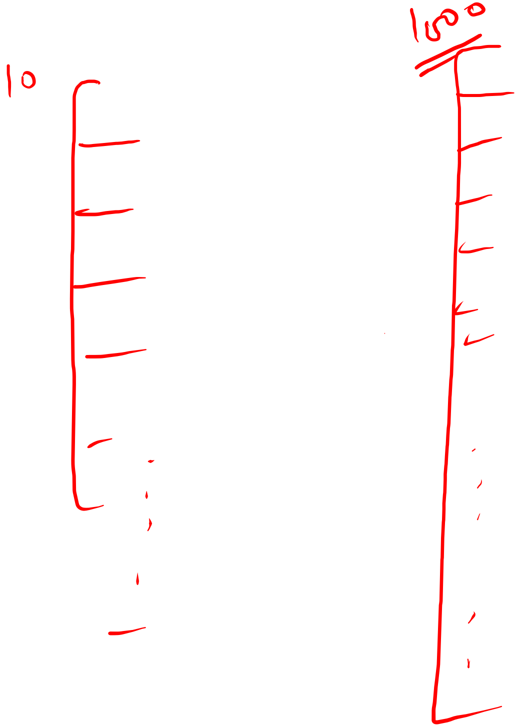
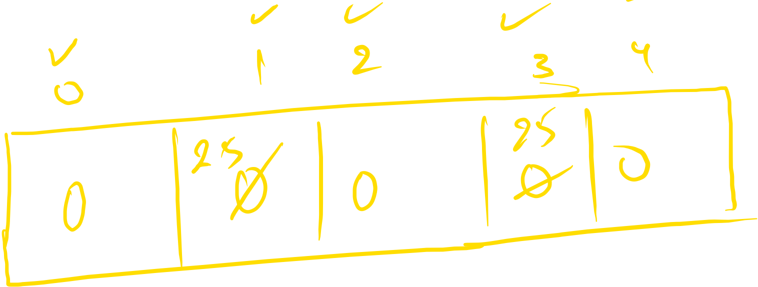
→ Variable



✓ Array → Continuous memory of same type

↳ Declare → int arr[] = new int[5];

arr =



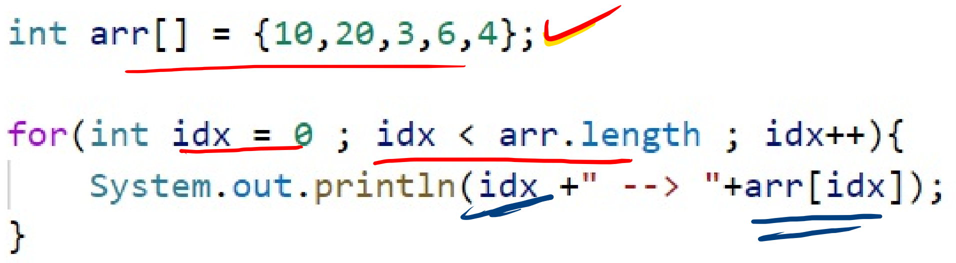
0	1	2	3	4
0	25 ~~0~~	0	25 ~~0~~	0

arr[1] = 25

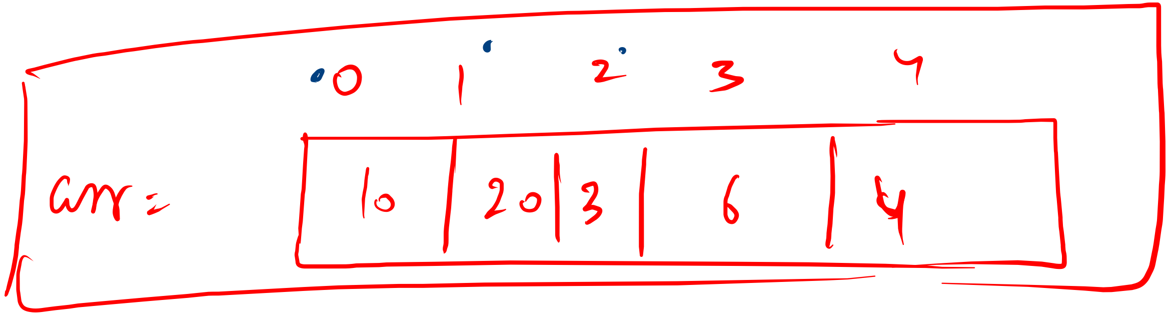
arr[3] = arr[2];

Index = (0 → length-1)

(arr.length)

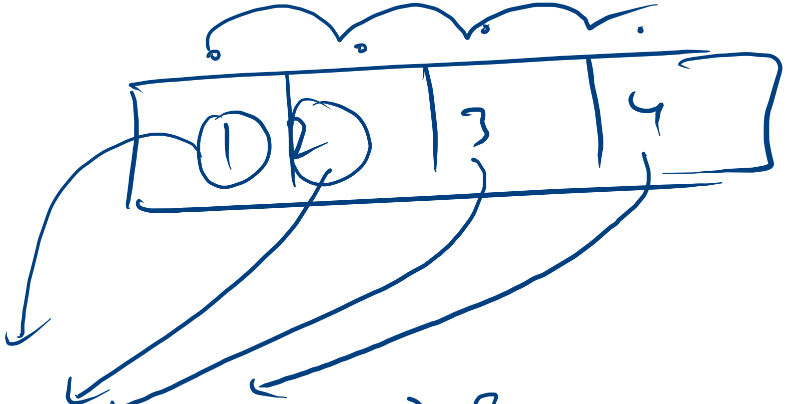


```
int arr[] = {10,20,3,6,4};

for(int idx = 0 ; idx < arr.length ; idx++){
    System.out.println(idx +" --> "+arr[idx]);
}
```

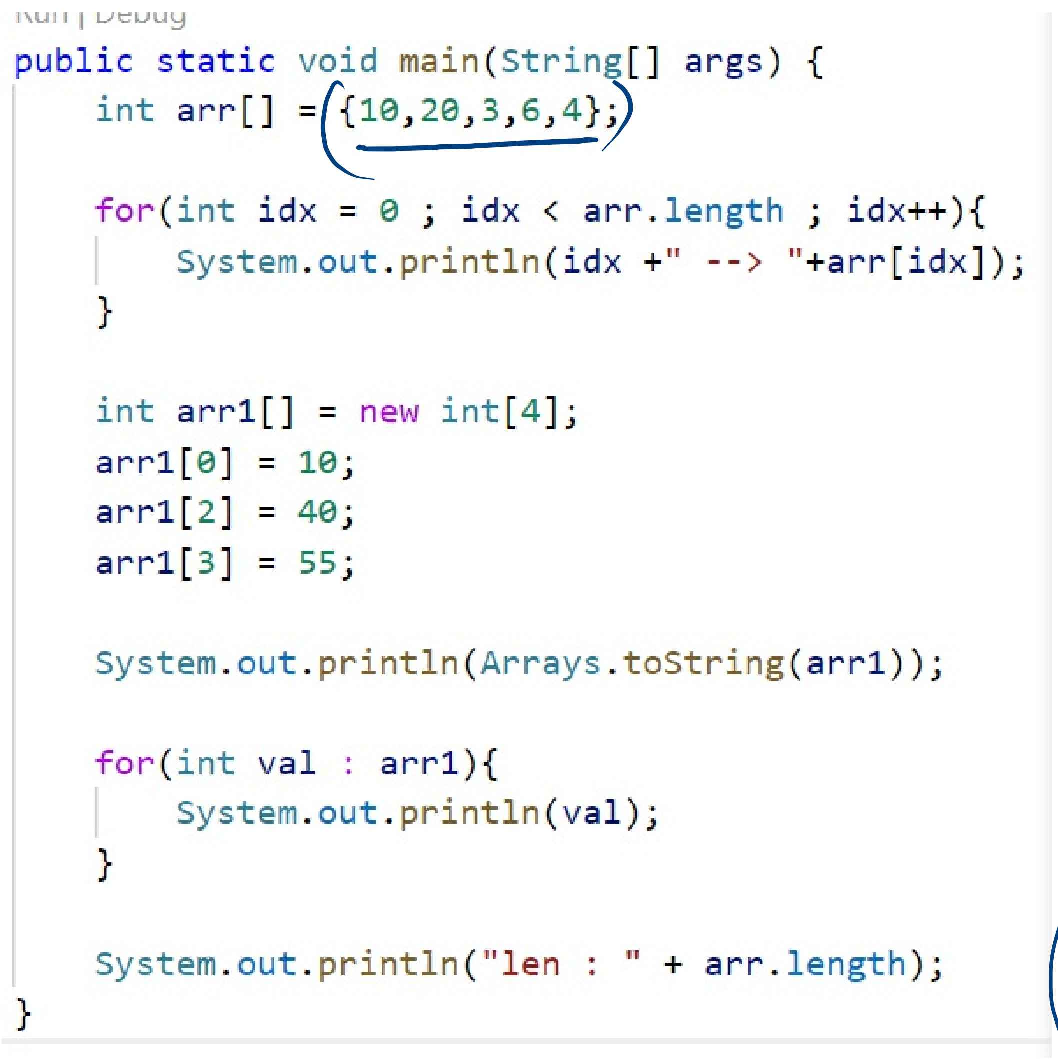


```
0 --> 10
1 --> 20
2 --> 3
3 --> 6
4 --> 4
```

for each (array)



```
for ( int val : arr ) {

}
```



```
Run | Debug
public static void main(String[] args) {
    int arr[] = {10,20,3,6,4};

    for(int idx = 0 ; idx < arr.length ; idx++){
        System.out.println(idx +" --> "+arr[idx]);
    }

    int arr1[] = new int[4];
    arr1[0] = 10;
    arr1[2] = 40;
    arr1[3] = 55;

    System.out.println(Arrays.toString(arr1));

    for(int val : arr1){
        System.out.println(val);
    }

    System.out.println("len : " + arr.length);
}
```

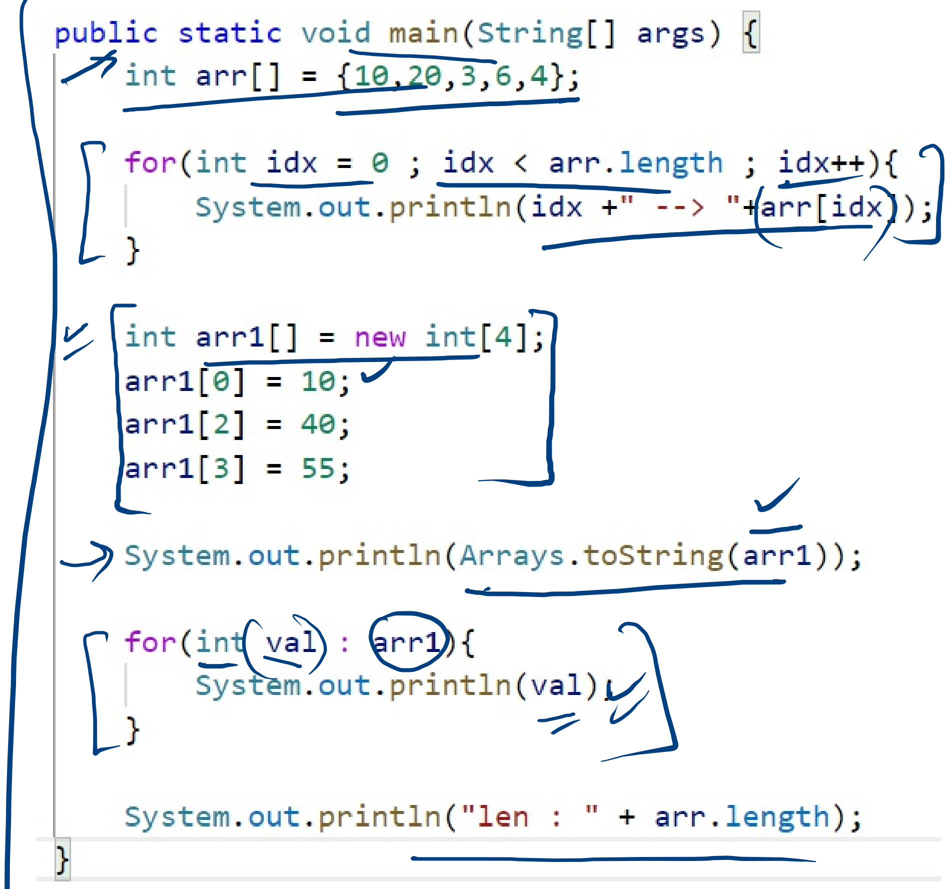
"[10, 20, 3, 6, 4]"

