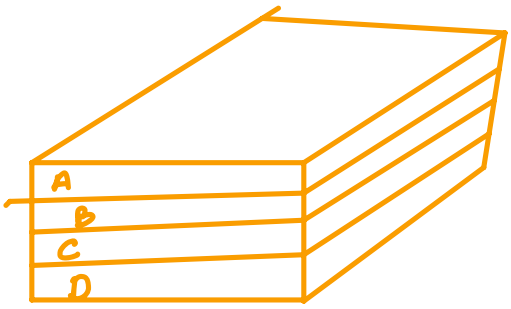
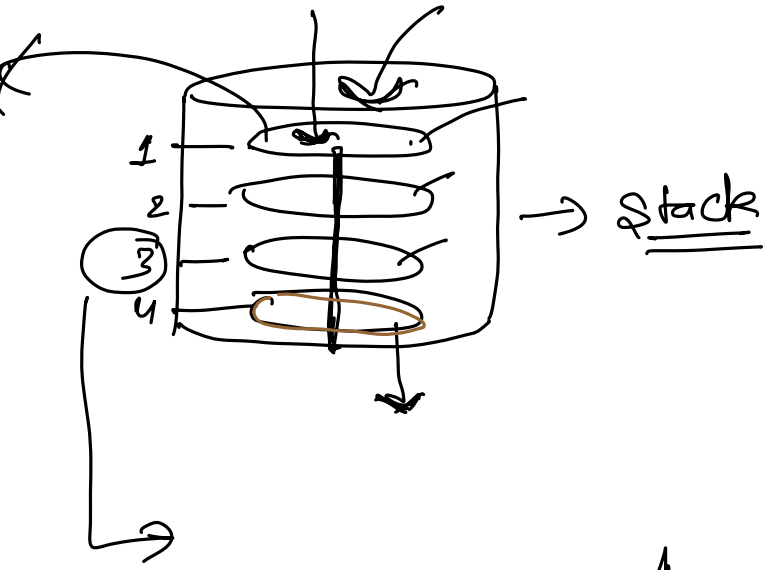
Stack



→ Stack of books



→ Stack

(1). we can only insert or remove from top.

(2) we only have access to the top.

