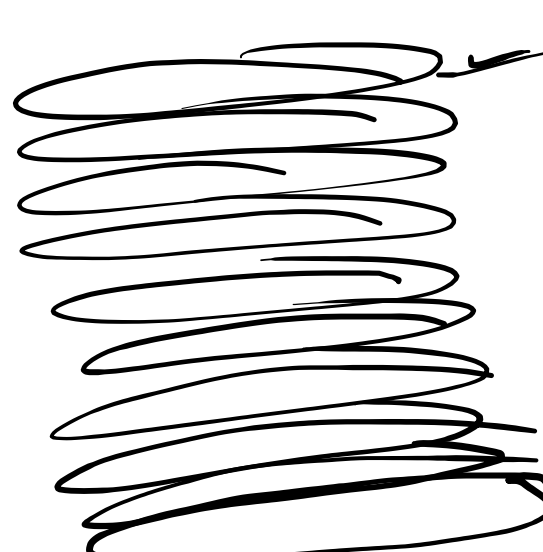
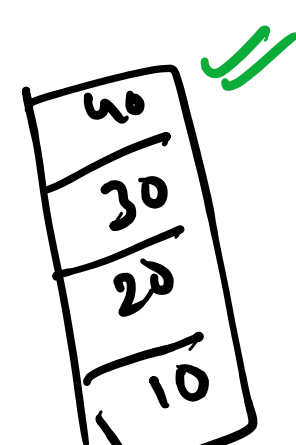
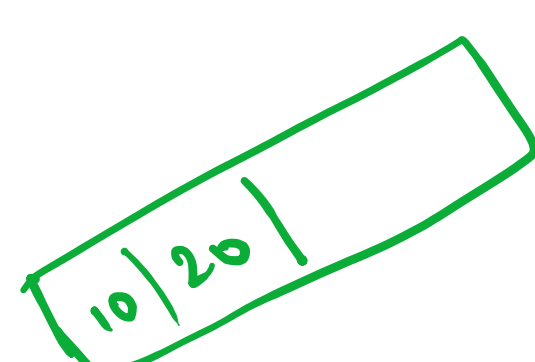