int → 0

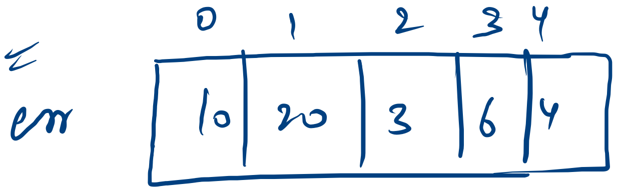
double/float → 0.0

boolean → false

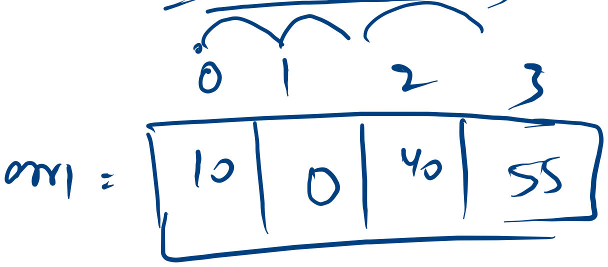
string → null



```
public static void main(String[] args) {
    int arr[] = {10,20,3,6,4};

    for(int idx = 0 ; idx < arr.length ; idx++){
        System.out.println(idx +" --> "+arr[idx]);
    }

    int arr1[] = new int[4];
    arr1[0] = 10;
    arr1[2] = 40;
    arr1[3] = 55;

    System.out.println(Arrays.toString(arr1));

    for(int val : arr1){
        System.out.println(val);
    }

    System.out.println("len : " + arr.length);
}
```



Index -> 0 -> (len-1)



[10, 0, 40, 55]

10

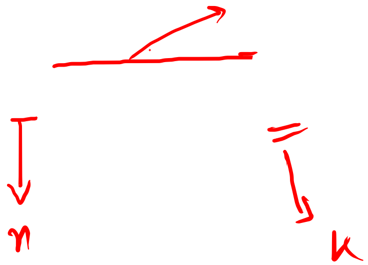
0

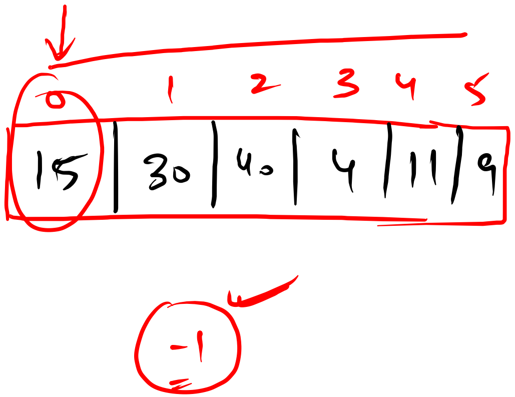
40

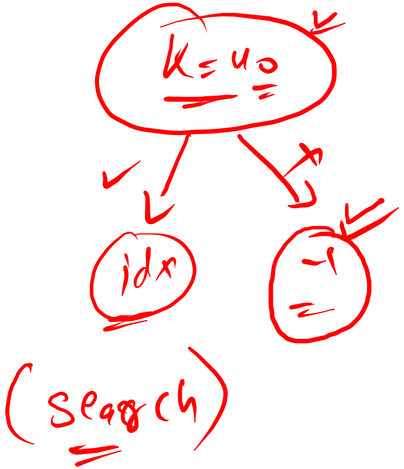
55

(len : 5)

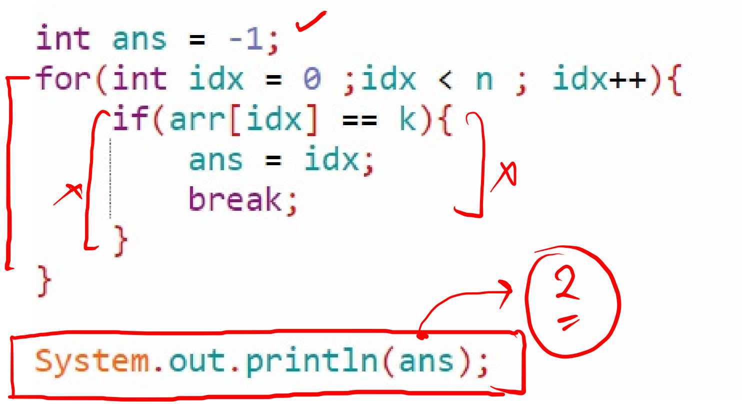
0	-->	10
1	-->	20
2	-->	3
3	-->	6
4	-->	4





