Cafeteria service

scenario

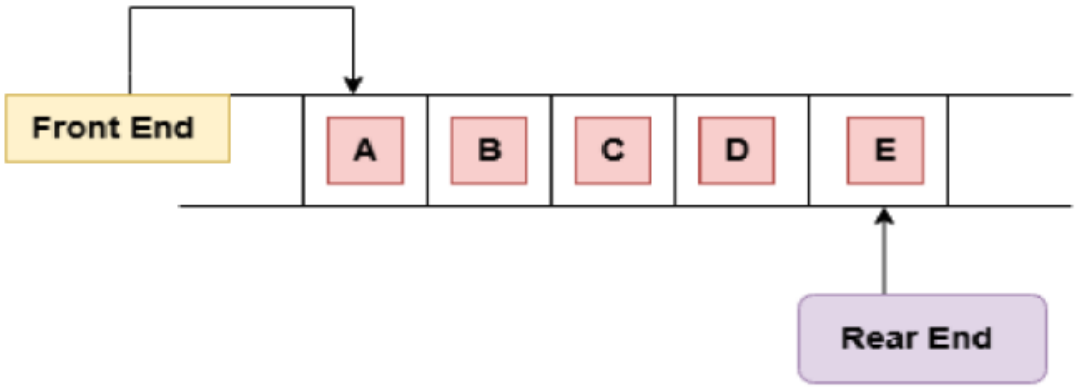
We implemented a Cafeteria Service using a queue data structure.

-There is a line of people waiting to be served, with a maximum of 10 people.

-Students can join the line as long as it has fewer than 10 people.

-If the queue is full, adding an 11th person is not allowed, and a message appears:
"Queue is full! Cannot add [name]".

-This ensures first-come-first-serve (FIFO) service, meaning the first student to arrive is served first.



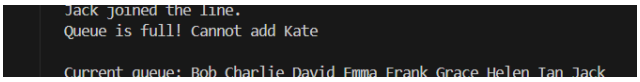
Representation of queue in data structure

- Cases :
  - If it have no value or people the output will show empty because the condition in printQueue

```
if (isEmpty()) {
        cout << "[empty]\n";
        return;
```

}

- When it full the output will show queue is full



Design Choice

Implementation Type: Array-based queue with compaction

-Array size: MAX = 10

-Pointers:

- front → index of first student in line
- rear → index of next available slot

-Compaction Trigger: When the array is full (rear == MAX) and front >= MAX * 0.5

-Reason for choice:

- Array is simple and easy to visualize for small queues.
- Compaction ensures that even if some people have left, we can still add new students without losing FIFO order.

Methods

Enqueue

- Adds name to the queue at rear

-If full and compaction threshold not met, prints "Queue is full!"

Dequeue

- Removes the student at front and serves them.

- If empty, prints "Queue is empty!".

IsEmpty

- Returns true if the queue is empty

IsFull

- Returns true if the array is full

Compact

- Shifts remaining students to index 0, resets front and rear

- Ensures space for new students

PrintQueue

- Prints all students from front to rear – 1

- Prints [empty] if queue is empty

Edge cases handled

Empty dequeue: Prevents errors, prints warning.
Full enqueue: Prevents adding 11th person, prints warning.
Compaction: Moves remaining students to front when the array is mostly full.
Memory safety: No dynamic memory is used; array handles up to 10 students safely.

Time Complexity

- Worst case : The queue is full, and compact() is triggered.
- Best case : The queue is not full just assign one element.

- Evidence (code output)