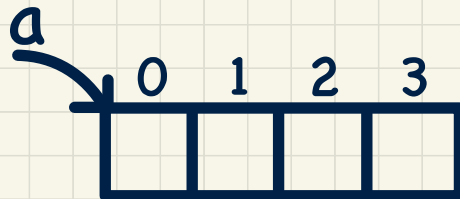


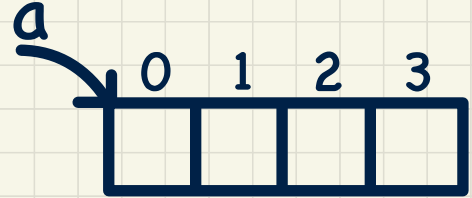
Iterative Algorithm: Precondition and Postcondition

```
1  int findMax (int[] a) {  
2    assert Q; /* precondition satisfied? */  
3    int i = 0;  
4    int result = a[i];  
5    assert LI; /* invariant established? */  
6    while(i != a.length) {  
7      if(a[i] > result) {  
8        result = a[i];  
9      }  
10     i = i + 1;  
11     assert LI; /* invariant preserved? */  
12   }  
13   assert R; /* postcondition satisfied? */  
14   return result;  
15 }
```



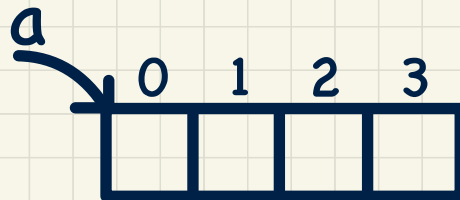
Iterative Algorithm: Loop Invariant (1)

```
1  int findMax (int[] a) {  
2    assert Q; /* precondition satisfied? */  
3    int i = 0;  
4    int result = a[i];  
5    assert LI; /* invariant established? */  
6    while(i != a.length) {  
7      if(a[i] > result) {  
8        result = a[i];  
9      }  
10     i = i + 1;  
11     assert LI; /* invariant preserved? */  
12   }  
13   assert R; /* postcondition satisfied? */  
14   return result;  
15 }
```



Iterative Algorithm: Loop Invariant (2)

```
1  int findMax (int[] a) {  
2    assert Q; /* precondition satisfied? */  
3    int i = 0;  
4    int result = a[i];  
5    assert LI; /* invariant established? */  
6    while(i != a.length) {  
7      if(a[i] > result) {  
8        result = a[i];  
9      }  
10     i = i + 1;  
11     assert LI; /* invariant preserved? */  
12   }  
13   assert R; /* postcondition satisfied? */  
14   return result;  
15 }
```



Iterative Algorithm: Correctness

```
1  int findMax (int[] a) {  
2    assert Q; /* precondition satisfied? */  
3    int i = 0;  
4    int result = a[i];  
5    assert LI; /* invariant established? */  
6    while(i != a.length) {  
7      if(a[i] > result) {  
8        result = a[i];  
9      }  
10     i = i + 1;  
11     assert LI; /* invariant preserved? */  
12   }  
13   assert R; /* postcondition satisfied? */  
14   return result;  
15 }
```