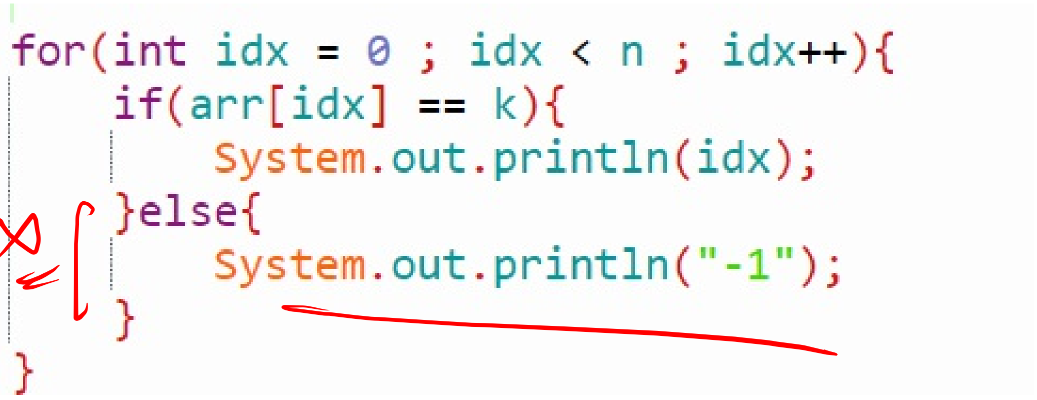


ans = 2



```
int ans = -1;
for(int idx = 0 ;idx < n ; idx++){
    if(arr[idx] == k){
        ans = idx;
        break;
    }
}

System.out.println(ans);
```

2



```
for(int idx = 0 ; idx < n ; idx++){
    if(arr[idx] == k){
        System.out.println(idx);
    }else{
        System.out.println("-1");
    }
}
```

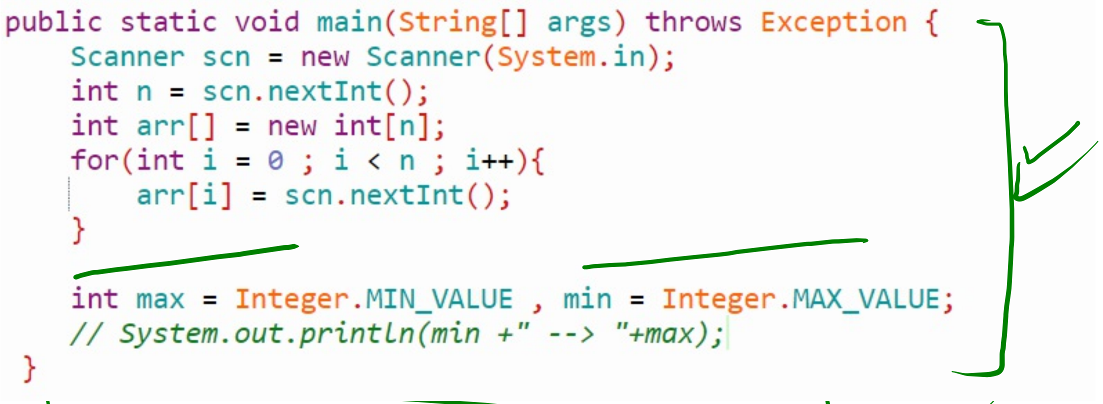
Span => Max - Min

arr =

0	1	2	3	4	5
15	30	40	4	11	9

max = ~~-∞~~ ~~15~~ ~~30~~ 40

min = ~~+∞~~ ~~15~~ 4



```
public static void main(String[] args) throws Exception {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int arr[] = new int[n];
    for(int i = 0 ; i < n ; i++){
        arr[i] = scn.nextInt();
    }

    int max = Integer.MIN_VALUE , min = Integer.MAX_VALUE;
    // System.out.println(min +" --> "+max);
}
```

val + 0 => val

val - 0 => val

val * 1 => val

36

Min (val, +∞) => val

Max (val, -∞) => val

+∞ → Integer.MAX_VALUE;

-∞ → Integer.MIN_VALUE

Span => 40-4 => 36

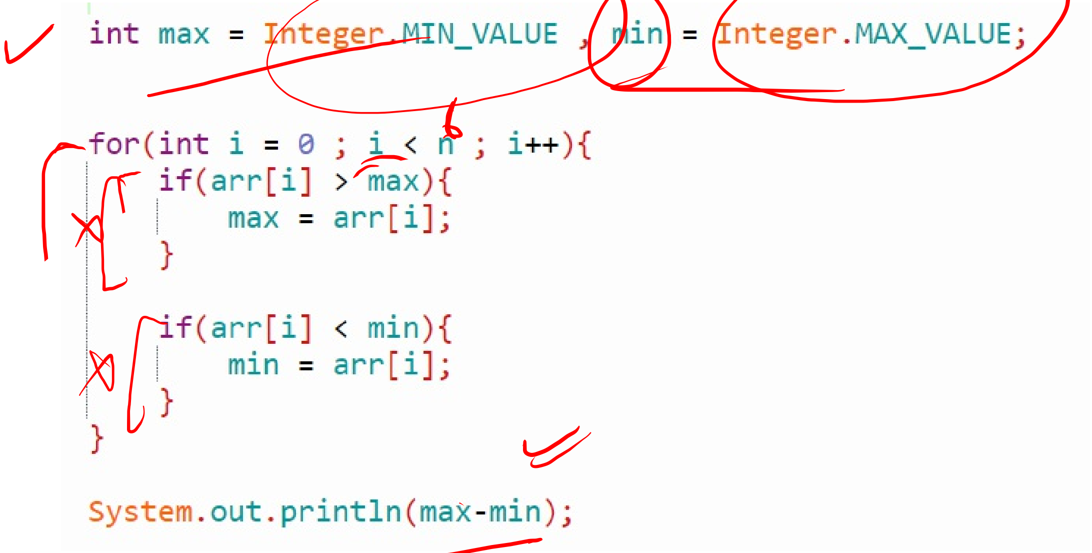
arr =

0	1	2	3	4	5
15	30	40	4	11	9

max = ~~-∞~~ ~~15~~ ~~30~~ 40

min = ~~+∞~~ ~~15~~ 4

Spend Max - mi



```
int max = Integer.MIN_VALUE , min = Integer.MAX_VALUE;

for(int i = 0 ; i < n ; i++){
    if(arr[i] > max){
        max = arr[i];
    }

    if(arr[i] < min){
        min = arr[i];
    }
}

System.out.println(max-min);
```

