3a. WAP to simulate the working of a queue of integers using an array. Provide the following operations: Insert, Delete, Display The program should print appropriate messages for queue empty and queue overflow conditions

```c
#include <stdio.h>
#define MAX 5
int queue[MAX];
int front = -1, rear = -1;
void insert(int value) {
if (rear == MAX - 1) {
printf("Queue Overflow! Cannot insert %d\n", value);
} else {
if (front == -1) {
front = 0;
}
rear++;
queue[rear] = value;
printf("%d inserted into the queue.\n", value);
}
}
void delete() {
if (front == -1 || front > rear) {
printf("Queue Underflow! Queue is empty.\n");
} else {
printf("Deleted element: %d\n", queue[front]);
front++;
}
}
void display() {
if (front == -1 || front > rear) {
printf("Queue is empty.\n");
} else {
printf("Queue elements: ");
for (int i = front; i <= rear; i++) {
printf("%d ", queue[i]);
}
printf("\n");
}
}
int main() {
int choice, value;
while (1) {
printf("\nQueue Operations:\n");
printf("1. Insert\n");
printf("2. Delete\n");
printf("3. Display\n");
printf("4. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
```

```c
switch (choice) {
case 1:
printf("Enter value to insert: ");
scanf("%d", &value);
insert(value);
break;
case 2:
delete();
break;
case 3:
display();
break;
case 4:
printf("Exiting program.\n");
return 0;
default:
printf("Invalid choice! Please try again.\n");
}
}
return 0;
}
```

OUTPUT:-

```
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 15
15 inserted into the queue.

Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 20
20 inserted into the queue.

Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter value to insert: 35
35 inserted into the queue.
```

```
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 15

Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 20 35
```