data;

    printf("Enter Your Choice:-");

    do

    {

        printf("\n\n1 for Insert the Data in Queue\n2 for show the Data in Queue \n3 for Delete the data from the Queue\n0 for Exit");

        scanf("%d", &opt);

        switch (opt)

        {

        case 1:

            printf("\nEnter the number of data");

            scanf("%d", &n);

            printf("\nEnter your data");

            i = 0;

            while (i < n)

            {

                scanf("%d", &data);

                enqueue(data);

                i++;

            }

            break;

        case 2:

            print();

            break;

        case 3:

            dequeue();

            break;

        case 0:

            break;

        default:

            printf("\nIncorrect Choice");

        }

    } while (opt != 0);

    return 0;

}

**Output:-**

Enter Your Choice:-

1 for Insert the Data in Queue

2 for show the Data in Queue

3 for Delete the data from the Queue

0 for Exit1

Enter the number of data1

Enter your data10

1 for Insert the Data in Queue

2 for show the Data in Queue

3 for Delete the data from the Queue

0 for Exit1

Enter the number of data1

Enter your data20

1 for Insert the Data in Queue

2 for show the Data in Queue

3 for Delete the data from the Queue

0 for Exit1

Enter the number of data1

Enter your data30

1 for Insert the Data in Queue

2 for show the Data in Queue

3 for Delete the data from the Queue

0 for Exit1

Enter the number of data1

Enter your data40

1 for Insert the Data in Queue

2 for show the Data in Queue

3 for Delete the data from the Queue

0 for Exit2

10

20

30

40

1 for Insert the Data in Queue

2 for show the Data in Queue

3 for Delete the data from the Queue

0 for Exit3

1 for Insert the Data in Queue

2 for show the Data in Queue

3 for Delete the data from the Queue

0 for Exit3

1 for Insert the Data in Queue

2 for show the Data in Queue

3 for Delete the data from the Queue

0 for Exit2

30

40

1 for Insert the Data in Queue

2 for show the Data in Queue

3 for Delete the data from the Queue

0 for Exit