

-11

- 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>
```

```
#include <string.h>
```

```
#define N = 5
```

```
int front = -1, rear = -1;
```

```
int queue[N]
```

```
void enqueue(int x)
```

```
{
```

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

```
    {
```

```
        printf("Underflow");
```

```
        front = rear = 0;
```

```
        queue[rear] = x;
```

```
        rear++;
```

```
    }
```

```
    else if (rear == N-1)
```

```
    {
```

```
        printf("Overflow");
```

```
    }
```

```
    else {
```

```
        queue[rear] = x;
```

```
        rear++;
```

```
    }
```

```
void dequeue {
```

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

```
    {
        printf("Underflow");
    }
```

```
    else if (front == rear)
```

```
    {
        printf(
            front = rear = -1;
            printf("Element Deleted");
        );
    }
```

```
    else {
```

```
        front = front + 1;
        printf("Element Deleted");
    }
```

```
}
```

```
void display {
```

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

```
    {
        printf("Queue is empty");
    }
```

```
    else {
```

```
        for (int i = front, i <= rear, i++)
```

```
        {
            printf("%d", queue[i]);
```

```
            printf(" ");
            i++;
        }
```

```
}
```

```
}
```

```
void main()
```

```
{
```

```
    char str[10], sym;
```

```
    int sym;
```



```

printf("Enter 1. for insertion, 2 for
deletion 3. for display, 4. exit)
while(1)
{
    printf("%s", charr);
    switch (ch)
    {

```

```

        case 1: printf("Enter the element string character");
                 printf scanf("%d", &sym);
                 enqueue(sym);
        case 2: printf("Deletion");
                 dequeue();
        case 3: printf("Display");
                 display();
        case 4: exit(0);
    }
}

```

Output

Enter 1. for insertion, 2 for deletion
3. for display, 4. exit.

1. Enter the element
1

1	1	1	1
---	---	---	---

~~2. Deletion~~

1. Enter the element

2

1	2		
---	---	--	--

2. Deletion

Element Deleted

3. Display

2

1	2		
---	---	--	--

~~4. exit~~

~~exit~~

= 1/exit

Write a program to simulate working of a circular queue using an array. Provide the following operations: insert, delete & display. The program should print appropriate conditions.

```
#include <stdio.h>
#include <string.h>
#define n 10
int queue[n];
```

```
void enqueue(int x)
{
    if (front == -1 && rear == -1)
    {
        front = rear = 0;
        queue[rear] = x;
    }
    else if ((rear + 1) % N == front)
    {
        printf("queue is full");
    }
    else
    {
        rear = (rear + 1) % N;
        queue[rear] = x;
    }
}
```



```
void dequeue
```

```
{
```

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

```
{
```

```
printf("Queue is empty");
```

```
}
```

```
else if (front == rear)
```

```
{
```

```
front = rear = -1;
```

```
}
```

```
else
```

```
{
```

```
printf("%d", queue[front]);
```

```
front = (front + 1) % N;
```

```
}
```

```
}
```

```
void display
```

```
{
```

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

```
{
```

```
printf("Queue is empty");
```

```
}
```

```
else
```

```
{ printf("Queue is: ");
```

```
while (i != rear)
```

```
{
```

```
printf("%d", queue[i]);
```

```
i = (i + 1) % N;
```

```
}
```

```
}
```

Output

Enter 1. insert 2. delete 3. display 4. exit

Enter an element 20

20	
----	--

Enter 1. insert 2. delete 3. display 4. exit

Enter an element 30

20	30
----	----

~~Enter 1. insert 2. delete 3. display 4. exit~~

Enter an element 40

20	30	40
----	----	----

Enter 1. insert 2. delete 3. display 4. exit

20	30	40
----	----	----

Enter 1. insert 2. delete 3. display 4. exit

Enter ~~an element~~ 1. insert 2. delete 3. display 4. exit

		40
--	--	----

Enter an element 30

30		40
----	--	----

~~Enter 1. insert 2. delete 3. display 4. exit~~

~~Display~~

~~30~~

40

30

08/01/24