

### **Program 3a**

Write A Program 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

Code:

```
#include <stdio.h>
#define max_size 4
int queue [max_size];
int front =-1;
int rear=-1;

void insert(int value){
    if (rear==max_size -1){
        printf("Queue overflow! Cannot insert elements");
    }
    else{
        if(front == -1){
            front =0;
        }

        queue[++rear]=value;
        printf("Insert %d into queue",value);
    }
}

void delete(){
    if(front== -1 || front>rear){
        printf("Queue underflow!Cannot delete ");
    }
    else{
        printf("Deleted %d from the queue",queue[front]);
        front++;
    }
}

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

```

}

int main(){
    int choice, value;
    while(1){
        printf("\n1.Insert");
        printf("\n2.Delete");
        printf("\n3.Display");
        printf("\n4.Exit");
        printf("\nEnter your choice:");
        scanf("%d",&choice);
        switch (choice){
            case 1: printf("Enter a value to insert:");
                    scanf("%d",&value);
                    insert(value);
                    break;
            case 2: delete();
                    break;
            case 3: display();
                    break;
            case 4: return 0;
            default: printf("Invalid choice! Please try again\n");
        }
    }
}

```

```

Enter your choice:1
Enter a value to insert:4
Insert 4 into queue
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:1
Enter a value to insert:5
Insert 5 into queue
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:1
Enter a value to insert:9
Insert 9 into queue
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:3
Queue Elements
4 5 9

1.Insert
2.Delete
3.Display
4.Exit
Enter your choice:4

```

14/10/24

Q3. WAP to simulate the working of a queue of integers using an array.

```

int queue[100];
int front, rear = -1;
int isfull()
{
    return rear == 99;
}

int isempty()
{
    return front == -1;
}

void enqueue(int value)
{
    if (isfull())
        printf("Queue overflow");
    else if (front == -1)
        front = 0;
    rear++;
    queue[rear] = value;
    printf("%d inserted to queue", value);
}

void dequeue()
{
    if (isempty())
        printf("Queue is empty");
    else if (front > rear)
        front = rear = -1;
    else
        front++;
}

void display()
{
    if (isempty())
        printf("Queue is empty");
    else
        printf("Queue elements: ");
    for (int i = front; i <= rear; i++)
        printf("%d ", queue[i]);
}

int main()
{
    int choice, value;
    do
    {
        printf("Queue operations:");
        printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n");
        choice = 1;
        value = 3;
        enqueue(value);
        choice = 2;
        dequeue();
        choice = 3;
        display();
        choice = 4;
        exit(0);
    } while (choice != 4);
}

```

14/10/24

Q3. WAP to simulate the working of a queue of integers using an array.

```

int queue[100];
int front, rear = -1;
int isfull()
{
    return rear == 99;
}

int isempty()
{
    return front == -1;
}

void enqueue(int value)
{
    if (isfull())
        printf("Queue overflow");
    else if (front == -1)
        front = 0;
    rear++;
    queue[rear] = value;
    printf("%d inserted to queue", value);
}

void dequeue()
{
    if (isempty())
        printf("Queue is empty");
    else if (front > rear)
        front = rear = -1;
    else
        front++;
}

void display()
{
    if (isempty())
        printf("Queue is empty");
    else
        printf("Queue elements: ");
    for (int i = front; i <= rear; i++)
        printf("%d ", queue[i]);
}

int main()
{
    int choice, value;
    do
    {
        printf("Queue operations:");
        printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n");
        choice = 1;
        value = 3;
        enqueue(value);
        choice = 2;
        dequeue();
        choice = 3;
        display();
        choice = 4;
        exit(0);
    } while (choice != 4);
}

```

Output

Queue operations:

1. Insert

2. Delete

3. Display

4. Exit

display Enter your choice : 3

Queue is empty!

insertion Enter your choice : 1

Enter the value to insert : 3

3 inserted into the queue

deletion Enter your choice : 2

3 deleted from queue

Enter your choice : 3

Queue elements are : 4 5 9

exit Enter your choice : 4

Exiting

overflow Enter value to insert : 9

Queue overflow! cannot insert 9

Verdict: 11.