Manav Rachna International Institute of Research and Studies Bachelor's in computer applications

Data Structures using C

Submitted by: Urvashi Pahuja

Department: School Of Computer Applications

Course: Bcahelors in Computers 1pplications

Roll no. : 24/SCA/BCA/087

Semester: 2nd

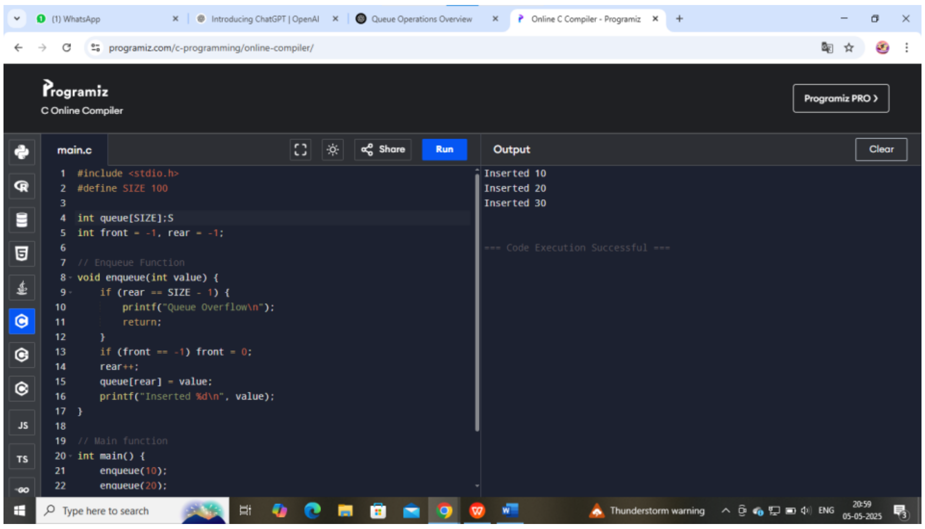
Subject: Data Structures Using C

DS FILE

AIM1: ENQUE OR THE INSERTION IN QUEUES

```
 #include <stdio.h>
#define SIZE 100

int queue[SIZE];
int front = -1, rear = -1;

// Enqueue Function
void enqueue(int value) {
    if (rear == SIZE - 1) {
        printf("Queue Overflow\n");
        return;
    }
    if (front == -1) front = 0;
    rear++;
    queue[rear] = value;
    printf("Inserted %d\n", value);
}

// Main function
int main() {
    enqueue(10);
    enqueue(20);
    enqueue(30);
    return 0;
}
```

OUTPUT:



```
#include <stdio.h>
#define SIZE 100

int queue[SIZE];S
int front = -1, rear = -1;

// Enqueue Function
void enqueue(int value) {
    if (rear == SIZE - 1) {
        printf("Queue Overflow\n");
        return;
    }
    if (front == -1) front = 0;
    rear++;
    queue[rear] = value;
    printf("Inserted %d\n", value);
}

// Main function
int main() {
    enqueue(10);
    enqueue(20);
```

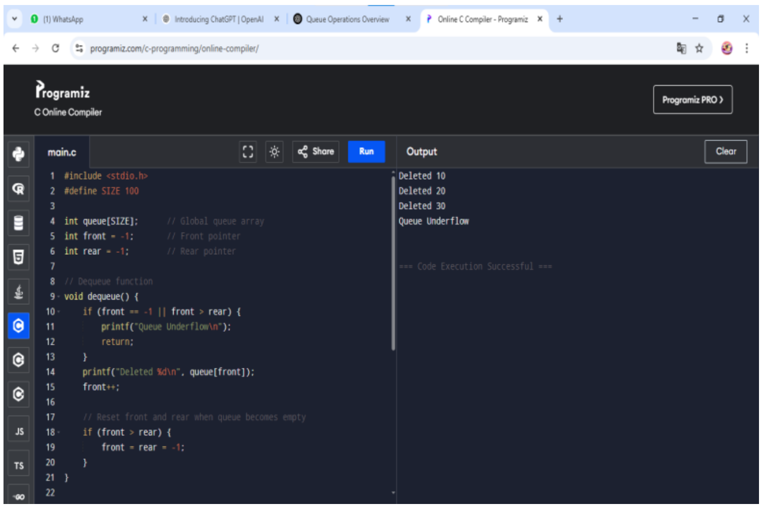
```
Inserted 10
Inserted 20
Inserted 30


=== Code Execution Successful ===
```