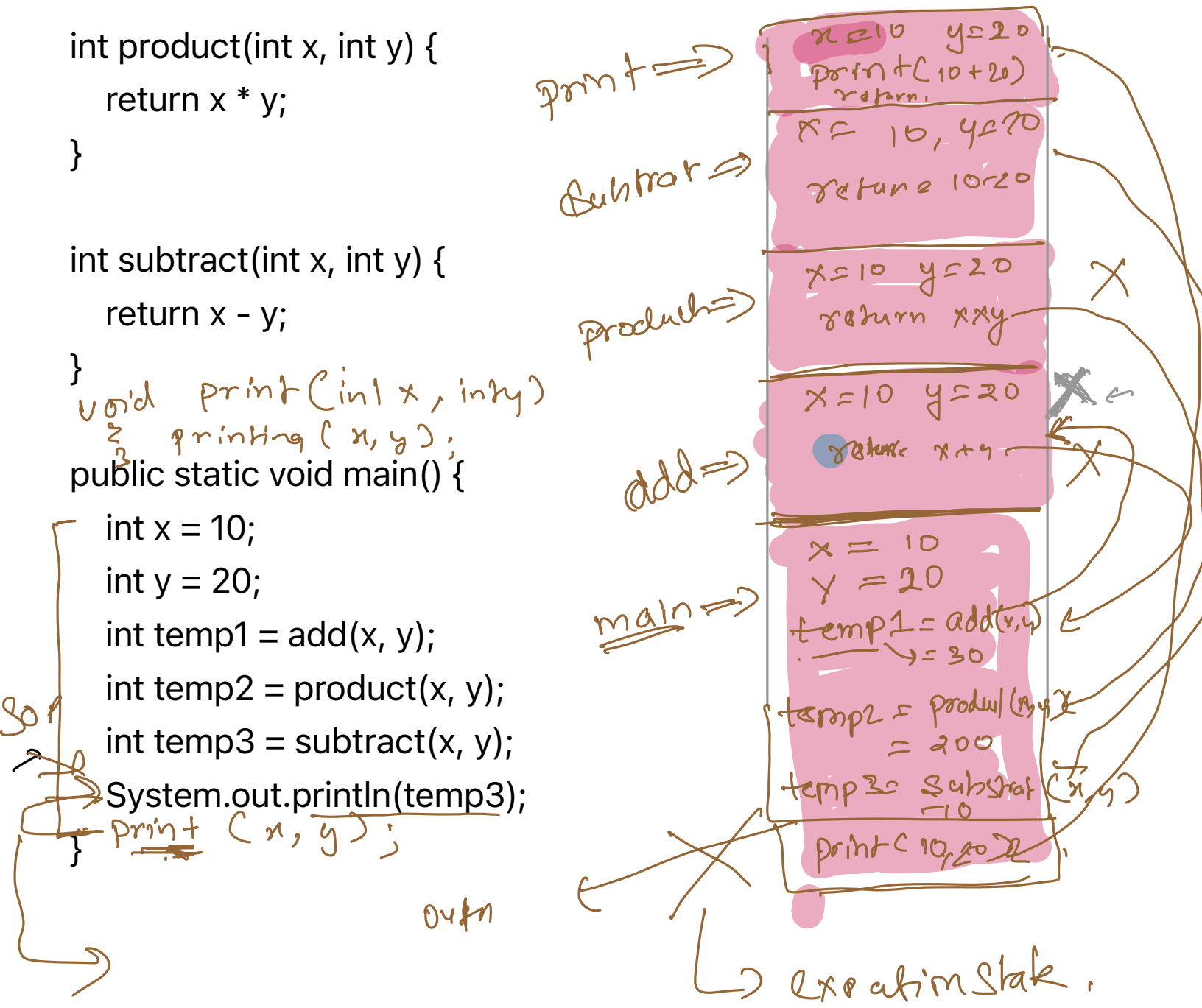
Third ← peek.

