

Ramirez Manriquez Luis Fernando

Capturas

The screenshot shows a code editor interface with a dark theme. The top bar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and Language (set to C). The main window displays a file named "main.cpp" containing C++ code for a circular queue. The code includes functions for initializing, enqueueing, dequeuing, and printing the queue. The output window below shows the execution results.

```
33 void printQueue(ColaCircular *c) {
34     printf("Cola: ");
35     if (isEmpty(c)) {
36         printf("(vacia)\n");
37         return;
38     }
39     int i = c->frente;
40     for (int k = 0; k < c->size; k++) {
41         printf("%d ", c->datos[i]);
42         i = (i + 1) % MAX;
43     }
44     printf("\nFrente: %d | Final: %d | Size: %d\n", c->frente, c->final, c
45 }
46
47 int main() {
48     ColaCircular c;
49     init(&c);
50
51     enqueue(&c, 5);
52     enqueue(&c, 10);
53     enqueue(&c, 15);
54     printQueue(&c);
55
56     dequeue(&c);
57     printQueue(&c);
58
59     enqueue(&c, 20);
60     enqueue(&c, 25);
61     enqueue(&c, 30);
62     printQueue(&c);
63
64     return 0;
65 }
66
```

input

```
Cola circular inicializada
Cola: 5 10 15
Frente: 0 | Final: 3 | Size: 3
Cola: 10 15
Frente: 1 | Final: 3 | Size: 2
Cola: 10 15 20 25 30
Frente: 1 | Final: 1 | Size: 5

...Program finished with exit code 0
Press ENTER to exit console.
```

The screenshot shows a C++ development environment with the following details:

- Toolbar:** Run, Debug, Stop, Share, Save, Beautify.
- Code Editor:** File named `main.cpp`. The code implements a `Deque` class with methods for insertion at front and rear, deletion from front and rear, and printing the deque's state. It also includes a `main` function demonstrating the class's usage.
- Output Window:** Labeled "input". It displays the program's output, which shows the initial deque state, the result of inserting elements, and the state after deleting elements. The output is:

```
Deque: 30 10 15 20 40
Front: 3 | Rear: 2 | Size: 5
Deque: 10 15 20
Front: 4 | Rear: 1 | Size: 3
```

...Program finished with exit code 0
Press ENTER to exit console.

Actividad

Preguntas (A)

1. ¿Por qué en la cola circular no se desperdician posiciones como en la cola lineal?

Porque la cola circular “da la vuelta” cuando llega al final. En vez de dejar huecos inutilizables, vuelve al inicio y sigue usando esas posiciones, así que el espacio se aprovecha al máximo.

2. ¿Qué pasa cuando el índice final llega al final del arreglo?

Literalmente solo “se reinicia”: si llega al último espacio, salta al índice 0. No se sale del arreglo, solo gira como un círculo.

3. ¿Qué sucede si intentas encolar cuando la cola está llena?

Pues simplemente no te deja. La estructura detecta que ya no cabe nada más y te marca que está llena, así que no mete el nuevo dato.

Preguntas (B)

1. ¿Qué diferencia tiene el deque frente a la cola circular normal?

El deque es más flexible porque puedes meter y sacar datos tanto por el frente como por atrás. La cola circular solo deja meter por atrás y sacar por el frente.

2. ¿Qué operaciones permiten simular tanto FIFO como LIFO?

Si usas el frente y el final estratégicamente, puedes comportarte como una cola normal (FIFO) o como una pila (LIFO). Todo depende de por dónde metas y saques los valores.

3. ¿Qué aplicaciones prácticas tiene deque?

Sirve mucho en cosas donde necesitas mover datos por los dos lados, como navegadores (adelante/atrás), simulaciones, buffers de tareas o algoritmos donde necesitas manejar elementos recientes y viejos rápidamente.