AIM 2: DEQUUE OR DELETION IN QUEUES.

```
#include <stdio.h>
#define SIZE 100

int queue[SIZE];      // Global queue array
int front = -1;       // Front pointer
int rear = -1;        // Rear pointer

// Dequeue function
void dequeue() {
    if (front == -1 || front > rear) {
        printf("Queue Underflow\n");
        return;
    }
    printf("Deleted %d\n", queue[front]);
    front++;

    // Reset front and rear when queue becomes
empty
```

```
    if (front > rear) {
        front = rear = -1;
    }
}

// Main function to test
int main() {
    // Simulate inserting some elements
    rear = 2;
    front = 0;
    queue[0] = 10;
    queue[1] = 20;
    queue[2] = 30;

    dequeue();  // Should delete 10
    dequeue();  // Should delete 20
    dequeue();  // Should delete 30
    dequeue();  // Should print underflow

    return 0;
}
```

OUTPUT:



```
#include <stdio.h>
#define SIZE 100

int queue[SIZE];      // Global queue array
int front = -1;       // Front pointer
int rear = -1;        // Rear pointer

// Dequeue function
void dequeue() {
    if (front == -1 || front > rear) {
        printf("Queue Underflow\n");
        return;
    }
    printf("Deleted %d\n", queue[front]);
    front++;

    // Reset front and rear when queue becomes empty
    if (front > rear) {
        front = rear = -1;
    }
}

```

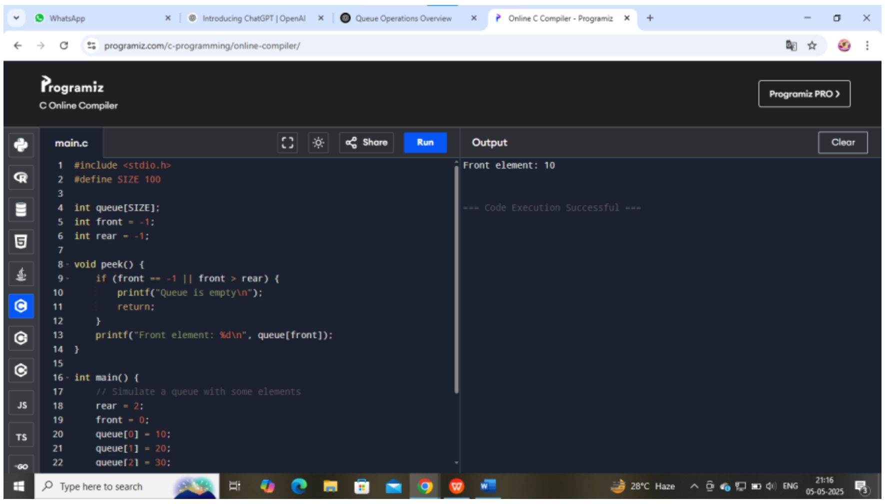
```
Deleted 10
Deleted 20
Deleted 30
Queue Underflow


=== Code Execution Successful ===
```

AIM 3: PEEK FUNCTION IN QUEUES.

```
#include <stdio.h>

#define SIZE 100


int queue[SIZE];

int front = -1;

int rear = -1;


void peek() {

    if (front == -1 || front > rear) {

        printf("Queue is empty\n");

        return;

    }
```

```c
    printf("Front element: %d\n", queue[front]);
}

int main() {
    // Simulate a queue with some elements
    rear = 2;
    front = 0;
    queue[0] = 10;
    queue[1] = 20;
    queue[2] = 30;

    peek();  // Should print: Front element: 10

    return 0;
}
```