Remove 2 → pop

Remove 1 → pop

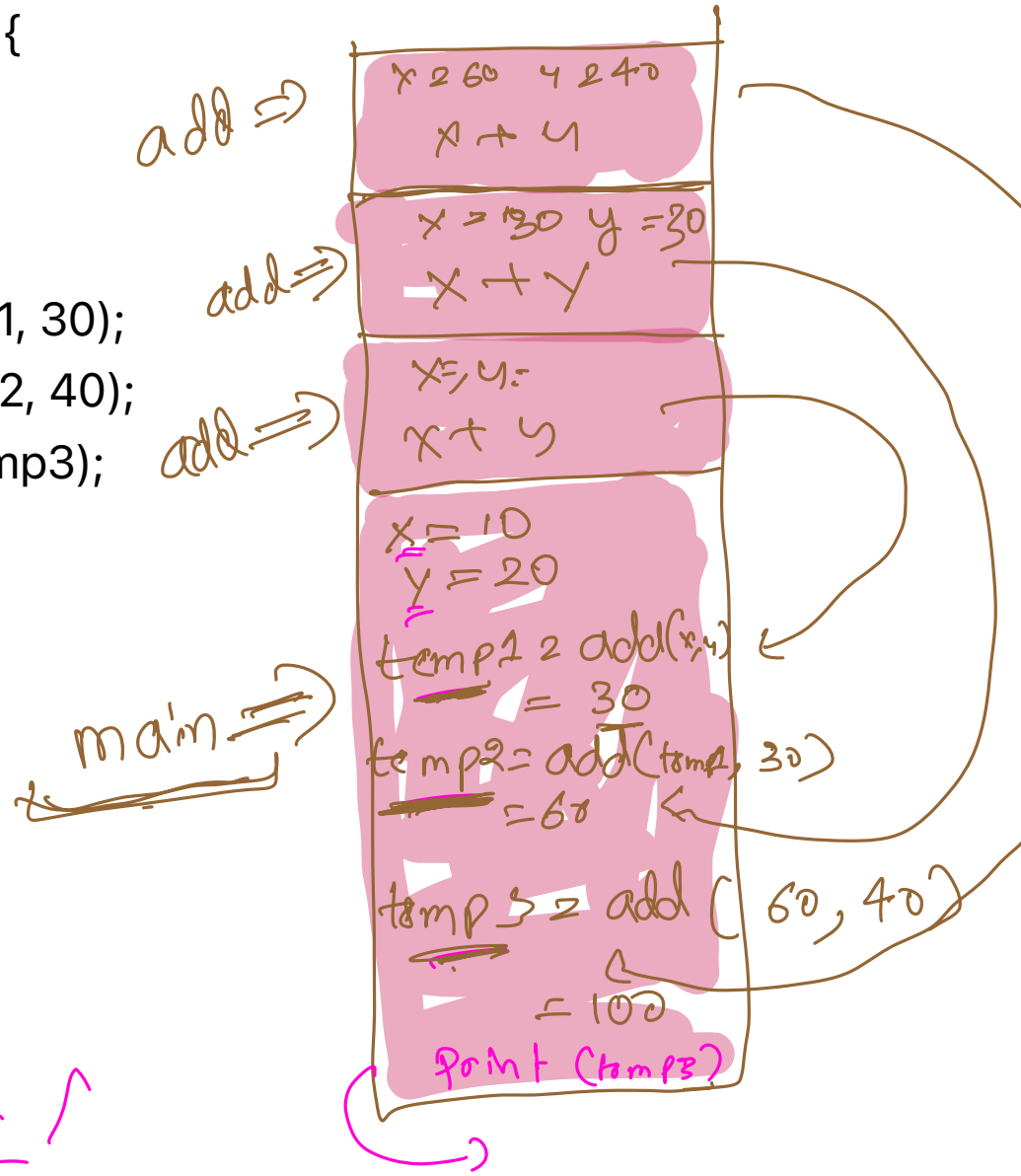
Q1 What is the output of the below code.

```
int add(int x, int y) {
    return x + y;
}

int product(int x, int y) {
    return x * y;
}

int subtract(int x, int y) {
    return x - y;
}
void print(int x, int y)
{ printing(x, y);
}
public static void main() {
    int x = 10;
    int y = 20;
    int temp1 = add(x, y);
    int temp2 = product(x, y);
    int temp3 = subtract(x, y);
    System.out.println(temp3);
    print(x, y);
}
```



