```
/*************************************ASSIGNMENT 7*************************************
Pizza parlor accepting maximum M orders. Orders are served on a first come first served basis.
Queues are frequently used in computer programming, and a typical example is the creation of a
job queue by an operating system. If the operating system does not use priorities, then the jobs
are processed in the order they enter the system. Write a program for simulating job queue. Write
functions to add jobs and delete jobs from the queue.
*****************************************************************************************/
#include <stdio.h>
#include <string.h>


#define MAX 5
#define NAME_LEN 30


char queue[MAX][NAME_LEN];
int front = -1, rear = -1;


// Add order (enqueue)
void enqueue(char order[]) {
if (rear == MAX - 1) {
printf("Queue is full. Cannot accept more orders.\n");
return;
}
if (front == -1) front = 0;
rear++;
strcpy(queue[rear], order);
printf("Order '%s' added to the queue.\n", order);
}


// Serve order (dequeue)
void dequeue() {
if (front == -1 || front > rear) {
printf("Queue is empty. No orders to serve.\n");
return;
}
printf("Order '%s' has been served.\n", queue[front]);
front++;
}


// Display current orders
void display() {
if (front == -1 || front > rear) {
printf("Queue is empty.\n");
return;
}
printf("Current orders in queue:\n");
for (int i = front; i <= rear; i++) {
```

```
printf("%d. %s\n", i - front + 1, queue[i]);
}
}


// Main function
int main() {
int choice;
char order[NAME_LEN];


do {
printf("\n--- Pizza Parlor Queue ---\n");
printf("1. Add Order\n");
printf("2. Serve Order\n");
printf("3. View Orders\n");
printf("4. Exit\n");
printf("Enter choice: ");
scanf("%d", &choice);
getchar(); // consume newline


switch (choice) {
case 1:
printf("Enter order name: ");
fgets(order, NAME_LEN, stdin);
order[strcspn(order, "\n")] = '\0'; // remove newline
enqueue(order);
break;
case 2:
dequeue();
break;
case 3:
display();
break;
case 4:
printf("Thank you! Exiting...\n");
break;
default:
printf("Invalid choice. Try again.\n");
}
} while (choice != 4);


return 0;
}
/*********************************************OUTPUT*****************************************
Enter choice: 1
Enter order name: Margherita
Order 'Margherita' added to the queue.


--- Pizza Parlor Queue ---
```

1. Add Order
2. Serve Order
3. View Orders
4. Exit
Enter choice: 1
Enter order name: Pepperoni
Order 'Pepperoni' added to the queue.

--- Pizza Parlor Queue ---
1. Add Order
2. Serve Order
3. View Orders
4. Exit
Enter choice: 1
Enter order name: Cheese Burst
Order 'Cheese Burst' added to the queue.

--- Pizza Parlor Queue ---
1. Add Order
2. Serve Order
3. View Orders
4. Exit
Enter choice: 3
Current orders in queue:
1. Margherita
2. Pepperoni
3. Cheese Burst

--- Pizza Parlor Queue ---
1. Add Order
2. Serve Order
3. View Orders
4. Exit
Enter choice: 2
Order 'Margherita' has been served.

--- Pizza Parlor Queue ---
1. Add Order
2. Serve Order
3. View Orders
4. Exit
Enter choice: 3
Current orders in queue:
1. Pepperoni
2. Cheese Burst

--- Pizza Parlor Queue ---

1. Add Order

2. Serve Order
3. View Orders
4. Exit
Enter choice: 2
Order 'Pepperoni' has been served.

--- Pizza Parlor Queue ---
1. Add Order
2. Serve Order
3. View Orders
4. Exit
Enter choice: 2
Order 'Cheese Burst' has been served.

--- Pizza Parlor Queue ---
1. Add Order
2. Serve Order
3. View Orders
4. Exit
Enter choice: 2
Queue is empty. No orders to serve.

--- Pizza Parlor Queue ---
1. Add Order
2. Serve Order
3. View Orders
4. Exit
Enter choice: 4
Thank you! Exiting...
**********/