

Ans: $[1, 2, 3, 4]$

$\text{foo}(\text{arr}) \rightarrow \underline{\text{clone}} \quad [1, 2, 3, 4]$
 $\text{foo}(\text{arr}, 3) \rightarrow \underline{[2, 3, 4]}$

foo(arr) → clone
name = arr. slice (0)

```

for (arr in n) {
  if (n == null) {
    arr[1] = 0
  } else {
    // ...
  }
}

```

$(1, 2, 13, 14)$
 $n = 2$
 $k = 4$
 $s = 72$
 $\{11, 12\}$

arr.slice(arr.length-n)

If $n \geq n_0$

first 3 elements
5
first n elements

$$(0, \text{Math.abs}(n))$$

B_{min} 5-6

T.C.
S.C.

creat redox demo

SOC. Compost
Battis

redux
bat Redux JS
- redux.js

A hand-drawn diagram illustrating the MapReduce system architecture. The components and their interactions are as follows:

- Reducers:** A box labeled "Reducer" is part of a group indicated by a bracket and the label "bats: 20". It receives "initial state defn" from the "Driver (main)".
- Driver (main):** A box labeled "Driver (main)" that sends data to the "Reducer" and receives data from the "Component".
- Store:** A box labeled "store" that receives data from the "Reducer" and sends data to the "Component". It is annotated with "create a global store".
- Component:** A box labeled "Component" that sends data to the "Driver (main)" and receives data from the "store". It is annotated with "mapState, prop, mapDispatchToProps" and "set".

Arrows indicate the flow of data and state between these components.

A handwritten diagram illustrating the MapReduce process. At the top, 'user job' is written with a circled '20' next to it. Below this, 'input' is written. The flow continues to 'Reducer' and 'Store', which are connected by a double-headed arrow. From 'Store', an arrow points to 'get'. Below 'get' is 'Action', which has an arrow pointing to 'n'. From 'Action', an arrow points to 'Dispatch'. Below 'Dispatch', there are two large curved arrows forming a cycle, labeled '2' and '3'. Finally, an arrow points down to 'reducer', which is underlined.

by 20 17 20

sell-but

by

```
type: buy-but  
payback: ✖  
}
```

but Redux
state
?

bats

3

Combined Reducers

ball: {balls: 50}
bat: {bats: 50}

3