Qus 2

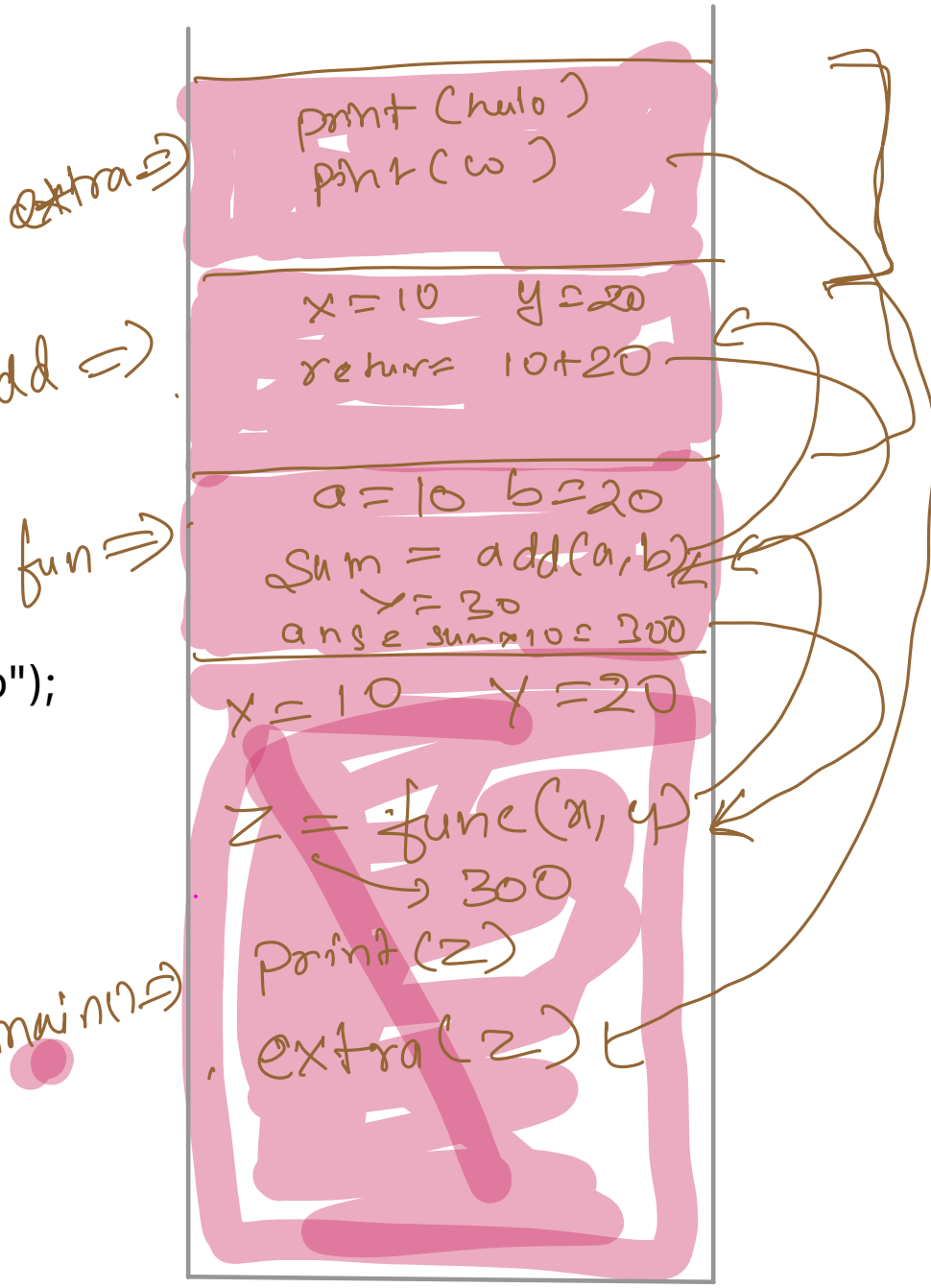
```
int add(int x, int y) {
    return x + y;
}

public static void main() {
    int x = 10;
    int y = 20;
    int temp1 = add(x, y);
    int temp2 = add(temp1, 30);
    int temp3 = add(temp2, 40);
    System.out.println(temp3);
}
```



output
= 100

Qus 3

```
int add(int x, int y) {
    return x + y;
}

static int fun(int a, int b) {
    int sum = add(a, b);
    int ans = sum * 10;
    return ans;
}
static void extra(int w){
    System.out.println("Hello");
    System.out.println(w);
}
public static void main() {
    int x = 10;
    int y = 20;
    int z = fun(x, y);
    System.out.println(z);
    extra(z);
}
```



Output

300
Hello
w

Types of memory

int, float, double, bool.

reference.

