7. DELETE ELEMENT AT POSITION IN AN ARRAY

APPROACH:

$Suppose\ we\ have\ size = 5\ and$

$And\ indices: a[0], a[1], a[2], a[3]\ and\ a[4].$

$And\ we\ want\ to\ delete\ element\ at\ index: 2$

$Which\ means\ we\ have\ to\ delete\ index: 2\ too.$

$Then\ we\ go\ through\ a\ process\ of\ Swapping:$

$a[2-1=1] = a[2+1=3] \longrightarrow elem: 3$

$a[3-1=2] = a[3+1=4] \longrightarrow elem: 4$

$a[4-1=3] = a[4+1=5] \longrightarrow elem: 5$

$And\ a[0] will\ be\ remain\ untouched.$

$Then\ we\ decrement\ the\ size:$

$size = size - 1\ , now\ size\ is\ 4, traversal\ will\ take\ place\ from\ a[0] = 1, a[1] = 3, a[2] = 4,\ and\ a[3] = 5\ .$

PROGRAM:

```
for (int i = pos-1 ; i <size-1; i++)
        {

            a[i] = a[i + 1];

        }
 size=size-1;
```

TIME COMPLEXITY OF DELETE ELEMENT AT POSITION IN AN ARRAY

$$As, loop\ goes\ from\ position - 1\ to\ size - 1\ i.e.$$

$$1\ to\ n\ times = O(n) and\ size\ decreased\ is: size - 1$$

$$= O(1)\ , hence\ total: O(n) + O(1) = O(n).$$

❖❖❖❖❖❖❖❖❖❖❖❖❖❖