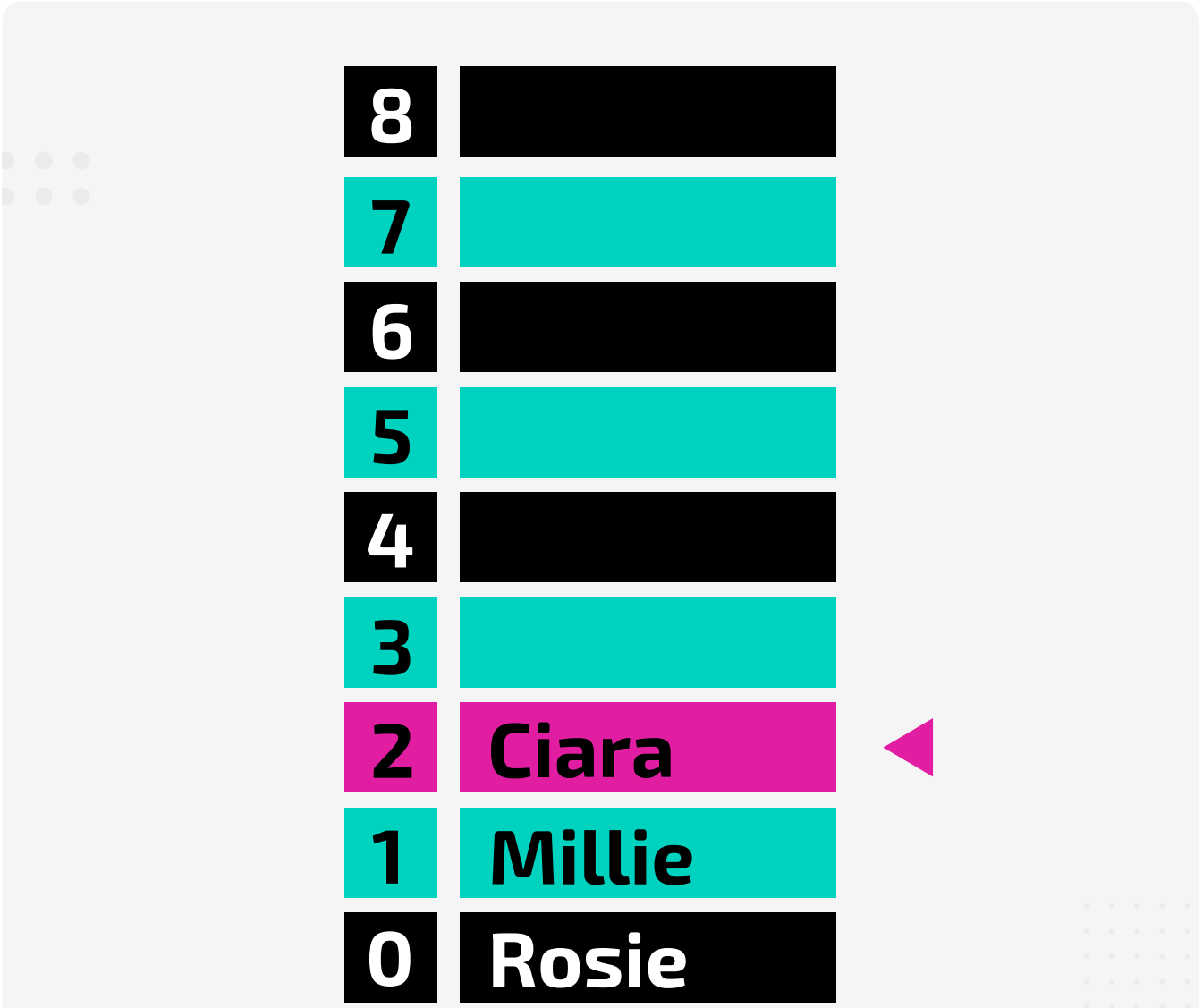
Stack: operations 1

A Level Advanced

P P P P

The following diagram is an abstraction of a stack. The stack currently contains three items (names). The pointer to the top of the stack is shown in pink.



