\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Data Structure Lab

CEN-391

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Program 7

Code :-

#include <iostream>

using namespace std;

void isEmpty(int size)

{

    cout << "isEmpty...\n";

    if (size == -1)

        cout << "Empty" << endl;

    else

        cout << "Not Empty" << endl;

}

void isFull(int size, int capacity)

{

    cout << "isFull...\n";

    if (size + 1 == capacity)

        cout << "Full" << endl;

    else

        cout << "Not Full" << endl;

}

void Display(int queue[], int size)

{

    cout << "Display...\n";

    if (size == -1)

    {

        cout << "Queue Is Empty" << endl;

        return;

    }

    for (int i = 0; i <= size; i++)

    {

        cout << queue[i] << " ";

    }

    cout << endl;

}

void Enqueue(int queue[], int &size, int capacity)

{

    cout << "Enqueue...\n";

    size++;

    if (size == capacity)

    {

        size--;

        cout << "Queue Overflow" << endl;

        return;

    }

    cout << "Enter The Element : ";

    cin >> queue[size];

    Display(queue, size);

}

void Dequeue(int queue[], int &size)

{

    cout << "Dequeue...\n";

    if (size == -1)

    {

        cout << "Queue Underflow" << endl;

        return;

    }

    cout<<queue[0]<<endl;

    for (int i = 1; i <= size; i++)

    {

        queue[i - 1] = queue[i];

    }

    size--;

    Display(queue, size);

}

void Front\_Rear(int queue[], int size)

{

    cout << "Front And Rear...\n";

    if (size == -1)

    {

        cout << "Queue Is Empty" << endl;

        return;

    }

    cout << "Front : " << queue[0] << endl;

    cout << "Rear : " << queue[size] << endl;

}

void Total\_Element(int size)

{

    cout << "Total Elements In Queue : " << size + 1 << endl;

}

void Bars()

{

    cout << "---------------------------------------------------------------\n";

}

bool Options(int queue[], int &size, int capacity)

{

    int opt;

    cin >> opt;

    Bars();

    switch (opt)

    {

    case 1:

        Enqueue(queue, size, capacity);

        break;

    case 2:

        Dequeue(queue, size);

        break;

    case 3:

        Front\_Rear(queue, size);

        break;

    case 4:

        isEmpty(size);

        break;

    case 5:

        isFull(size, capacity);

        break;

    case 6:

        Total\_Element(size);

        break;

    case 7:

        Display(queue, size);

        break;

    case 8:

        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.isFull \n";

    cout << "6.Total Elements \n";

    cout << "7:Display \n";

    cout << "8.Exit \n";

    cout << "Enter Your Choice : ";

}

int main()

{

    system("cls");

    cout << "\_\_\_\_\_Vicky\_Gupta\_20BCS070\_\_\_\_\_\n\n";

    cout << "Enter The Size Of The Queue : ";

    int capacity, size = -1;

    cin >> capacity;

    int queue[capacity] = {0};

    cout << "\n\n";

    while (true)

    {

        Menu();

        if (!Options(queue, size, capacity))

            break;

    }

    cout << "Exiting...\n";

    Bars();

    return 0;

}

Output :-

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated