Summary → Arrays

→ Collection of similar datatype which is Continuous in memory

0	12	13	15	12	11

int

Syntax

Datatype of [ ]   Name of   =   new Datatype [Size]
Array   Array   Size   of
Array

int =>   int [ ] arr = new int [10]

0	1	2	3	4	5	6	7	8	9
11	12	10	13	9	4	3	5	6	8
1	2	3	4	5	6	7	8	9	10

Position

Idx = Position - 1

datatypes -) Prim or non Prim

String -) String [] arr = new String [10]

0 1   2 3   4 5   6   7 8 9
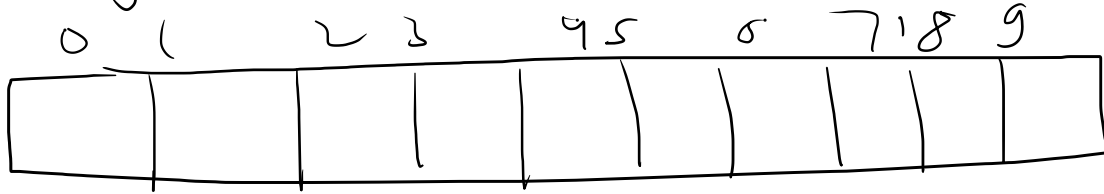
float -) float [] arr = new float [10]

double

long

Continous in mm   Non Prem

                                                         → motorows 10 bysis   hid → 4 bysts
                                                              of Vmay   of
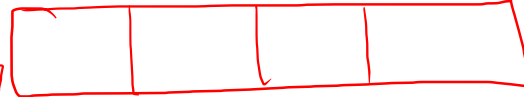   memory
                                        ↓
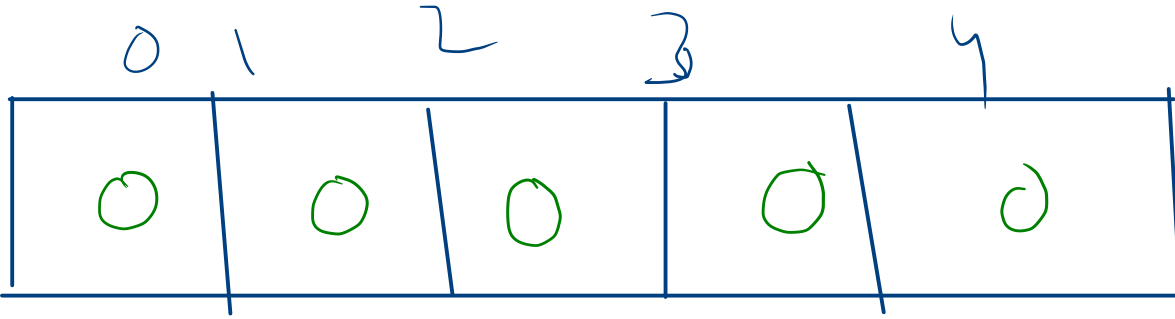
                                        0   1   2   3

Primitive
(int, blost, double
  long, char
  boolen) | 2k | 2k+4 | 2k+8 | 2k+12 |
        +   Refor
 References of   of
 Non Prom   arr

         Stuck   Heap memory
         memory
                         Ver
Referrence →   int [ ]   to   arr = [new]   int [size]   motorows
 Variable ←   LHS   RHS   of
                                                                        Memory

0	1	2	3	4
O	O	O	O	O

```
// 2 ⟶ t1 = t2
System.out.println(arr[0]); // ⟶ t1  ⟶ O(1)
System.out.println(arr[4]); // ⟶ t2  ⟶ O(1)
```

$$T_1 = T_2$$

arr

0	1	2	3	4
10	11	12	13	14

2h   2h+4   2h+8   2h+12   2h+16

arr[0] ⟶   2h ⟶ O(1)

arr[i] = addrs (arr[0]) ⟵   O(1)

1 x y ⟶ int
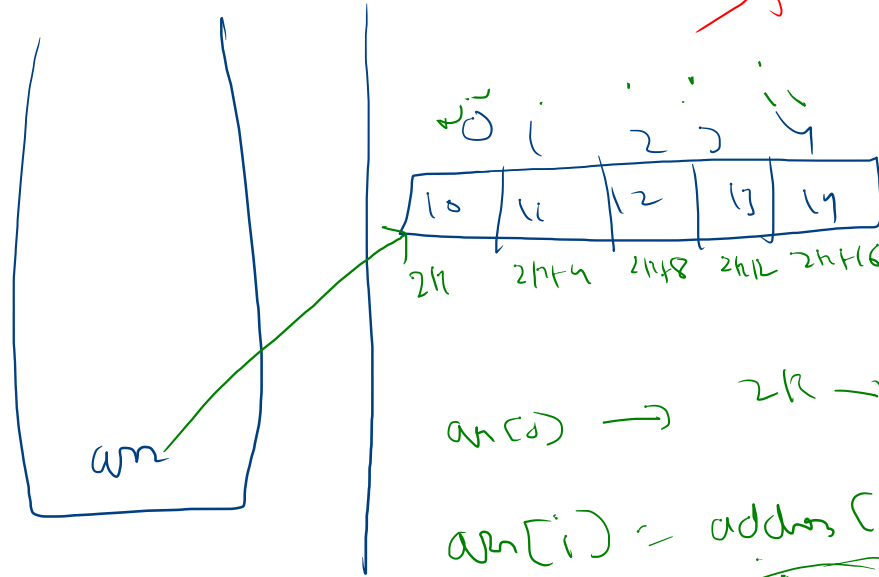
2h + y x y ⟹ 2h+16

Find Element In An Array

$n \to z|p = 5$.

	0	1	2	3	4
	5	2	1	10	12

$d \to z|p$

1). $d = 10 \to$ Present $\to 3$

2). $d = 15 \to$ Not Past $\to -1$

if $(arr[i] == d) \{$
  return i
3

$2 == 10$ ✗
$1 == 10$ ✗
$10 == 10$ ✓ $\to$ return 3

```
public static void findElementInArr(int[] arr, int d) {

    for (int i = 0; i < arr.length; i++) {
        if (arr[i] == d) {
            return i;
        }
    }
    return -1;
}
```
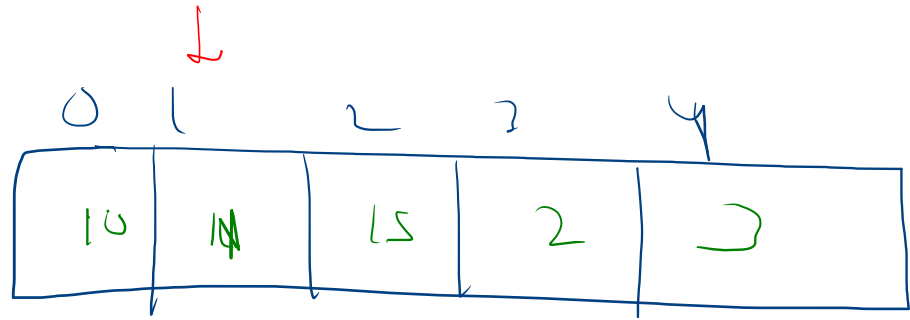

d = 15

i=0 → 10 == 15 ✗

i=1 → 11 == 15 ✗

i=2 → 15 == 15 ✓

d = 100

i=0 → 100 == 10 ✗

i=1 → 100 == 11 ✗

i=2 →

i = 3

i = 4

Min_Max_Array

$$Span = Max - Min$$

0	1	2	3	4	5
10	-2	5.	-1	0	12

SIZE = 6

max = 12 ⎫
min = -2 ⎬
             ⎭

Span = 12 - (-2)
     = 12 + 2
     = 14

→ Max = Integer.MIN_VALUE   ~~10~~ 12
       -2147---

→ Min = Integers.MAX_VALUE   ~~10~~
       247 --/-
             -2

If (Max < arr [i])
    Max = arr [i]

↓
If (Min > arr [i]) [
    Min = arr [i]
]