6

36

max = 7

st = 3

row = 0

idx = 0

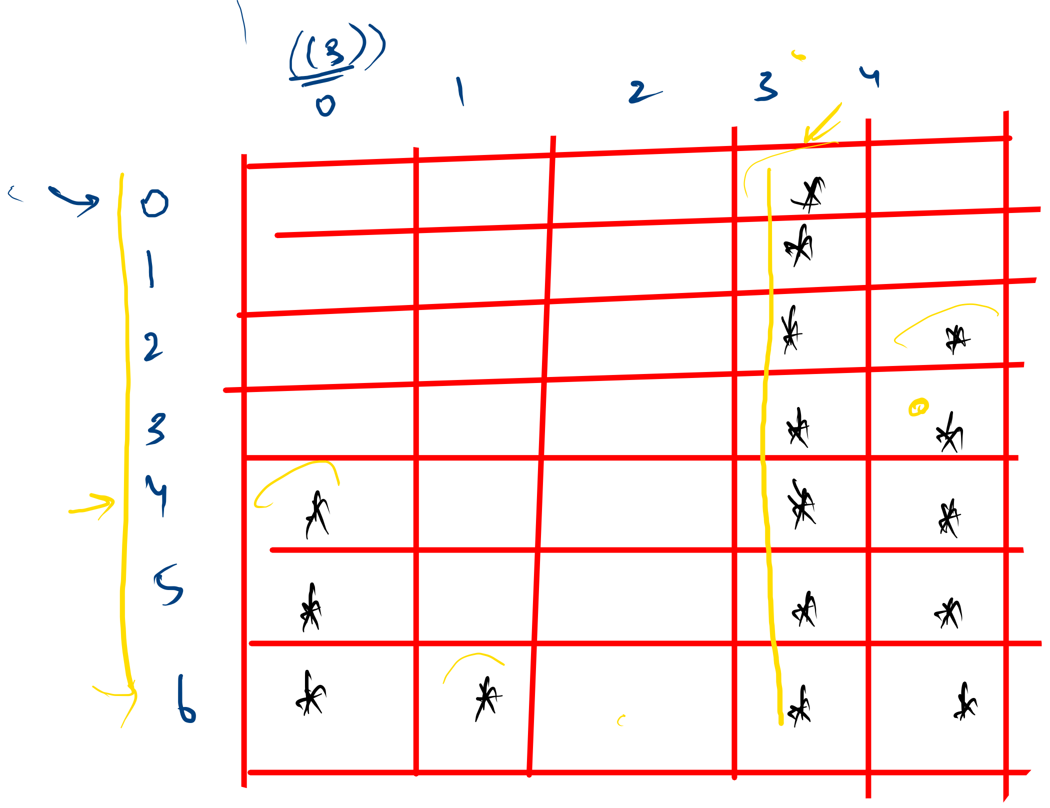
7 - 5

2

7 - 3 = 4

7 - 1 = 6

6



$arr[c_i] = \frac{\#stars}{}$

↳ column

$n_r = maxlen$

$n_c \Rightarrow arr.len$

7 - 3

✓	✓	—	✓	✓
3	1	0	7	5

$(r >= max - arr[idx])$

7 - 0

7

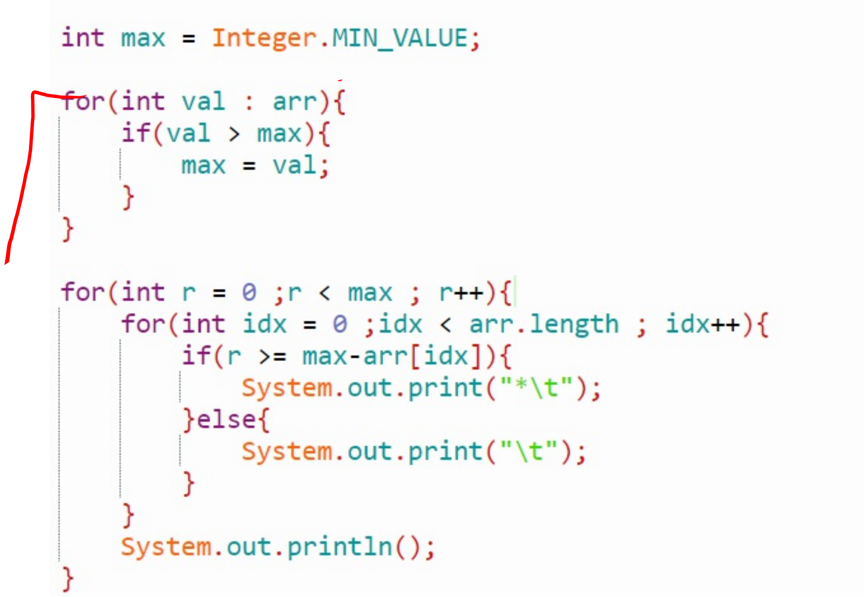
7 - 7 = 0

	0	1	2	3	4
arr =	3	1	0	7	5

max = ~~-∞~~ ~~3~~ 7

r	idx
0	~~0~~ ~~1~~ ~~2~~ ~~3~~ 4
1	~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

	idx	0(3)	1(1)	2(0)	3(7)	4(5)
r	0	—	—	—	*	—
✓	1	—	—	—	*	—
	2					
	3					
	4					
	5					
	6					