OUTPUT:


WhatsApp
Introducing ChatGPT | OpenAI
Queue Operations Overview
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Programiz
C Online Compiler
Programiz PRO >
main.c
Share
Run
Output
Clear
Type here to search
28°C Haze
ENG
21:16
05-05-2025

```c
#include <stdio.h>
#define SIZE 100

int queue[SIZE];
int front = -1;
int rear = -1;

void peek() {
    if (front == -1 || front > rear) {
        printf("Queue is empty\n");
        return;
    }
    printf("Front element: %d\n", queue[front]);
}

int main() {
    // Simulate a queue with some elements
    rear = 2;
    front = 0;
    queue[0] = 10;
    queue[1] = 20;
    queue[2] = 30;
```

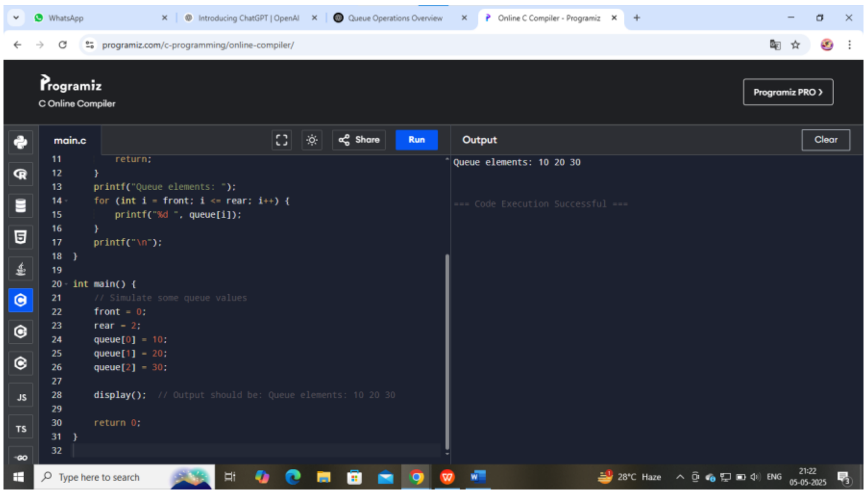
```
Front element: 10


=== Code Execution Successful ===
```

<u>AIM 4:</u> DISPLAY FUNCTION IN QUEUES.

```
#include <stdio.h>
#define SIZE 100

int queue[SIZE];
int front = -1;
int rear = -1;

void display() {
    if (front == -1 || front > rear) {
        printf("Queue is empty\n");
        return;
    }
    printf("Queue elements: ");
    for (int i = front; i <= rear; i++) {
        printf("%d ", queue[i]);
    }
    printf("\n");
}
```

```c
int main() {
    // Simulate some queue values
    front = 0;
    rear = 2;
    queue[0] = 10;
    queue[1] = 20;
    queue[2] = 30;

    display();  // Output should be: Queue elements: 10 20 30

    return 0;
}
```

OUTPUT:



WhatsApp
Introducing ChatGPT | OpenAI
Queue Operations Overview
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Programiz
C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear
Type here to search
28°C Haze
ENG
21:22
05-05-2025

```c
            return;
    }
    printf("Queue elements: ");
    for (int i = front; i <= rear; i++) {
        printf("%d ", queue[i]);
    }
    printf("\n");
}

int main() {
    // Simulate some queue values
    front = 0;
    rear = 2;
    queue[0] = 10;
    queue[1] = 20;
    queue[2] = 30;

    display();  // Output should be: Queue elements: 10 20 30

    return 0;
}
```

```
Queue elements: 10 20 30


=== Code Execution Successful ===
```