1. Similarities between an array and a stackgeeksforgeeks+1

- Both are linear data structures where elements are stored in a sequence.
- Both can use contiguous memory (array implementation), allowing constant-time access by index.
- Both can store multiple values of the same data type.

---

2. How a programmer might use a stack in a compileracceldata+1

- To check balanced symbols like (), {}, [] during parsing: push on opening symbol, pop on closing symbol and check for a match.
- To evaluate expressions (e.g., converting infix to postfix) by pushing operands and operators and then popping them in the correct order.
- To implement the call stack: storing return addresses, parameters, and local variables for each function call and popping them when the function returns.

---

3. Stack code: detailed output (Google Docs-friendly)

Stack rules (LIFO):

- push(v): add v to the top
- pop(): remove and return the top item
- top(): return the top item without removing itbuildingjavaprograms+1

Code and trace (use normal “Preformatted” style in Google Docs):

```
Stack s = new Stack(10); // []
s.push(5); //
s.push(8); //reddit+1
int x = s.pop(); // x = 8,
s.push(x); //reddit+1
s.push(12); //reddit+1
s.push(13); //reddit+1
int y = s.pop(); // y = 13, stackreddit+1
System.out.println(x + " " + y); // prints 8 13
y = s.pop(); // y = 12, stackacceldata+1
x = s.top(); // x = 8, stackreddit+1
System.out.println(x + " " + y); // prints 8 12
```

Explanation:

- First pop() sets x = 8.
- Later pop() sets y = 13, then y = 12.

- top() reads 8.

Final output:

- First line: 8 13
- Second line: 8 12geeksforgeeks+1

---

4. “Hot plate” problem as a stackgeeksforgeeks+1

- New plates are always placed on top of the pile (like push).
- Customers always take the plate from the top of the pile (like pop).
- This is last-in, first-out (LIFO): the last plate added (still warm) is the first one taken.

---

5. Queue code: detailed output (Google Docs-friendly)

Queue rules (FIFO):

- enqueue(v): add v at the rear
- dequeue(): remove and return the front item
- front(): return the front item without removing it

Code and trace:

```
Queue q = new Queue(10); // []
q.enqueue(5); //
q.enqueue(8); //reddit+1
int x = q.dequeue(); // x = 5, queue
q.enqueue(x); //acceldata+1
q.enqueue(12); //acceldata+1
q.enqueue(13); //reddit+1
int y = q.dequeue(); // y = 8, queue
System.out.println(x + " " + y); // prints 5 8
y = q.dequeue(); // y = 5, queue
x = q.front(); // x = 12, queue
System.out.println(x + " " + y); // prints 12 5
```

Explanation:

- First dequeue() sets x = 5 (first inserted).
- Next dequeue() sets y = 8, then y = 5.
- front() reads 12.

Final output:

- First line: 5 8
- Second line: 12 5

---

6. Difference between FIFO and LIFOyoutubegeeksforgeeks+1

(You can recreate this as a 2-column table in Google Docs.)

- FIFO (First In, First Out):
    - Structure: Queue
    - The first item inserted is the first removed (front).
    - Example: people in a line at a store.
- LIFO (Last In, First Out):
    - Structure: Stack
    - The last item inserted is the first removed (top).
    - Example: pile of plates or books.

---

7. Two real-world queue examples (beyond simple “line of people”)

- Print spooler: print jobs enter a queue and the printer processes them in arrival order.
- OS ready queue: processes waiting for CPU time are stored and scheduled in queue order.

---

8. True / False (with short reasons)

a) A stack data structure has a front and a rear.

- False. A stack has only a top; front and rear describe a queue.geeksforgeeks+1

b) A stack can be emptied.

- True. Repeated pops (or clear) can remove all items.

c) In a stack, top refers to the first item pushed onto a stack.

- False. Top is the most recently pushed item.buildingjavaprograms+1

d) The isEmpty operation returns an int value.

- False. It conceptually returns a boolean (true/false).

e) A queue can hold more than one data item.

- True. A queue can hold zero, one, or many items.

f) In a queue, all removals are made at the rear.

- False. Removals (dequeue) are made at the front.

g) The enqueue operation adds an item to the front of the queue.

- False. Enqueue adds at the rear.

h) The first item in a linked list is called the head.

- True. The head points to the first node.

i) A node refers to an item in a stack.

- False in general terminology. A node is a linked-structure element; a stack item is just an element (though a stack can be implemented with nodes).approvedwritershub+1

j) The number of items in a stack or queue can be determined with the length operation.

- True in many definitions. Size/length returns the current number of elements.scribd+1

k) In a linked list, the tail points to null.

- True for a standard singly linked list: the last node's next reference is null.