1.) Stack : All the primitive data type and references will be stored in Stack

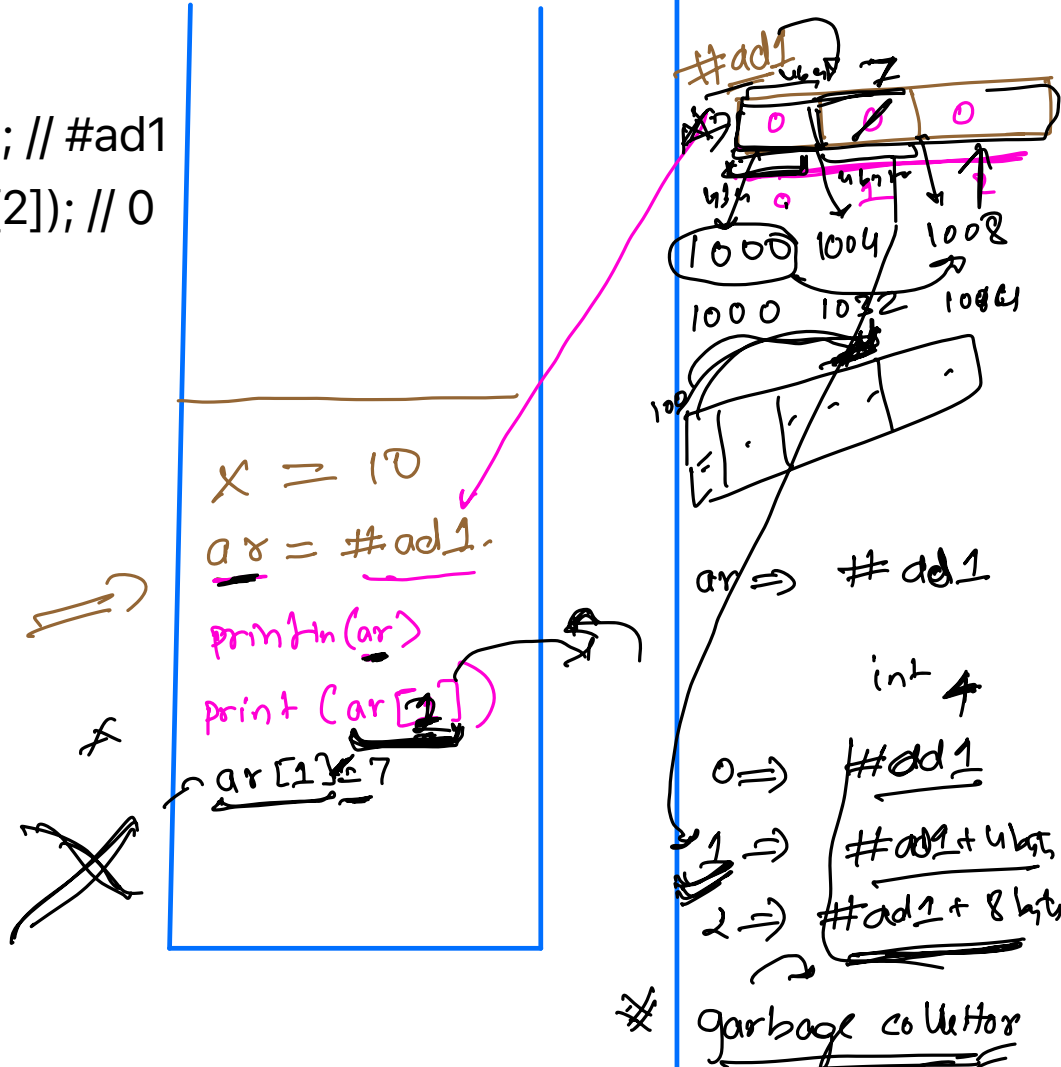
2) Heap : Container of that reference is stored in heap. Arrays, Arraylist, Objects are created inside heap.
↳ anything which we declare with new goes in heap.

Example 1

```
public static void main() {
    int x = 10;
    int[] ar = new int[3];
    System.out.println(ar); // #ad1
    System.out.println(ar[2]); // 0
    ar[1] = 7;
}
```

Stack

Heap.



