

```
① #include <stdio.h>
```

```
int que[N];
```

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

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

```
void enqueue(int n) {
```

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

```
        front = 0;
```

```
        rear = 0;
```

```
        que[front] = n;
```

```
}
```

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

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

```
}
```

```
else {
```

```
    que[rear] = n;
```

```
    rear++;
```

```
}
```

```
void dequeue() {
```

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

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

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

~~```
 printf("the popped element is %d"; que[front]);
```~~

```
 front = -1;
```

```
 rear = -1;
```

```
}
```

```
else {
```

~~```
    printf("the popped element is : %d"; que[front]);
```~~

```
    front++;
```

```
}
```

```
void display() {
    for(int i=0; i<=max; i++) {
        printf("%d ", que[front]);
    }
}

int main() {
    int P;
    while(1) {
        char choice;
        printf("enter the choice : a. enqueue\n"
               "b. dequeue c. display d. exit : \n");
        scanf("%c", &choice);
        switch(choice) {
            case('a'):
                printf("enter the number to be\n"
                       "pushed : ");
                scanf("%d", &P);
                enqueue(P);
                break;
            case('b'):
                dequeue();
                break;
            case('c'):
                display();
                break;
            case('d'):
                break;
                return 0;
            default:
                printf("Invalid choice");
                break;
        }
    }
}
```

return 0;

Output:-

enter the choice : a.enqueue b.dequeue c.display
d.exit.

a.

enter the number to be pushed :

1

enter the choice : a.enqueue b.dequeue c.display
d.exit.

a.

enter the number to be pushed :

2.

enter the choice : a.enqueue b.dequeue c.display
d.exit.

a.

enter the number to be pushed

3.

enter the choice : a.enqueue b.dequeue . c.display
d.exit.

b

~~enter~~

the popped element : 1

enter the choice : a.enqueue b.dequeue c.display
d.exit.

b.

the popped element : 2.

enter the choice : a.enqueue b.dequeue c.display
d.exit.

c

~~the~~ 3 4 5.

② #include < stdio.h >

#define N 5

int queue[N];

int front = -1;

int rear = -1;

void enqueue (int x) {

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

front = 0;

rear = 0;

queue[front] = x;

}

else if (front == (rear + 1) % N) {

printf ("Queue is full\n");

}

else {

rear = (rear + 1) % N;

queue[rear] = x;

3. ~~3~~

void dequeue () {

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

printf ("Queue is empty\n");

}

else if (front == rear) {

printf ("Element is : %d\n" queue[front]);

front = -1;

rear = -1;

3

else {

```
    printf("the popped element is : %d\n",
           que[front]);
    front = (front + 1) % N;
```

}

void display() {

```
    int i = front;
```

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

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

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

}

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

}

int main() {

```
    char choice;
```

```
    printf("enter the choice : a. enqueue b.
```

```
          dequeue c. display d. exit : \n");
```

~~```
scanf("%c", &choice);
```~~~~```
switch(choice) {
```~~~~```
 case('a') {
```~~

```
 printf("enter the number : \n");
```

~~```
        scanf("%d", &p);
```~~~~```
 enque(p);
```~~~~```
        break;
```~~~~```
 case('b') {
```~~~~```
        dequeue();
```~~~~```
 break;
```~~

case ('c') :

    display();  
    break;

case ('d')

    break;  
    return 0;

default:

    printf("invalid choice\n");  
    break;

}

return 0;

W  
u III  
0142

Output :-

enter the choice a.enqueue b.dequeue c.display  
d.exit.

a.

enter the number to be pushed

1.

enter the choice a.enqueue b.dequeue c.display  
d.exit

b.

enter the number to be pushed

2.

enter the choice a.enqueue b.dequeue c.display  
d.exit

c.

enter the number to be pushed

3.

enter the choice a.enqueue b.dequeue c.display  
d.exit.