function - Syntax

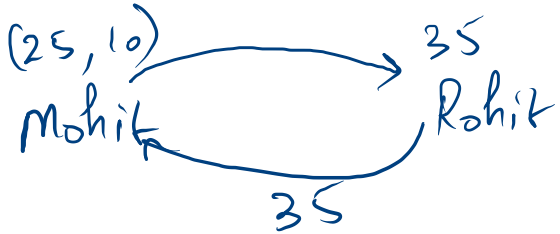
public static returntype functionName (parameters) {

}

Special function (automatically call upon code execution)

Main

Calling?



1. print
2. return

function control return

① forceful termination of code

② no more lines to execute

```java
import java.util.*;

public class Main {
    public static void add1(int n1,int n2){
        int ans = n1+n2;
        System.out.println(ans);
    }
    public static int add2(int n1,int n2){
        int ans = n1+n2;
        return ans;
    }
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int num1 = scn.nextInt();
        int num2 = scn.nextInt();

        add1(num1,num2);
        add2(num1,num2);
    }

}
```

Blank

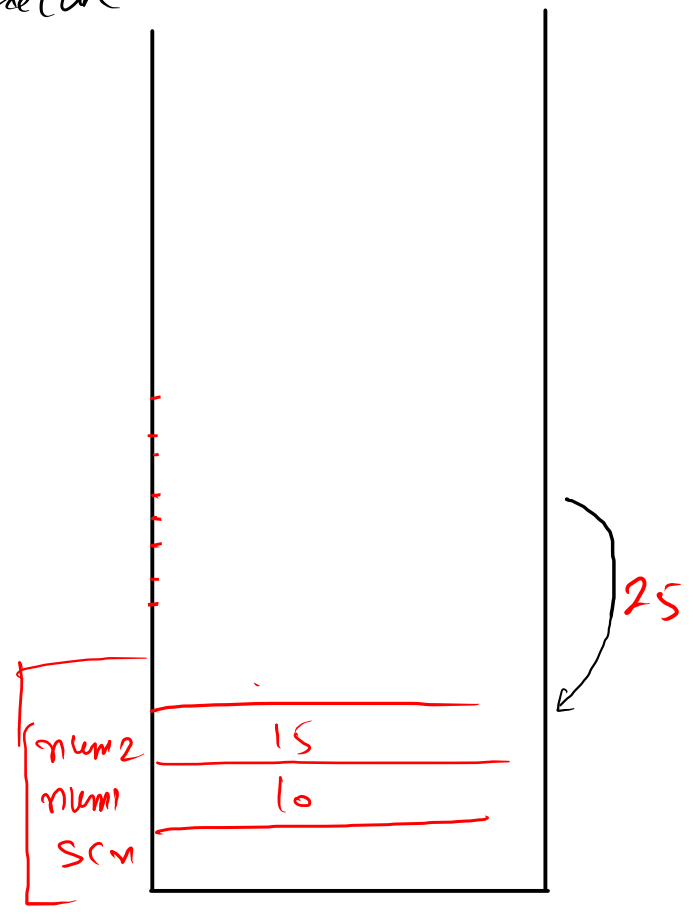
Code Execute

→ Program stack / call stack

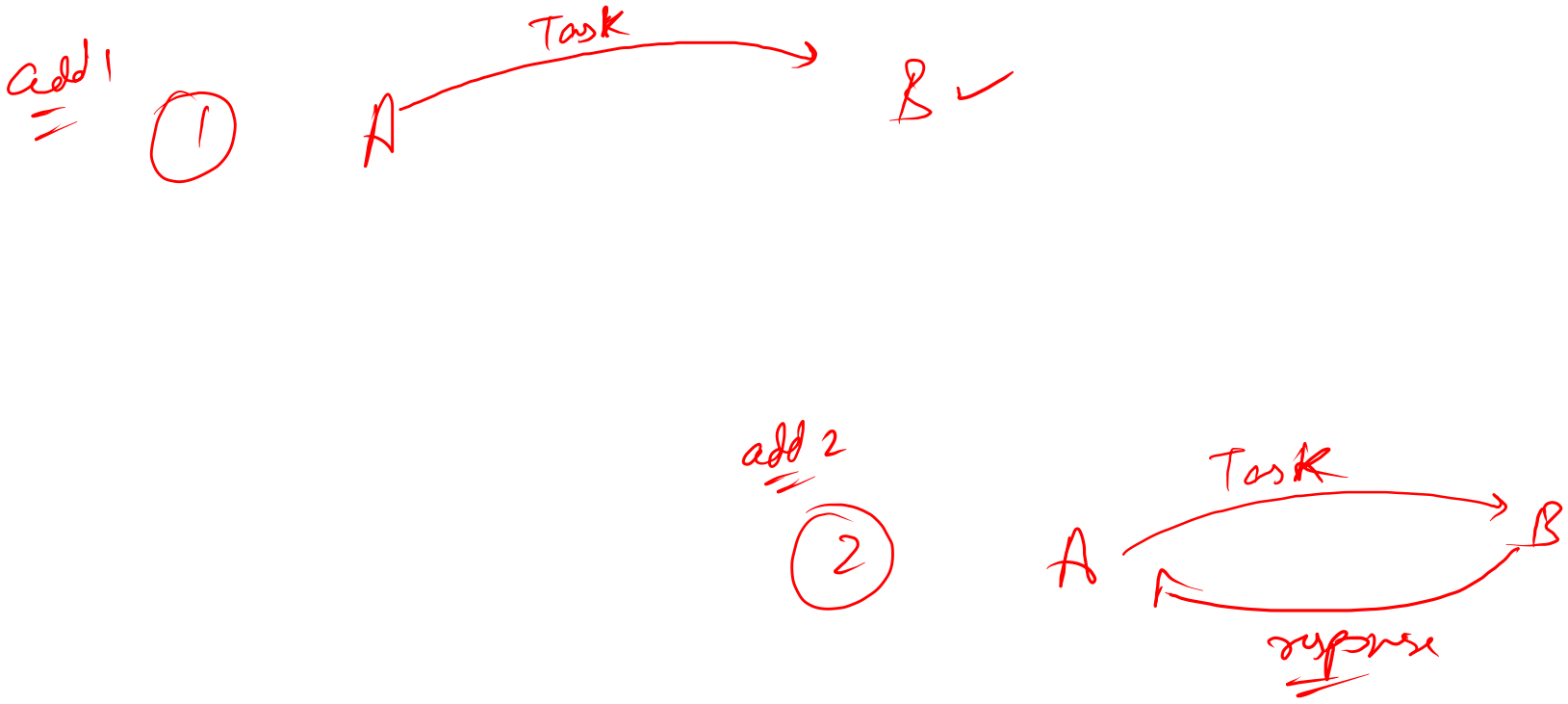
→ Heap

→ 25

Call stack



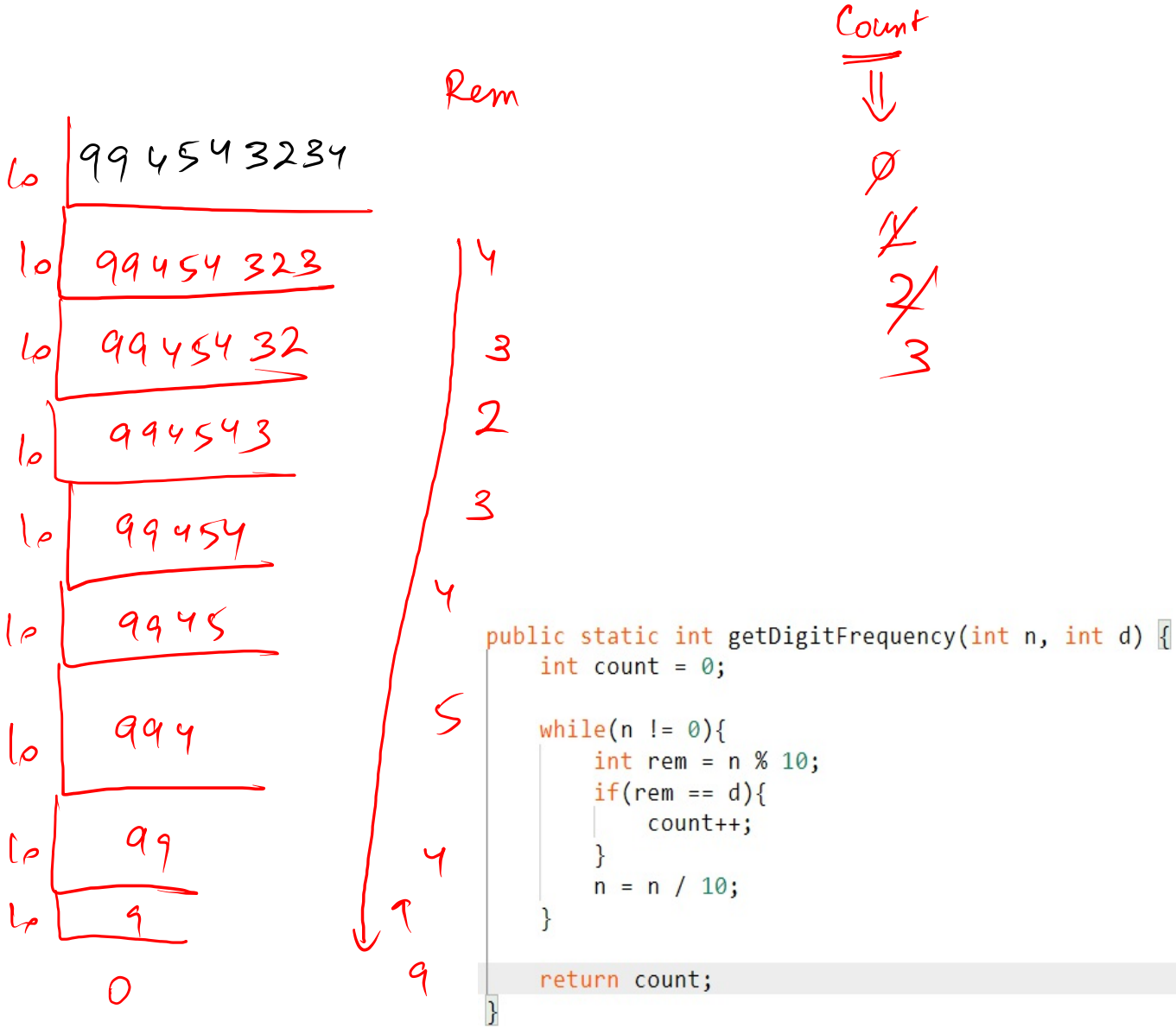
Heap


add 1
1
Task
A
B
add 2
2
Task
A
B
response

```
994543234
 4
```

Sample Output

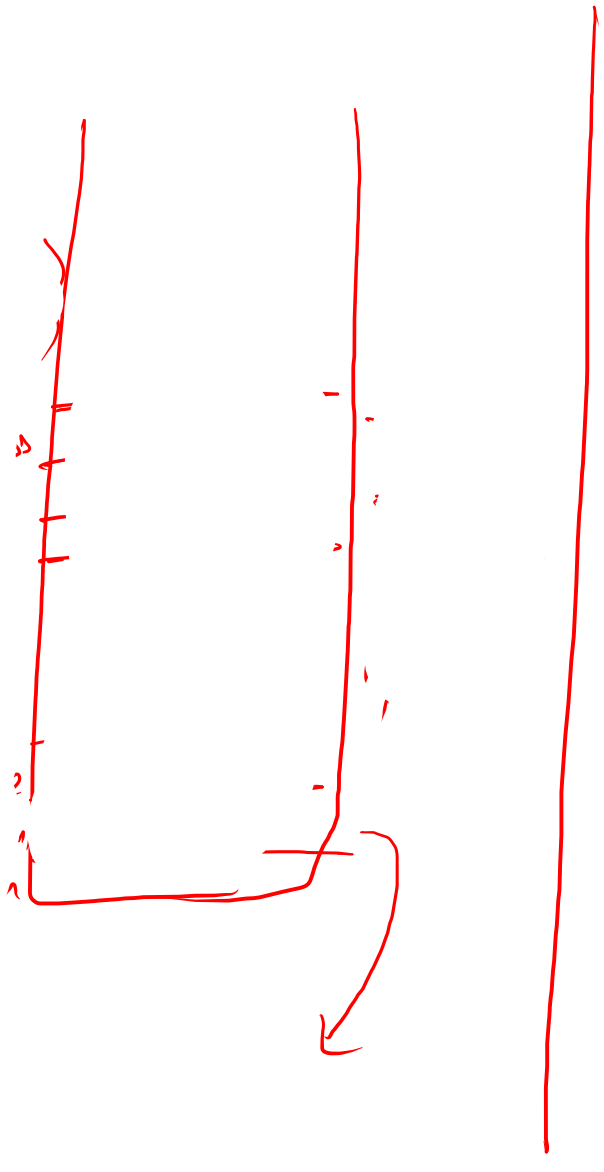
```
3
```




```java
public static int getDigitFrequency(int n, int d) {
    int count = 0;

    while(n != 0){
        int rem = n % 10;
        if(rem == d){
            count++;
        }
        n = n / 10;
    }

    return count;
}
```

```java
import java.util.*;

public class Main {
    public static void add1(int n1,int n2){
        int ans = n1+n2;
        System.out.println(ans);
    }
    public static int add2(int n1,int n2){
        int ans = n1+n2;
        return ans;
    }
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int num1 = scn.nextInt();
        int num2 = scn.nextInt();

        add1(num1,num2);
        System.out.println(add2(num1,num2));
    }

}
```