A stack

The following operations are carried out in the order given:

- push Harry
- pop
- push Luna
- peek
- pop
- pop

Part A

To which **position** will the `top` pointer point after all the operations have been carried out?

Part B

When all of the operations have been carried out, which **name** will be at the top of the stack?

Stack: operations 3

A Level Advanced
P P P P

A stack has been used to store a sequence of numbers. The stack is as shown in the table below (**state 1**). The number 3 is at the top of the stack.

3
71
19

At the end of a sequence of operations, the stack looks like this (**state 2**):

1
19

Select the correct set of operations to transform the stack from **state 1** to **state 2**.

- peek, push 59, pop, push 91, pop, pop, push 1
- peek, pop, pop, pop, push 1, push 59, pop, push 19, peek
- pop, push 59, pop, push 91, pop, peek, pop, push 1
- push 1, pop, push 59, pop, push 91, pop, peek, pop, push 17

Stack: operations 2

A Level Advanced
P P P P

A stack is used to store the changes made to some text in a simple word processor. Every time a change is made, the details are pushed onto the stack. Each time the 'Undo' button is pressed, the last operation is popped from the stack and the operation is undone.

The state of the stack after a few minutes of editing is shown in the table below. The operation 'style — Arial' is currently at the top of the stack.

style — Arial
emphasis — italic
align — right
align — left
emphasis — normal
style — Georgia
size — 12
colour — black

The following operations are then carried out by the user in the order shown:

1. change colour to blue
2. undo
3. undo
4. change size to 14

How will the text be formatted at the end of the sequence of operations?

- colour — black
  style — Georgia
  emphasis — italic
  align — right
  size — 14

- colour — blue
  style — Arial
  emphasis — italic
  align — right
  size — 14

- colour — blue
  style — Georgia
  emphasis — normal
  align — right
  size — 14

- colour — black
  style — Arial

emphasis — italic
align — right
size — 12

Stack: pop algorithm

A Level Advanced
P P P P

The following statements, in structured English, form the algorithm for the **pop** operation on a stack. Put them into the correct order. You do not need to indent any of the statements.

Available items

- Check if the stack is empty
- Decrement stack_pointer by one
- If stack empty, display error and return / halt
- Else return the data item indexed by stack_pointer

