

Arrays Challenge (Max till i)

Problem:

Given an array $a[]$ of size n . For every i from 0 to $n-1$ output $\max(a[0], a[1], \dots, a[i])$.

Example:

	1	0	5	4	6	8
	(0)	(1)	(2)	(3)	(4)	(5)
Max till i:	1	1	5	5	6	8

Approach:

1. Keep a variable mx which stores the maximum till i^{th} element.
2. Iterate over the array and update,
 $mx = \max(mx, a[i])$

Iterations:

- At $i = 0$:

Given Array:

0	-9	1	3	-4	5
---	----	---	---	----	---



$mx = 0$

- At $i = 1$:

Given Array:

0	-9	1	3	-4	5
---	----	---	---	----	---



$mx = 0$

- At $i = 2$:

Given Array:

0	-9	1	3	-4	5
---	----	---	---	----	---



$mx = 1$

- At $i = 3$:

Given Array:

0	-9	1	3	-4	5
---	----	---	---	----	---



$mx = 3$

- At $i = 4$:

Given Array:

0	-9	1	3	-4	5
---	----	---	---	----	---



$mx = 3$

- At $i = 5$:

Given Array:

0	-9	1	3	-4	5
---	----	---	---	----	---



$mx = 5$

Code:

```
int main()
{
    int n;
    cin >> n;

    int a[n];
    for(int i=0; i<n; i++)
    {
        cin >> a[i];
    }

    int mx = -199999;
    for(int i=0; i<n; i++)
    {
        mx = max(mx, a[i]);
        cout << mx << endl;
    }
    return 0;
}
```

Time Complexity: $O(n)$.

Apni Kaksha