

- 1) Write a program to simulate the working of the queue of integers using an array. provide the following operations, Insert, delete, display the program should print appropriate message for overflow and underflow condition.

```
#include <stdio.h>
```

```
#define SIZE 10
```

```
void enqueue();
```

```
void dequeue();
```

```
void show();
```

```
int arr[SIZE];
```

```
int rear = -1;
```

```
int front = -1;
```

```
main()
```

```
{
```

```
    int ch;
```

```
    while (1)
```

```
    {
```

```
        printf("\n 1. Enqueue operation\n 2. dequeue operation\n 3. display the queue\n 4. exit");
```

```
        printf("Enter your choice of operations:");
```

```
        scanf("%d", &ch);
```

```
        switch (ch)
```

```
        {
```

```
            case 1:
```

```
                enqueue();
```

```
                break;
```

```
            case 2:
```

```
                dequeue();
```

```
                break;
```

```

case 3 :
show();
break;
case 4 :
exit (0);
default :
printf (" Incorrect choice \n");

```

void enqueue ()

```

{
    int insert_item;
    if (rear == size - 1)
        printf ("overflow \n");
    else
        if (front == -1)
            front = 0;
        printf (" Element to be inserted in the Queue is: ");
        scanf ("%d", &insert_item);
        rear = rear + 1;
        inp_arr[rear] = insert_item;
}

```

void dequeue ()

```

{
    if (front == -1 || front > rear)
        printf (" underflow \n");
    return;
}

```



else

```
{  
    printf("Element deleted from the queue : %d\n",  
           inp_arr[front]);  
    front = front + 1;  
}
```

void show()

```
{  
    if (front == -1)  
        printf("Empty Queue\n");  
    else  
    {  
        printf("Queue : \n");  
        for (int i = front; i <= rear; i++)  
            printf("%d ", inp_arr[i]);  
        printf("\n");  
    }  
}
```

Output :

1. Enqueue operation
2. Dequeue operation
3. display.
4. Exit

Enter your choice of operation : 1

Element to be inserted in the queue : 10

Enter your choice of the operation : 1  
Enter the element to be ~~add~~ inserted in the queue : 15

Enter your choice of the operation : 1  
Element to be inserted in the queue : 20

Enter your choice of the operation : 1  
Element to be inserted in the queue : 25

Enter your choice of the operation : 1  
overflow

Enter your choice of the operation : 3  
queue : 10 15 20 25

Enter your choice of the operation : 2.  
Element deleted from the queue : 10.  
Element deleted from the queue : 15  
Element deleted from the queue : 20  
Element deleted from the queue : 25

Enter your choice of the operation : 2.  
underflow

Enter your choice of the operation : 3  
queue is empty



2. Write a program to simulate the working of a circular queue using an array. Provide the following operations insert, delete & display. The program should print appropriate message for queue empty and queue overflow condition.

```
#include <stdio.h>
```

```
#define max 6
```

```
int queue [max];
```

```
int front = -1;
```

```
int rear = -1;
```

```
void enqueue (int element)
```

```
{ if (front == -1 && rear == -1)
```

```
{
```

```
    front = 0;
```

```
    rear = 0;
```

```
    queue [rear] = element;
```

```
}
```

```
else if (rear + 1 && max == front)
```

```
{
```

```
    printf ("Queue is overflow..");
```

```
}
```

```
else
```

```
{
```

```
    rear = (rear + 1) % max;
```

```
    queue [rear] = element;
```

```
}
```

```
}
```

```

int dequeue ()
{
    if ((front == -1) && (rear == -1))
    {
        printf("\nQueue is underflow ..");
    }
    else if (front == -rear)
    {
        printf("\n the dequeued Element is %d", queue[front]);
        front = -1;
        rear = -1;
    }
    else
    {
        printf("\n the dequeued element is %d", queue[front]);
        front = (front + 1) % MAX;
    }
}

```

```

void display ()
{
    int i = front;
    if (front == -1 && rear == -1)
    {
        printf("\n Queue is empty ..");
    }
    else
    {
        printf("\n Element in a queue are :");
        while (i <= rear)
    }
}

```



4

```
printf("\n Elements in a queue are : ");
while (i <= rear)
```

```
{
```

```
printf("%d ", queue[i]);
i = i + 1;
}
```

```
}
```

```
}
```

```
int main ()
```

```
{
```

```
int choice = 1, x;
```

```
while (choice < 4 && choice != 0)
```

```
{
```

```
printf("\n 1. insert an element : \n 2. delete an element \n 3. display the element \n 4. exit ");
```

```
scanf("%d", &choice);
```

```
switch (choice)
```

```
{
```

```
case 1:
```

```
printf("Enter the element which is to be inserted:");
```

```
scanf("%d", &x);
```

```
enqueue(x);
```

```
break;
```

```
case 2:
```

```
dequeue();
```

```
break;
```

```
case 3:
```

```
display();
```

```
}
```

```
return 0;
```

Output :

Press 1: Insert an Element

Press 2: delete an Element

Press 3: display the element

Enter your choice: 1

Enter the element which is to inserted : 1

Enter your choice: 1

Enter the element which is to inserted : 2

Enter your choice: 1

Enter the element which is to inserted : 3

Enter your choice: 1

Enter the element which is to inserted : 4

Enter your choice: 1

Queue is overflow...

Enter your choice: 3

Queue: 1, 2, 3, 4

Enter your choice: 2

the dequeued element is 1

the dequeued element is 2

the dequeued element is 3

the dequeued element is 4

Enter your choice: 2

Queue is underflow...

Enter your choice: 3

Queue is empty..