Linear queue: operations

A Level Advanced

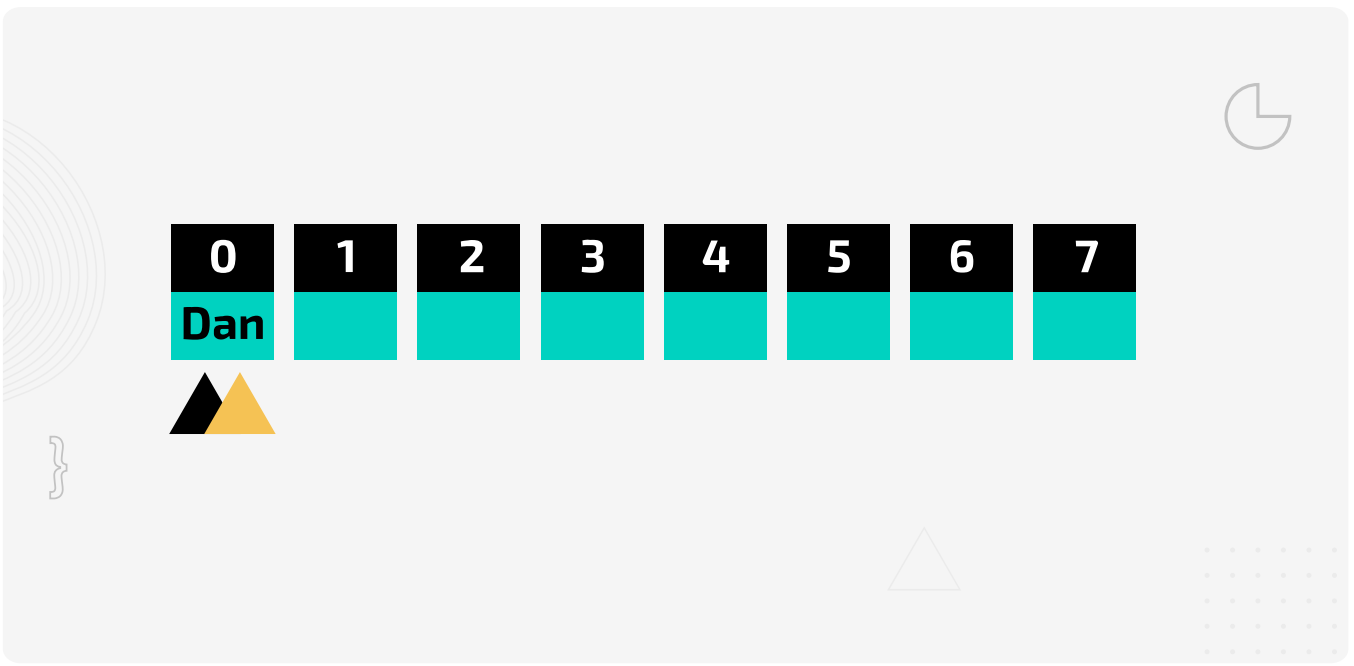
P P P P

The following diagram is an abstraction of a **linear queue**.

The `front` pointer (black) points to the start of the queue. The `rear` pointer (yellow) points to the end of the queue.

Dan is at the front of the queue in **position 0**.
The following names are **enqueued** in the order given:

- Adam
- Sasha
- Mohammed
- Jay



An abstraction of a linear queue

Part A Front pointer

To which position will `front` point after all the names have been enqueued?

Part B Rear pointer

To which position will `rear` point after all the names have been enqueued?

Part C Where is Mohammed?

Which position will `Mohammed` occupy after all the names have been enqueued?

Linear queue: enqueue algorithm

A Level C C Advanced C C

Niamh is planning to use a queue as a data structure in her software project. She is designing the algorithm to enqueue data onto a linear queue based on an array. Can you help her by dragging the structured English statements (shown below) into the correct order?

In the statements, **rear** is a pointer to the end of the queue and **maxsize** is the maximum size of the queue. **You must use appropriate indentation**.

Available items

```
increment rear by 1
```

```
if rear + 1 is equal to maxsize then
```

```
else
```

```
display message "Queue is full"
```

```
end if
```