→25
→25



Data Structure

1) Array => Contig. storage of same type

Syntax

```
datatype var_name[];
```

```
var_name = new datatype[length];
```

Example

```
int arr[];
arr = new int[100];
```

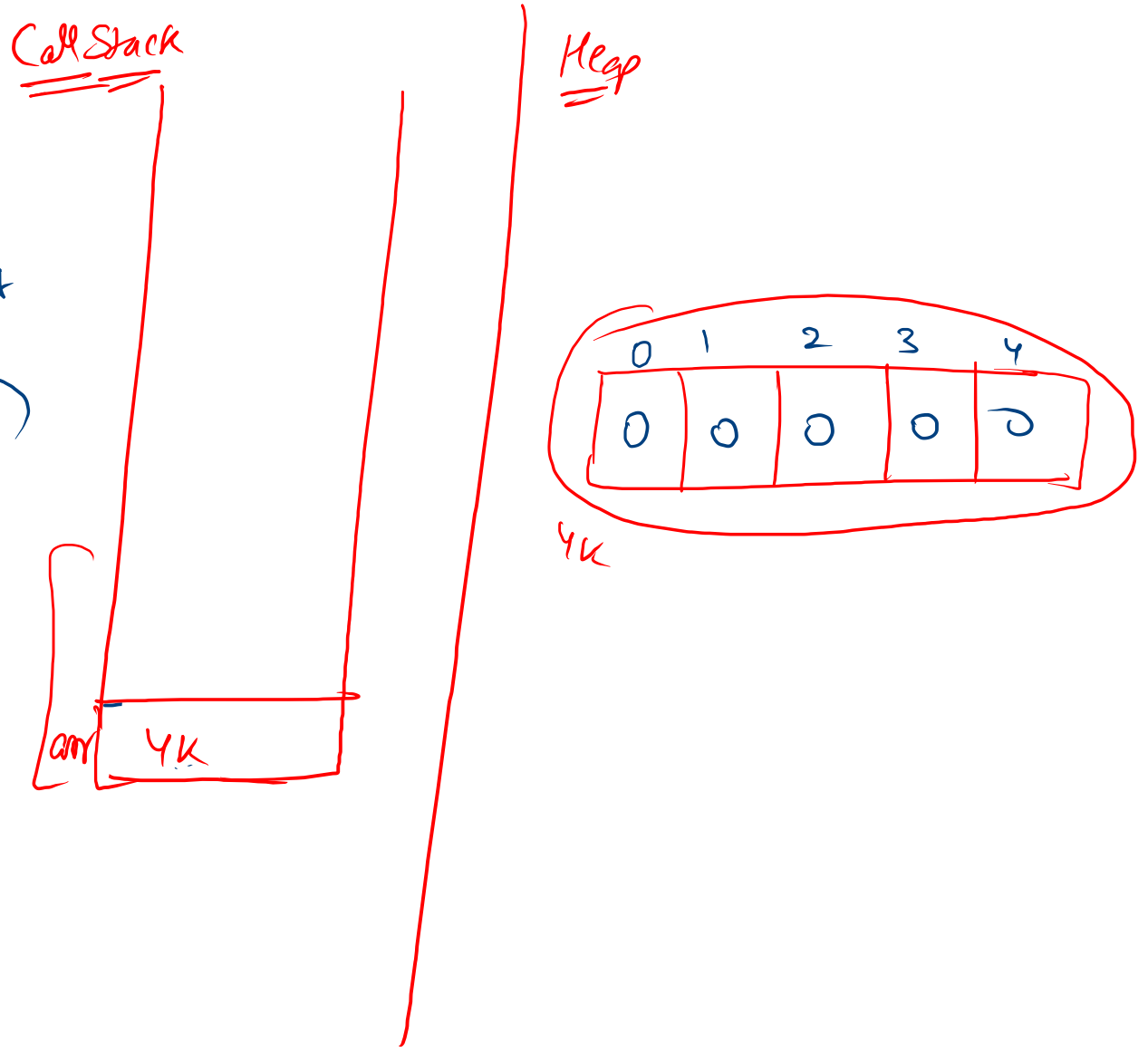
```
int arr[] = new int[100];
```

```java
import java.util.*;

public class Main {

    public static void main(String[] args) {
        int arr[] = new int[5];
    }
}
```

array → index

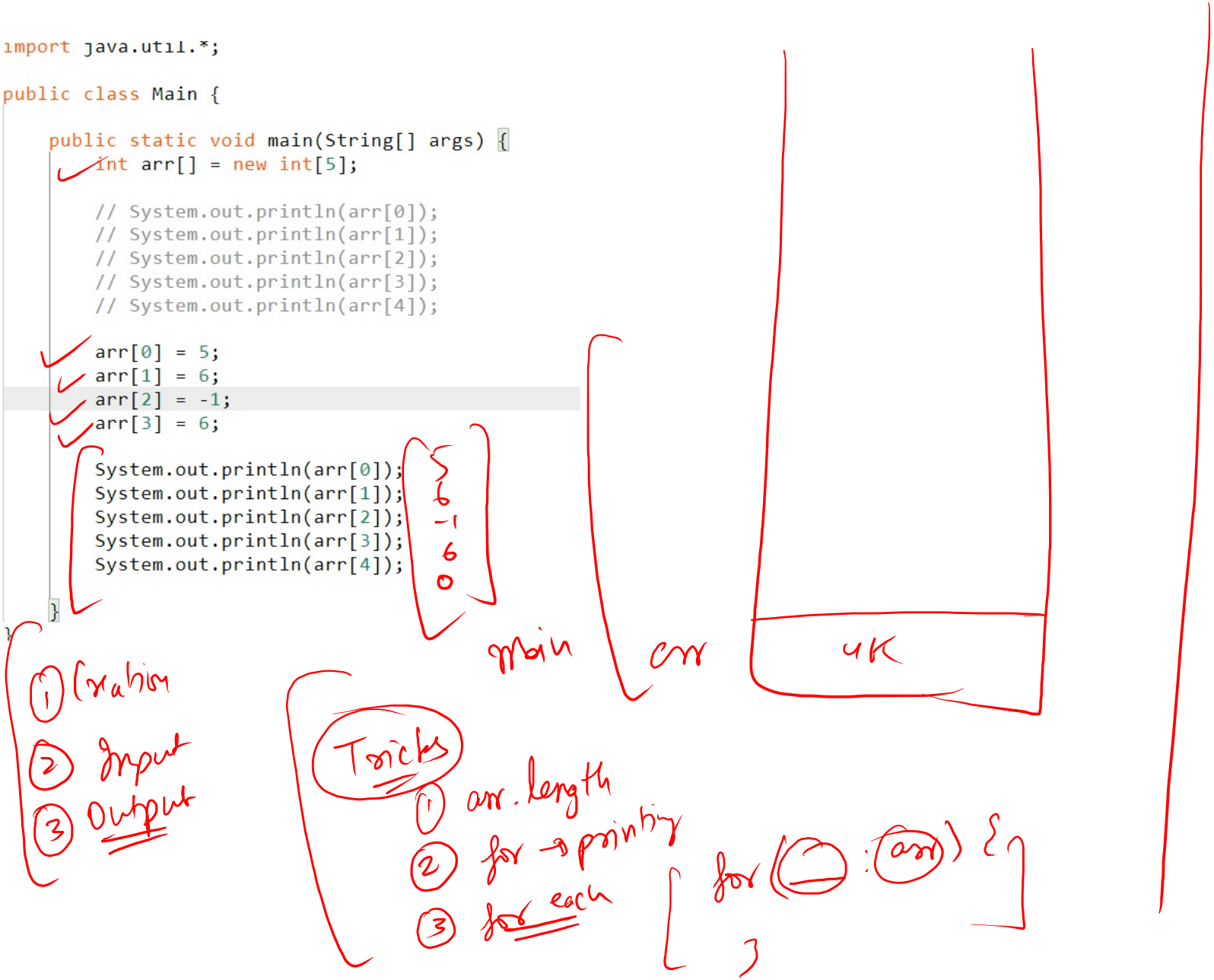
int → default value (0)

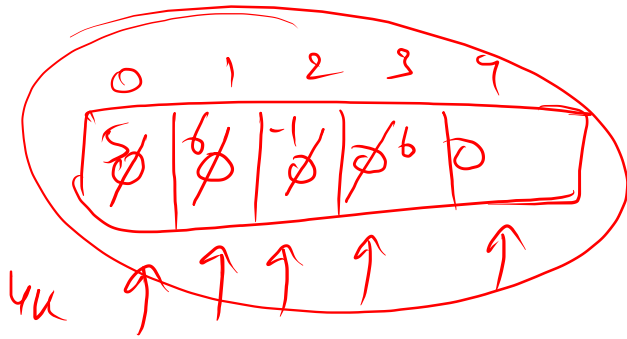
new → keyword



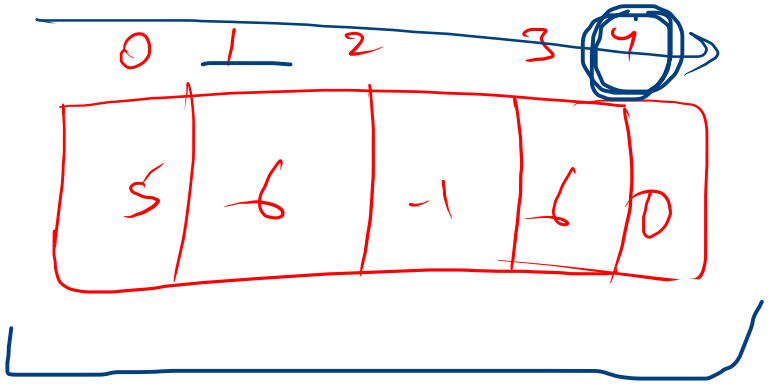
```java
import java.util.*;

public class Main {

    public static void main(String[] args) {
        int arr[] = new int[5];

        // System.out.println(arr[0]);
        // System.out.println(arr[1]);
        // System.out.println(arr[2]);
        // System.out.println(arr[3]);
        // System.out.println(arr[4]);

        arr[0] = 5;
        arr[1] = 6;
        arr[2] = -1;
        arr[3] = 6;

        System.out.println(arr[0]);
        System.out.println(arr[1]);
        System.out.println(arr[2]);
        System.out.println(arr[3]);
        System.out.println(arr[4]);

    }
}
```

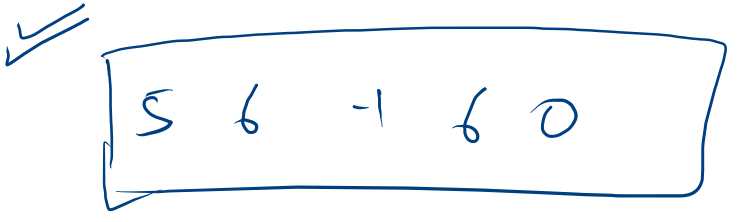
5
6
-1
6
0





```java
public static void main(String[] args) {
    int arr[] = new int[5];

    // System.out.println(arr[0]);
    // System.out.println(arr[1]);
    // System.out.println(arr[2]);
    // System.out.println(arr[3]);
    // System.out.println(arr[4]);

    arr[0] = 5;
    arr[1] = 6;
    arr[2] = -1;
    arr[3] = 6;

    // System.out.println(arr[0]);
    // System.out.println(arr[1]);
    // System.out.println(arr[2]);
    // System.out.println(arr[3]);
    // System.out.println(arr[4]);

    // System.out.println("length :"+arr.length);

    // for(int idx = 0 ; idx <= arr.length-1 ; idx++){
    //     int val = arr[idx];
    //     System.out.print(val+" ");
    // }
    // System.out.println();


    for(int val : arr){
        System.out.print(val+" ");
    }
    System.out.println();
}
```



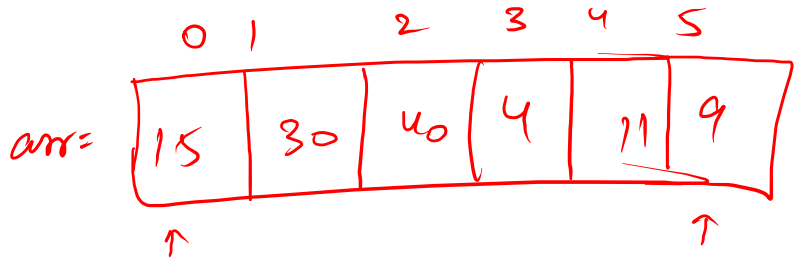


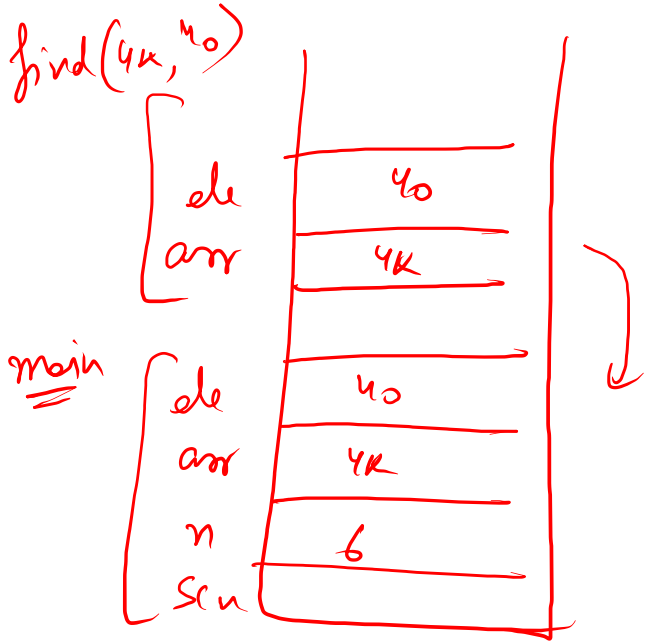
```
6
15
30
40
4
11
9
40
```

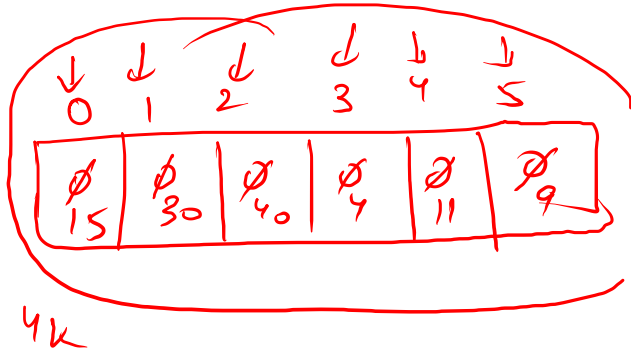
Sample Output

```
2
```

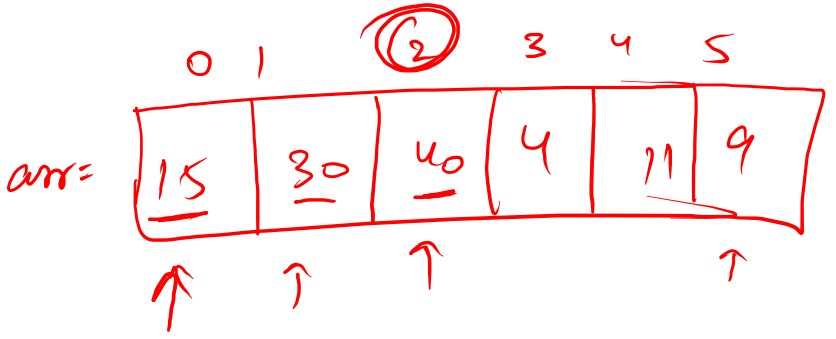
```java
public static void main(String[] args) throws Exception {
    Scanner scn = new Scanner(System.in);

    int n = scn.nextInt(); // len
    int arr[] = new int[n]; // arr
    for(int idx = 0 ; idx <= n-1 ; idx++){
        arr[idx] = scn.nextInt();
    }
    int ele = scn.nextInt(); // element

    int res = find(arr,ele);
    System.out.println(res);
}
public static int find(int arr[],int ele){
    // logic
}
```





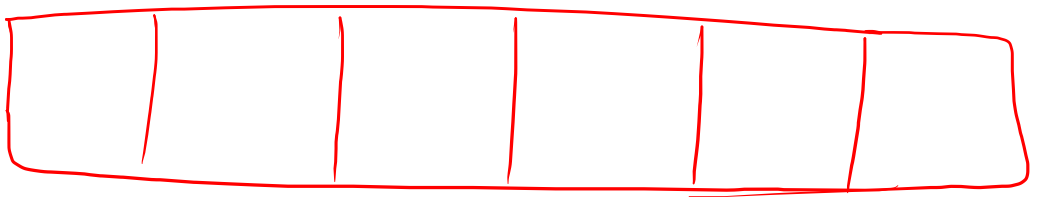


ele = 40



idx = 0, 1, 2, 3, 7, 5

```
public static int find(int arr[],int ele){
    for(int idx = 0 ; idx <= arr.length-1 ; idx++){
        if(arr[idx] == ele){
            return idx; // element found
        }
    }
    return -1; // element not found
}
```



```java
public static int find(int arr[],int ele){
    for(int idx = 0 ; idx <= arr.length-1 ; idx++){
        if(arr[idx] == ele){
            return idx; // element found
        }
    }
    return -1; // element not found
}
```