data size

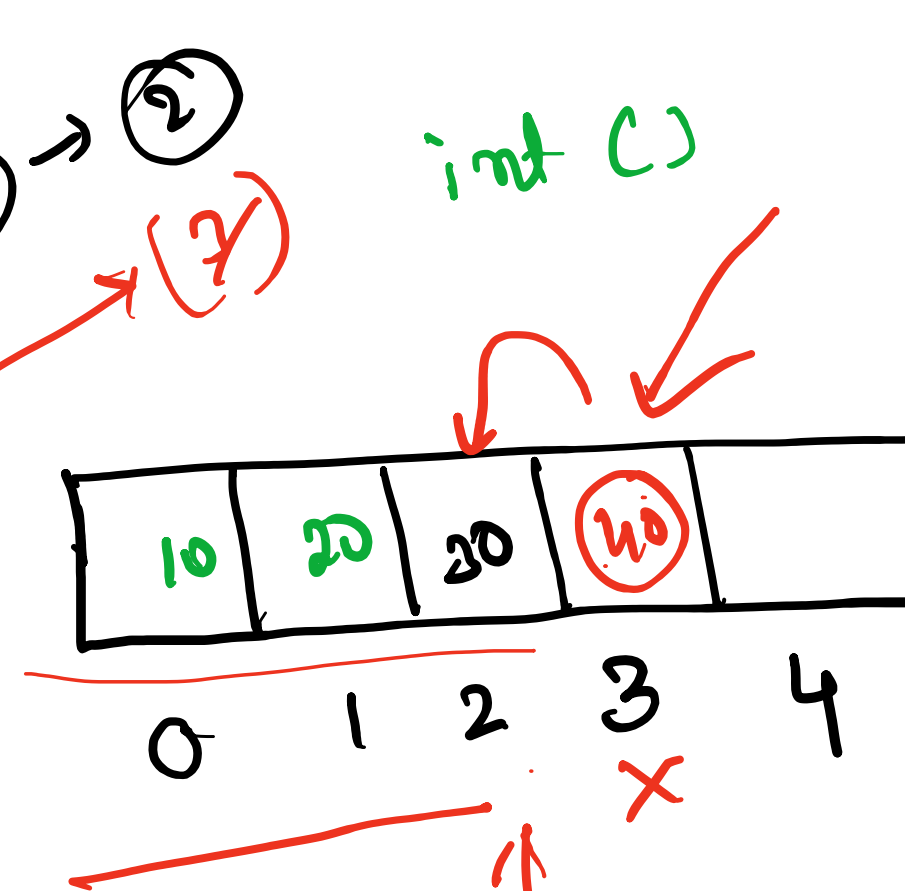
⑪ \Rightarrow {LIFO}



Push (1st 1km) ? i dx

i dx + x

$9x + i dx = 11km$

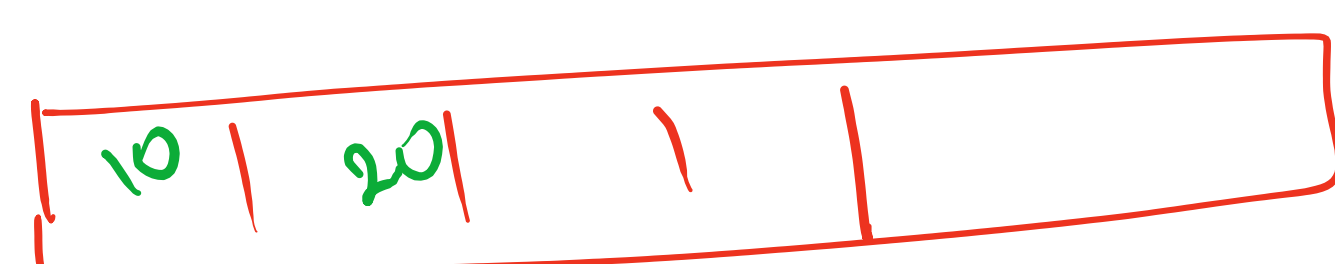


push(10) ✓
push(20) ✓
push(30) ✓
push(40)

Five

(Linear circuit)

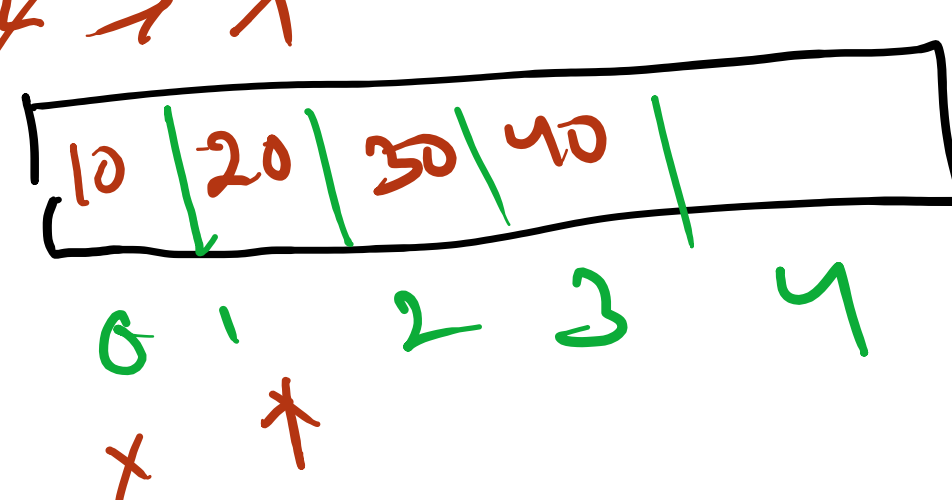
$O(n \cdot size) = 144n$
size 4/



Call
Emails
deance ✓
getFront()
is empty
no front!

ΣΤΟΙΧΕΙΑ

size = 9
direction = 1 km
street + 1



$\Sigma q_m(10) \leftarrow$
 $\Sigma q_m(20) \checkmark$
 $\Sigma q_m(30) \checkmark$
 $\Sigma q_m(40) \cup$
 $\Sigma q_m(50) \checkmark$

```
print v[::-1])
print ++,
size--
```

```

public void Enqueue(int item) throws Exception {
    if (isfull()) {
        throw new Exception("Bklol Queue full hogya hai ");
    }
    → arr[size] = item;
    size++;
}

public int Dequeue() throws Exception {
    if (isEmpty()) {
        throw new Exception("Bklol Queue Empty hai ");
    }
    int v = arr[front];
    front = front + 1;
    size--;
    return v;
}