Output

#ad1

0

main ⟹

10: 24 pm

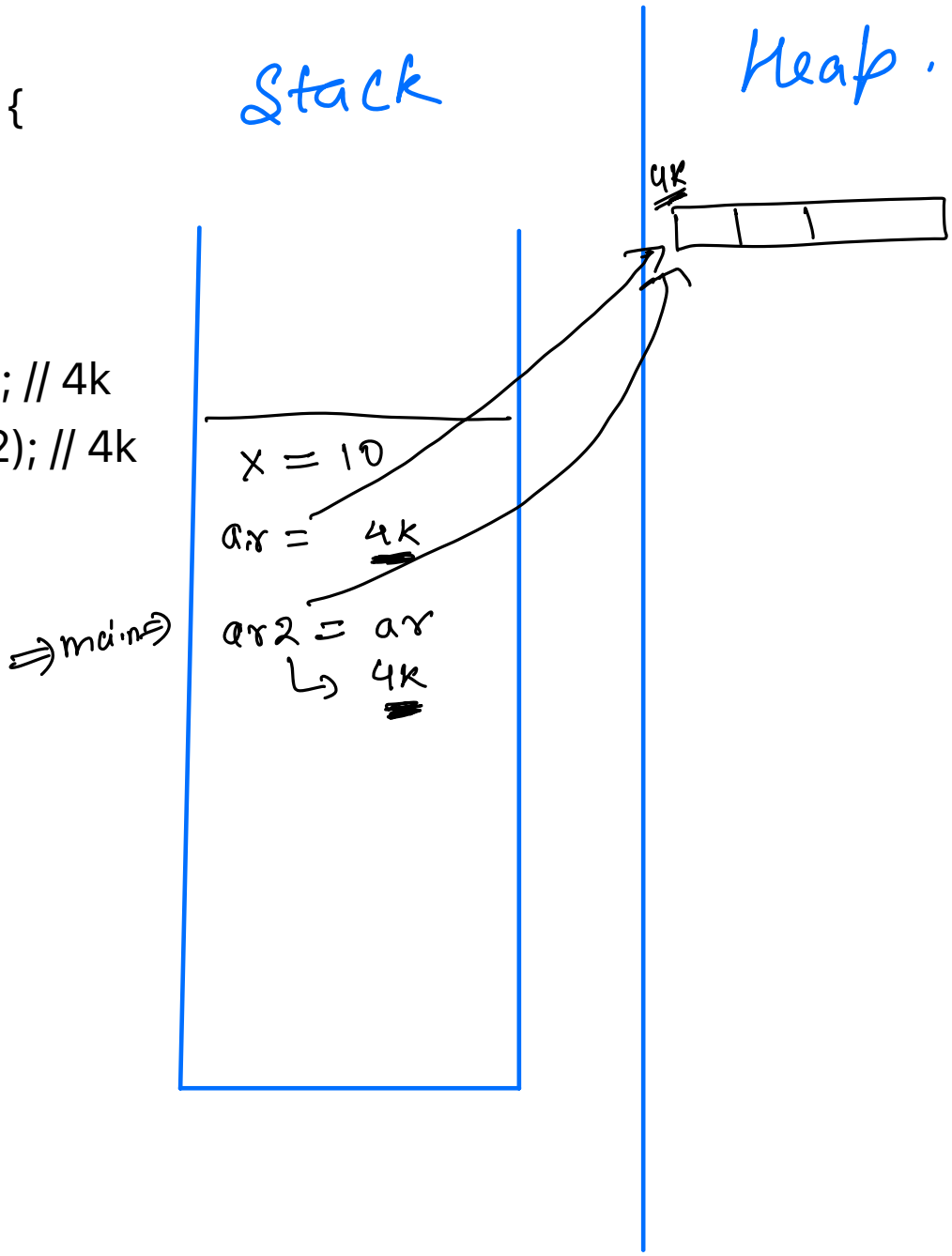
Example 2

```
public static void main() {
    int x = 10;
    int[] ar = new int[3];
    int[] ar2 = ar;
    System.out.println(ar); // 4k
    System.out.println(ar2); // 4k
}
```

