```c
int main(){
    int ch;
    while(1){
        printf("\nselect operation on queue to perform\n");
        printf("1.Insertion\t2.Deletion\t3.Display\t4.Exit\n");
        scanf("%d",&ch);
        switch(ch){
            case 1:
                printf("\nEnter an element to be inserted:");
                int x;
                scanf("%d",&x);
                enqueue(x);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                exit(0);
            default:
                printf("Invalid choice\n");
                break;
        }
    }
    return 0;
}
```

```c
#include<stdio.h>
#include<stdlib.h>
#define N 5
int queue[N];
int front=-1,rear=-1;
void enqueue(int x){
    if( rear==-1 && front==-1){
        front=rear=0;
        queue[rear]=x;
        printf("%d is inserted into queue\n",x);
    }
    else if(rear==N-1){
        printf("Queue overflow\n");
    }
    else{
        queue[++rear]=x;
        printf("%d is inserted into queue\n",x);
    }
}
void dequeue(){
    if (front==-1 | front>rear){
        printf("Queue underflow\n");
    }
    else{
        printf("%d is deleted from queue\n",queue[front++]);
    }
}
void display(){
    if (front==-1 | front>rear){
        printf("Queue underflow\n");
    }
    else{
        printf("Elements in the queue are\n");
        for(int i=front;i<=rear;i++){
            printf("%d\t",queue[i]);
        }
        printf("\n");
    }
```

```
select operation on queue to perform
1.Insertion     2.Deletion      3.Display       4.Exit
1

Enter an element to be inserted:10
10 is inserted into queue

select operation on queue to perform
1.Insertion     2.Deletion      3.Display       4.Exit
2
10 is deleted from queue

select operation on queue to perform
1.Insertion     2.Deletion      3.Display       4.Exit
2
Queue underflow

select operation on queue to perform
1.Insertion     2.Deletion      3.Display       4.Exit
1

Enter an element to be inserted:106
106 is inserted into queue

select operation on queue to perform
1.Insertion     2.Deletion      3.Display       4.Exit
1

Enter an element to be inserted:978
978 is inserted into queue

select operation on queue to perform
1.Insertion     2.Deletion      3.Display       4.Exit
1

Enter an element to be inserted:32
32 is inserted into queue

select operation on queue to perform
1.Insertion     2.Deletion      3.Display       4.Exit
3
Elements in the queue are
106     978     32

select operation on queue to perform
1.Insertion     2.Deletion      3.Display       4.Exit
4

Process returned 0 (0x0)   execution time : 29.574 s
Press any key to continue.
```

(8) Write a program to stimulate 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

Algorithm:-

Step 1 : start

Step 2 :- Define an array queue with fixed size MAX

Step 3 :- Set two variables FRONT = -1 & REAR = -1 to initialize an Empty array queue

Step 4 :- ENQUEUE

→ if ~~Rear~~ = MAX-1 , print Overflow

else if (FRONT == -1 && Rear == -1) . set Front = Rear = 0.

insert into queue , queue[Rear] = x;

else:

queue[++rear] = x;

Step 5 :- dequeue

→ if (~~Front > Rear~~ or Front == -1) , print underflow

→ else ~~Queue[Front]~~

item = Queue[front]

front++ ;

Step 5 :- display

→ if (front == -1 && rear == -1) print underflow

else

Iterate from i=front to Rear using for loop and print Each Element

NG

Code :-

```
#include <stdio.h>
#include <stdlib.h>
#define N 5
int queue[N];
int front = -1, rear = -1;

void enqueue(int x) {
    if (rear == -1 && front == -1) {
        front = rear = 0;
        queue[rear] = x;
        printf("%d is inserted into queue\n", x);
    } else if (rear == N-1) { printf("overflow\n"); }
    else {
        queue[++rear] = x;
        printf("%d is deleted from queue\n", queue[front++]);
    }
}

void dequeue() {
    if (front == -1 || front > rear) {
        printf("Queue underflow\n");
    } else {
        printf("%d is deleted from queue\n", queue[front++]);
    }
}

void display() {
    if (front == -1 || front > rear) {
        printf("Queue underflow\n");
    } else {
        printf("Elements in the queue are\n");
        for (int i = front; i <= rear; i++) {
            printf("%d\t", queue[i]);
        }
        printf("\n");
    }
}
```

```
int main(){
    int ch;
    while(1){
    printf("\n Select operation on queue to perform\n");
    printf("1.Insertion\t2.Deletion\t3.Display\t4.exit\n");
    Scanf("%d", &ch);
    switch(ch){
        case 1:
            printf("\n Enter an element to be inserted: ");
            int x;
            Scanf("%d", &x);
            Enqueue(x);
            break;
        case 2:
            dequeue();
            break;
        case 3:
            display();
            break;
        case 4:
            exit(0);
        default:
            printf("Invalid choice\n");
            break;
    }
    }
    return 0;
}
```

o/p:-

Select operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
1
Enter an element to be inserted: 978
978 is inserted into queue.

Select operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
2
978 is deleted from queue

Select operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
2
Queue underflow

Select operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
1
Enter an element to be inserted: 21
21 is inserted into queue

Select operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
1
Enter an element to be inserted: 324
324 is inserted into queue

Select operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
1
Enter an element to be inserted: 106
106 is inserted into queue

Select operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
3

Elements in the queue are
21 324 106

MG
13/10/25.