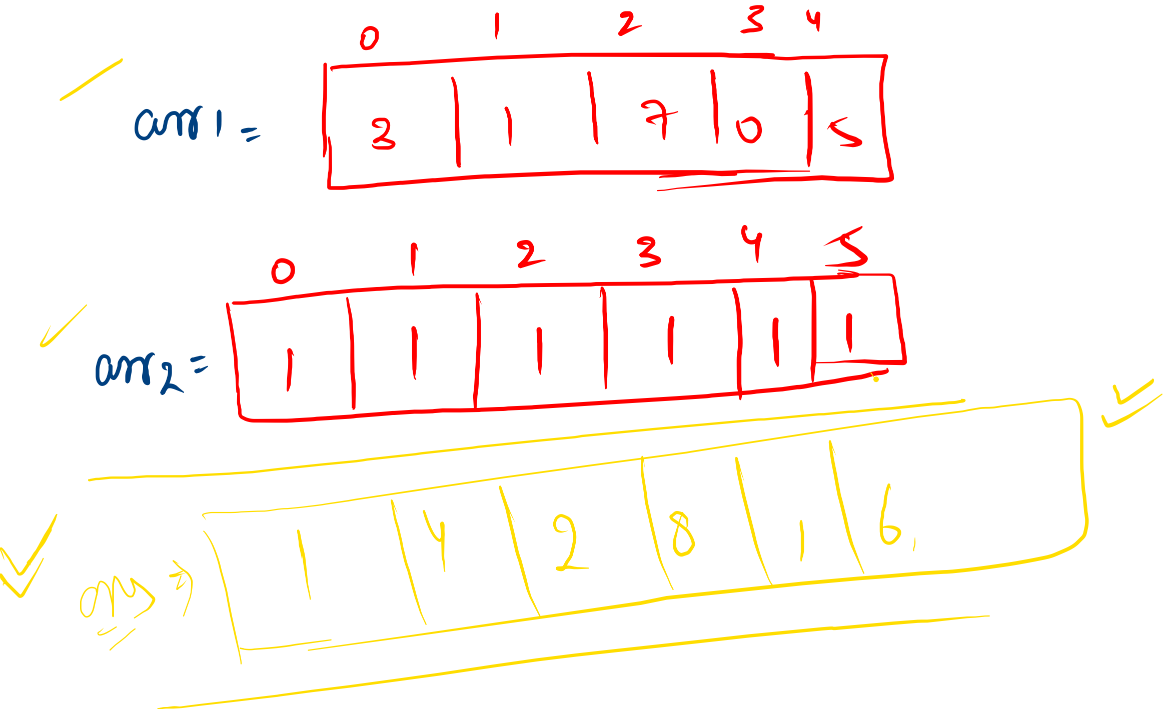


```
int max = Integer.MIN_VALUE;

for(int val : arr){
    if(val > max){
        max = val;
    }
}

for(int r = 0 ;r < max ; r++){
    for(int idx = 0 ;idx < arr.length ; idx++){
        if(r >= max-arr[idx]){
            System.out.print("*\t");
        }else{
            System.out.print("\t");
        }
    }
    System.out.println();
}
```

r >= max - arr[idx]

1 >= 7 - 5

5
3
1
0
7
5
6
1
1
1
1
1
1



arr1 =

0	1	2	3	4
3	1	7	0	5

arr2 =

0	1	2	3	4	5
1	1	1	1	1	1

ans =>

1	4	2	8	1	6,

0

arr_1 =

1

0	1	2
1	2	6

$P_1 = \cancel{2}\,\cancel{1}\,\cancel{0}\,\cancel{-1}\,-2$

v_1 =

$Carry = \cancel{0}\,1$

$\rightarrow Sum = v_1 + v_2 + Carry$

arr_2 =

1

0	1	2	3
9	9	8	7

=

=

$P_2 = \cancel{3}\,\cancel{2}\,\cancel{1}\,\cancel{0}\,-1$

v_2 =

res =

1

0	1	2	3
0	1	1	3

$P_3 = \cancel{3}\,\cancel{2}\,\cancel{1}\,\cancel{0}\,-1$

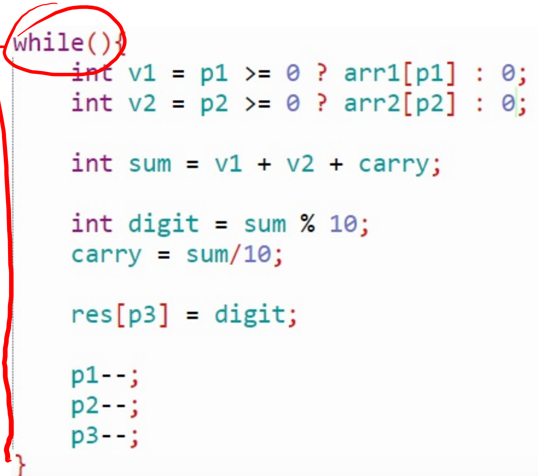
digit = Sum % 10

=

=

Carry = Sum / 10 =

✓



```
while(){
    int v1 = p1 >= 0 ? arr1[p1] : 0;
    int v2 = p2 >= 0 ? arr2[p2] : 0;

    int sum = v1 + v2 + carry;

    int digit = sum % 10;
    carry = sum/10;

    res[p3] = digit;

    p1--;
    p2--;
    p3--;
}
```

1

$$\begin{array}{ccc} & 1 & \\ 9 & 9 & 9 \\ 9 & 9 & 9 \\ \hline & & \boxed{18} \\ & & 8 \end{array}$$

Anybase Subtraction

H.W.