```

Handwritten notes and diagrams:

- Red arrow from `arr[size]` to `(front + size) / 2` with a checkmark.
- Red box at the bottom right containing `60` and `30` with arrows pointing to them.

size = 0 12 = 0 + 2845

12x 84 23 39 28

hogya hai ");

10m ✓

hai ");

30	40	50
----	----	----

Σ 9m (10) ✓

Σ 2 (20) ✓

Σ 2 (70) ✓

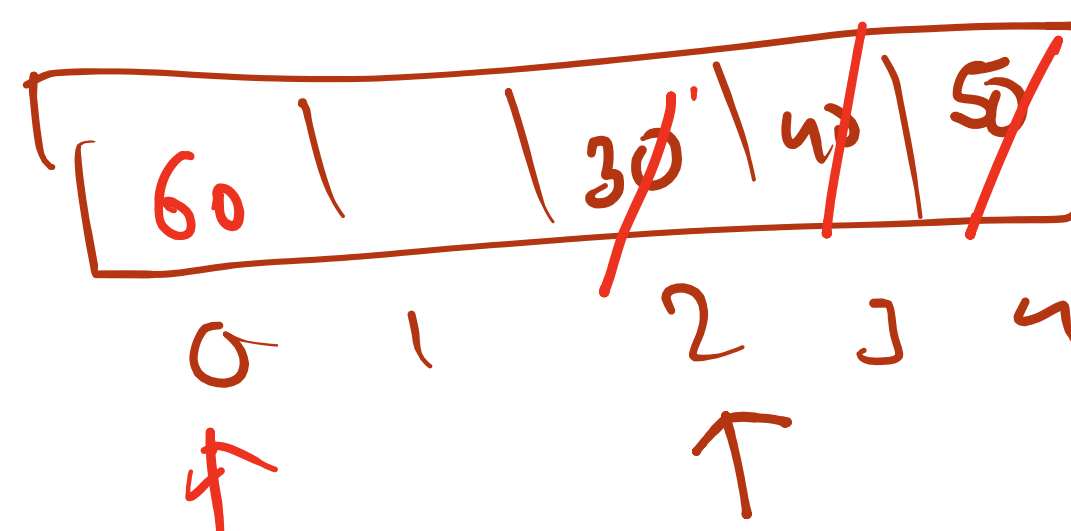
Σ 2 (40) ✓

20 (10) ✓

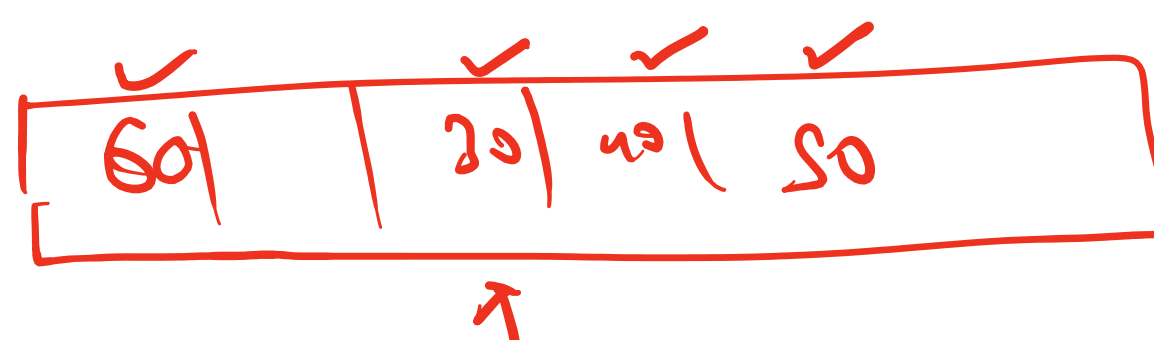
20 (20) ✓

Σ 2 (80)

Σ 2 (60)



$20n = 2 \checkmark$
 $(2+0) \cdot 1 = 2 \checkmark$
 $(2+1) \cdot 1 = 3 \checkmark$
 $(2+2) \cdot 1 = 4 \checkmark$
 $(2+3) \cdot 1 = 5 \checkmark$



for (i = 0; i < size; i++)
 arr[i] = (i + 1) % 10;