
Data Structure Lab

CEN-391

Program 9

Code :-

```
#include <iostream>
using namespace std;

struct Node
{
    int data;
    Node *next;
};

void
isEmpty(int size)
{
    cout << "isEmpty...\n";
    if (size == 0)
        cout << "Empty" << endl;
    else
```

```

        cout << "Not Empty" << endl;
    }

void Display(Node *head, int size)
{
    cout << "Display...\n";
    if (size == 0)
    {
        cout << "Queue Is Empty" << endl;
        return;
    }
    while (head != nullptr)
    {
        cout << head->data << " ";
        head = head->next;
    }
    cout << endl;
}

void Enqueue(Node *&head, Node *&tail, int &size)
{
    cout << "Enqueue...\n";
    size++;
    Node *newnode = (Node *)malloc(1 * sizeof(Node));
    if (newnode == nullptr)
    {
        cout << "Memory Not Assigned" << endl;
        return;
    }
    cout << "Enter The Element : ";
    int val;
    cin >> val;
    newnode->data = val;
    newnode->next=nullptr;
    if (head != nullptr)
    {
        tail->next = newnode;
        tail = tail->next;
    }
}

```

```

        else
        {
            head = newnode;
            tail = newnode;
        }
        Display(head, size);
    }

void Dequeue(Node *&head, int &size)
{
    cout << "Dequeue...\n";
    if (size == 0)
    {
        cout << "Queue Underflow" << endl;
        return;
    }
    cout << head->data << endl;
    size--;
    Node *todelete = head;
    head = head->next;
    delete todelete;
    Display(head, size);
}

void Front_Rear(Node *head, Node *tail, int size)
{
    cout << "Front And Rear...\n";
    if (size == 0)
    {
        cout << "Queue Is Empty" << endl;
        return;
    }
    cout << "Front : " << head->data << endl;
    cout << "Rear : " << tail->data << endl;
}

void Total_Element(int size)
{
    cout << "Total Elements In Queue : " << size << endl;
}

```

```

void Bars()
{
    cout << "-----\n";
}
bool Options(Node *&head, Node *&tail, int &size)
{
    int opt;
    cin >> opt;
    Bars();
    switch (opt)
    {
        case 1:
            Enqueue(head, tail, size);
            break;
        case 2:
            Dequeue(head, size);
            break;
        case 3:
            Front_Rear(head, tail, size);
            break;
        case 4:
            isEmpty(size);
            break;
        case 5:
            Total_Element(size);
            break;
        case 6:
            Display(head, size);
            break;
        case 7:
            cout << "Exit...\n";
            return 0;
        default:
            cout << "Invalid Input!\nTry Again!\n";
    }
    Bars();
    return 1;
}

```

```

}

void Menu()
{
    cout << "_____Operations_On_Queue_____ \n";
    cout << "1.Enqueue \n";
    cout << "2.Dequeue \n";
    cout << "3.Front And Rear Element \n";
    cout << "4.isEmpty \n";
    cout << "5.Total Elements \n";
    cout << "6:Display \n";
    cout << "7.Exit \n";
    cout << "Enter Your Choice : ";
}

int main()
{
    system("cls");
    cout << "_____Vicky_Gupta_20BCS070_____ \n\n";
    int size = 0;
    Node *head = nullptr, *tail = nullptr;
    while (true)
    {
        Menu();
        if (!Options(head,tail,size))
            break;
    }
    cout << "Exiting...\n";
    Bars();
    return 0;
}

```

Output :-

```
_____Vicky_Gupta_20BCS070_____
```

```
_____Operations_On_Queue_____
```

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 1

Enqueue...

Enter The Element : 33

Display...

33

```
_____Operations_On_Queue_____
```

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 1

Enqueue...

Enter The Element : 22

Display...

33 22

_____Operations_On_Queue_____

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 1

Enqueue...

Enter The Element : 11

Display...

33 22 11

_____Operations_On_Queue_____

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 3

Front And Rear...

Front : 33

Rear : 11

_____Operations_On_Queue_____

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 2

Dequeue...

33

Display...

22 11

_____Operations_On_Queue_____

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 5

Total Elements In Queue : 2

_____Operations_On_Queue_____

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 2

Dequeue...

22

Display...

11

_____Operations_On_Queue_____

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 6

Display...

11

_____Operations_On_Queue_____

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 6

Display...

11

_____Operations_On_Queue_____

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 2

Dequeue...

11

Display...

Queue Is Empty

_____Operations_On_Queue_____

- 1.Enqueue
- 2.Dequeue
- 3.Front And Rear Element
- 4.isEmpty
- 5.Total Elements
- 6:Display
- 7.Exit

Enter Your Choice : 7

Exit...

Exiting...