H.W → find

3. You are required to find the span of input. Span is defined as difference of maximum value and minimum value.

6
15
30
40
4
11
9

arr =

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

Span = Max − Min

int max = 15 30 40

for (idx: 1 → len-1)

if (arr[idx] > max)
    max = arr[idx]

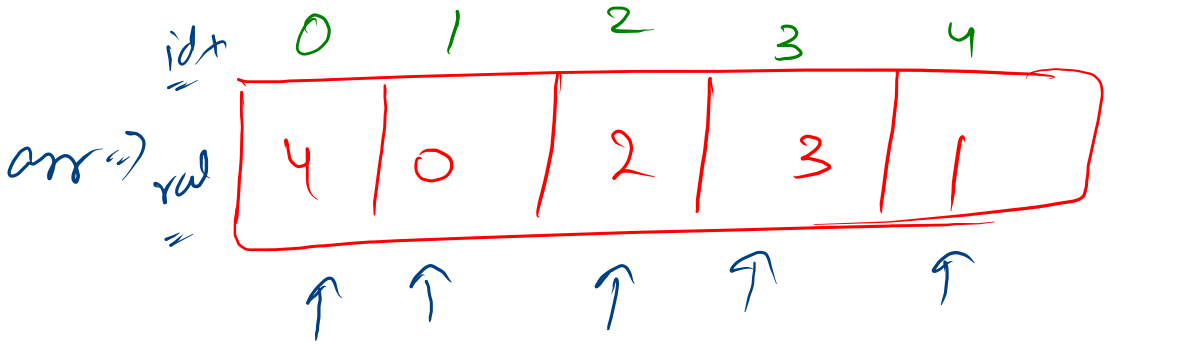
int min = 15 4

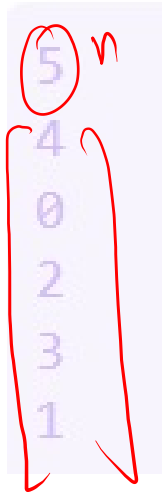
if(arr[idx] < min) {
    min = arr[idx];
}

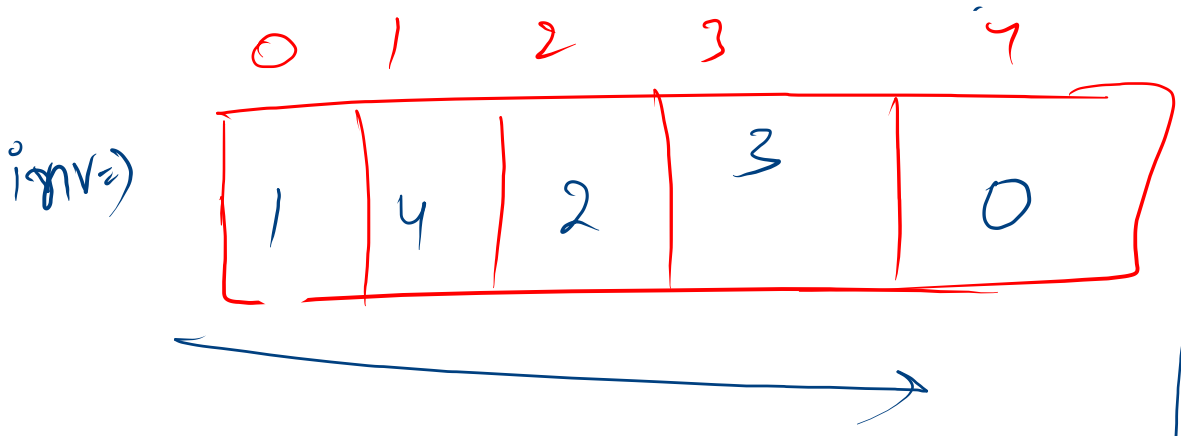
return max - min;

36

idx, val








```
inv[]
for (idx=0 → 4) {
    int val = arr[idx];

    inv[val] = idx;

}
```