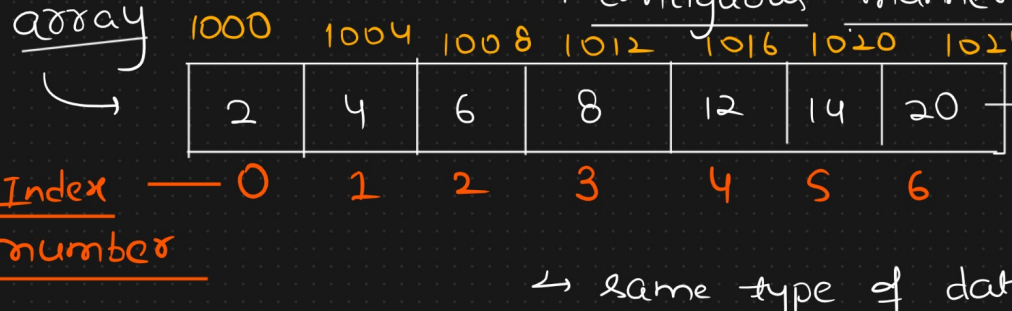
PW SKILLS

Array

↳ Linear Data Structure

↳ contiguous manner



↳ same type of data

↳ $n = 7$ Start index = 0

Last index = $n-1$

↳ Random access

array[3] = 8

↳ array.length = n

↳ size of an array

Data Structure

- Linear Data Structure
  - ↳ Array
  - ↳ Linked List
  - ↳ Stack & Queue
- Non-Linear Data Structure
  - ↳ Tree
  - ↳ Graph

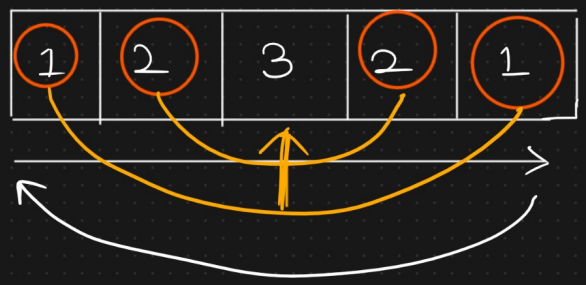
Array Implementation

```
int[] array = {2, 4, 6, 8, 10, 12, 14}
int array[] = {2, 4, 6, 8, 10, 12, 14}
```

Palindrome

mid = 5/2 = 2

array



0 to 4
1, 2, 3, 2, 1
1, 2, 3, 2, 1
4 to 0

Equal to each other

n = 5

Approach 1

new_array

0	1	2	3	4
1	2	3	2	1

Palindromic array

Approach 1

Time complexity = O(n)

Space complexity = O(n)

can we have some approach where we can reduce time & space complexity ??

n = 5 5/2 = 2

i=0

1 != 1

i=1

2 != 2

i = 2 < 2

```
for(i=0; i<n/2; i++) {
    if (arr(i) != arr(n-1-i))
        print (Not a palindromic array)
}
```

Print (Palindromic array)

Approach 2

Time complexity = O(n)

Space complexity = O(1)

↳ optimized approach