Borrow = ~~0~~ ~~1~~ 0

arr2 =

0	1	2	3
1	0	0	0

arr1 = 0

0	1	2
3	2	7

res =

0	1	2	3
0	6	7	3

$p_2 = \cancel{0}\ -1$

$v_2 = 1$

$p_1 = \cancel{-1}\ -2$

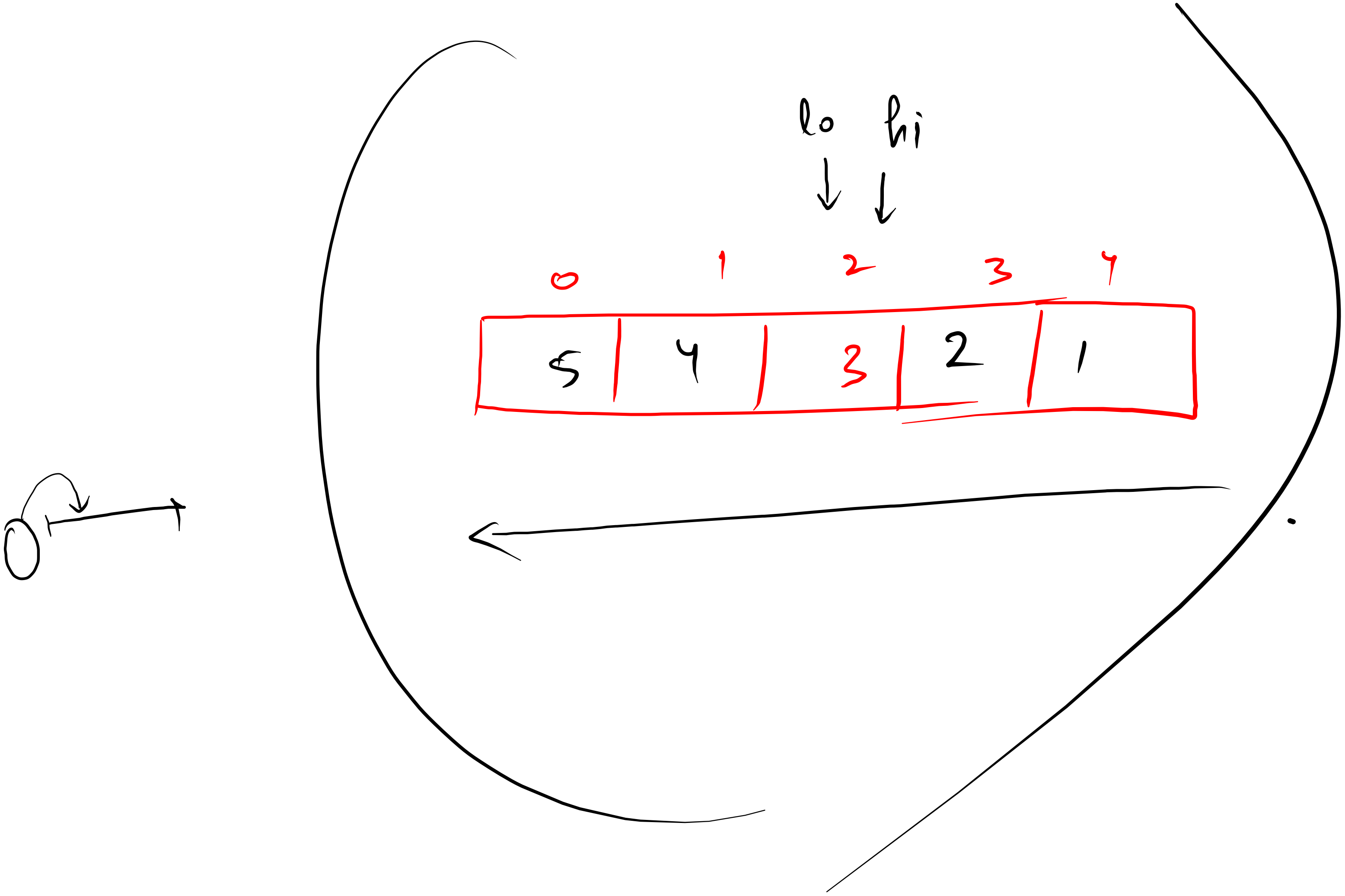
$v_1 = 0$

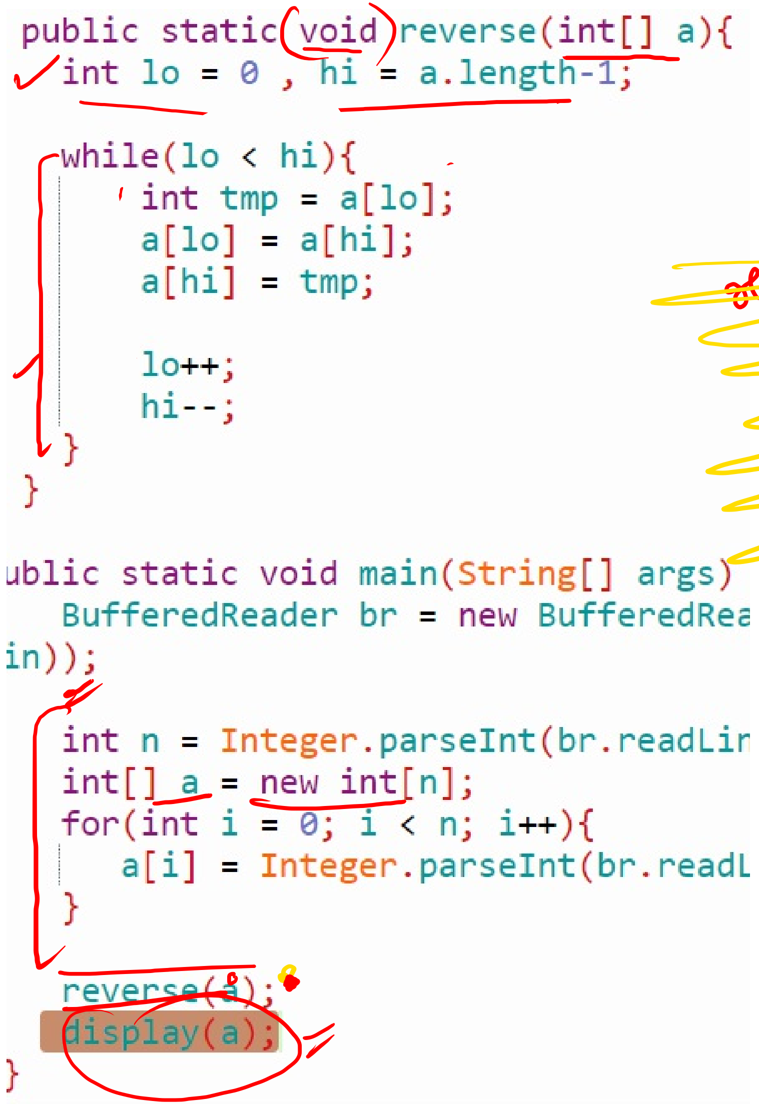
$p_3 = \cancel{0}\ -1$

$Diff = v_2 - v_1 - borrow$

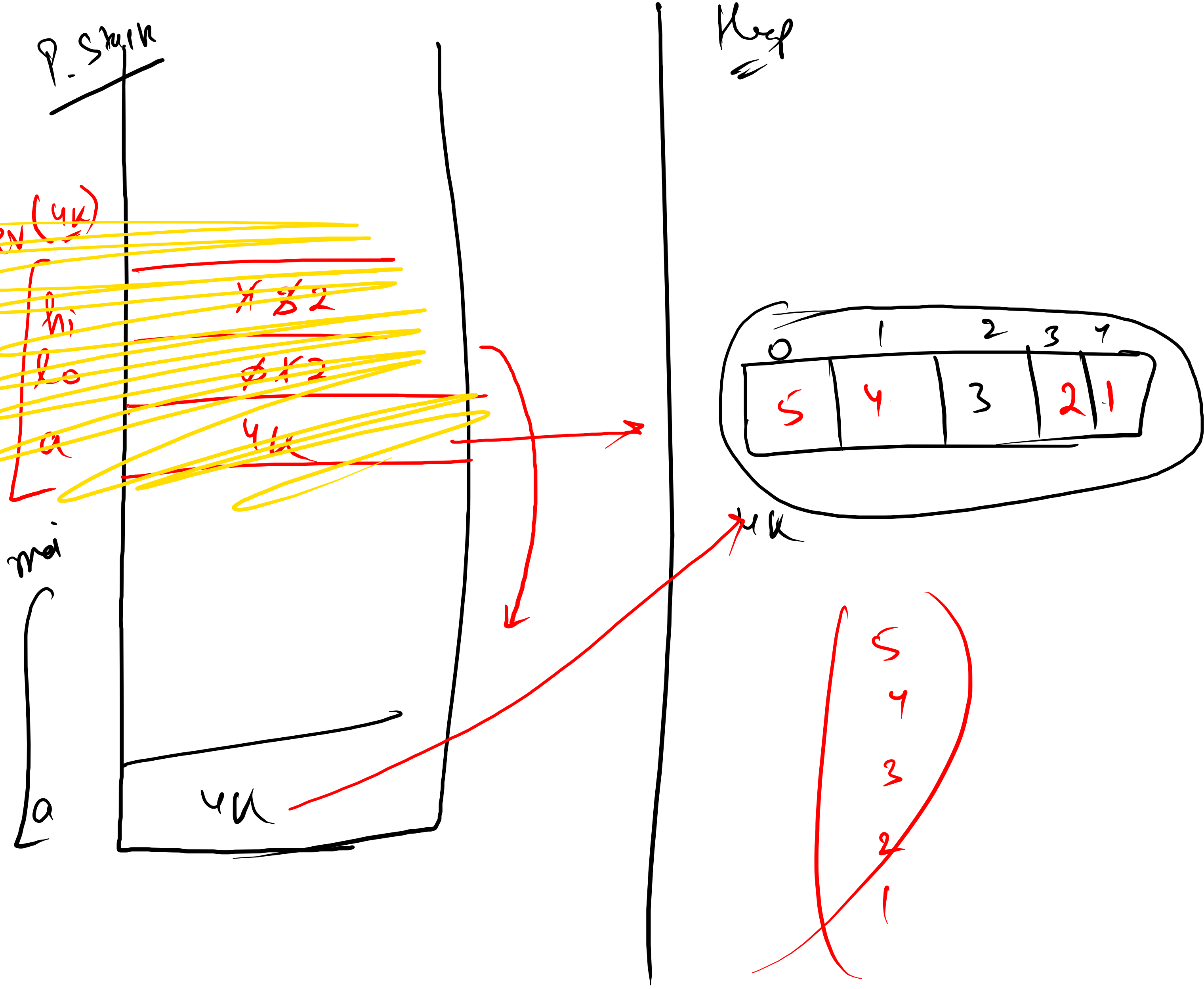
$= 1 - 0 - 1 \Rightarrow \underline{0}$

```
if(diff < 0){
    diff = diff + 10;
    borrow = 1;
}else{
    borrow = 0;
}
```


lo
hi
0
1
2
3
4
5
4
3
2
1
0



```
public static void reverse(int[] a){
 int lo = 0 , hi = a.length-1;

  while(lo < hi){
    int tmp = a[lo];
    a[lo] = a[hi];
    a[hi] = tmp;

    lo++;
    hi--;
  }
}

ublic static void main(String[] args)
  BufferedReader br = new BufferedRea
in));

  int n = Integer.parseInt(br.readLin
  int[] a = new int[n];
  for(int i = 0; i < n; i++){
    a[i] = Integer.parseInt(br.readL
  }

  reverse(a);
  display(a);
}
```



Thurs → 06:00-10:00

✓ Sat

Sun] → 10:00 - 02:00