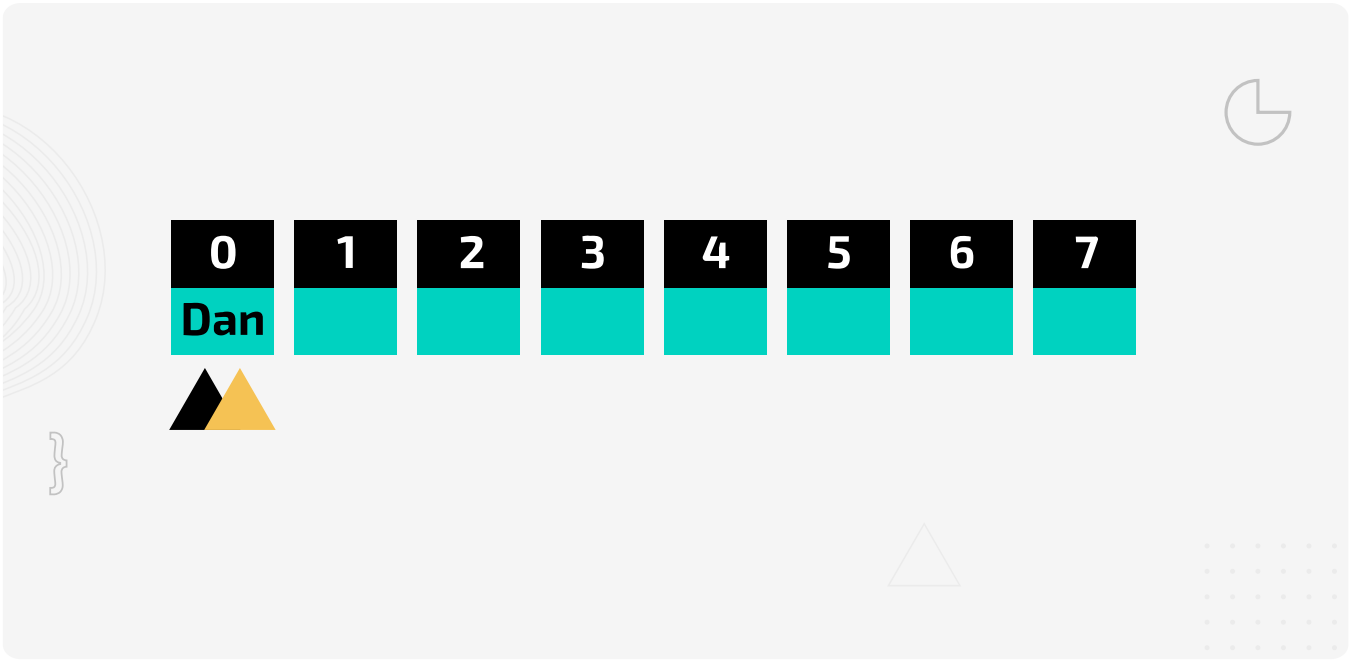
```
insert data at queue[rear]
```

Circular queue: operations

A Level Advanced

C C C C

The following diagram is an abstraction of a queue. The queue will be implemented using a static array and **circular queue** methods are used to make efficient use of space.
The `front` pointer (black) points to the start of the queue. The `rear` pointer (yellow) points to the end of the queue.



An abstraction of a circular queue

Dan is at the front of the queue in **position 0**. A series of operations occur as follows:

- enqueue Nigel,
- enqueue Anil,
- enqueue Sasham,
- **dequeue**,
- enqueue Jordan,
- **dequeue**,
- enqueue Aayna,
- enqueue Germaine,
- **dequeue**,
- **dequeue**,
- enqueue Tom,
- enqueue Ben,
- **dequeue**,
- enqueue Amar

Part A Front pointer

To which position will `front` point after all the operations have been carried out?

Part B Rear pointer

To which position will `rear` point after all the operations have been carried out?

Part C Where is Germaine?

In which position will Germaine be after all the operations have been carried out?

Priority queue: operations

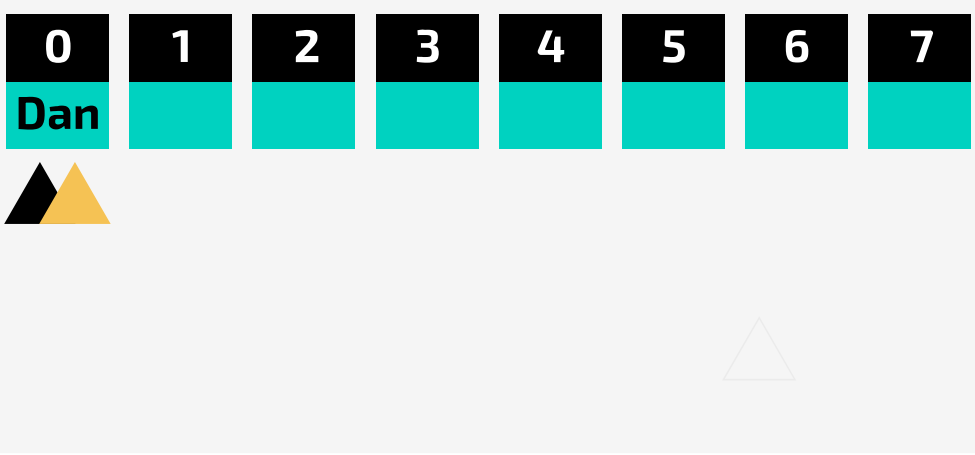
A Level Advanced

P P P P

The following diagram is an abstraction of a **priority queue**. Teachers have higher priority than students in the queuing system. Dan, a student, is at the front of the queue in position 0.

The `front` pointer (black) points to the start of the queue.

The `rear` pointer (yellow) points to the end of the queue.



A priority queue

Part A Front pointer

Adam (Teacher), Sasha (Student), and Mohammed (Teacher) are added to the queue (enqueued) in the order given.

To which position will `front` point after all the names have been added?

Part B Rear pointer

To which position will `rear` point after all the names have been added?

Part C Where is Mohammed?

In which position will Mohammed be after all the names have been added?