Output

4K

4K



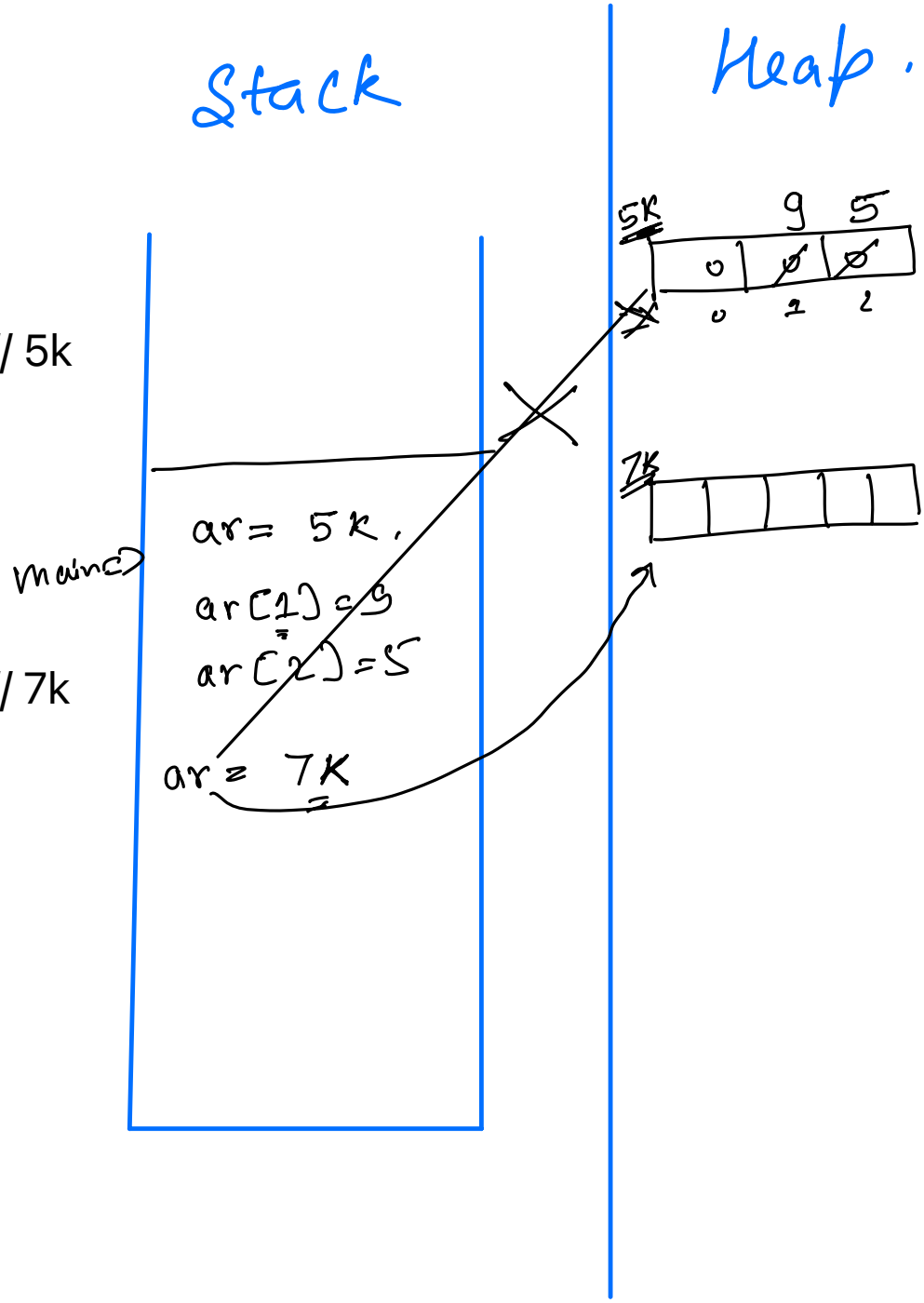
Example 3

```
public static void main() {
    int[] ar = new int[3];
    System.out.println(ar); // 5k
    ar[1] = 9;
    ar[2] = 5;

    ar = new int[5];
    System.out.println(ar); // 7k
}
```

Output

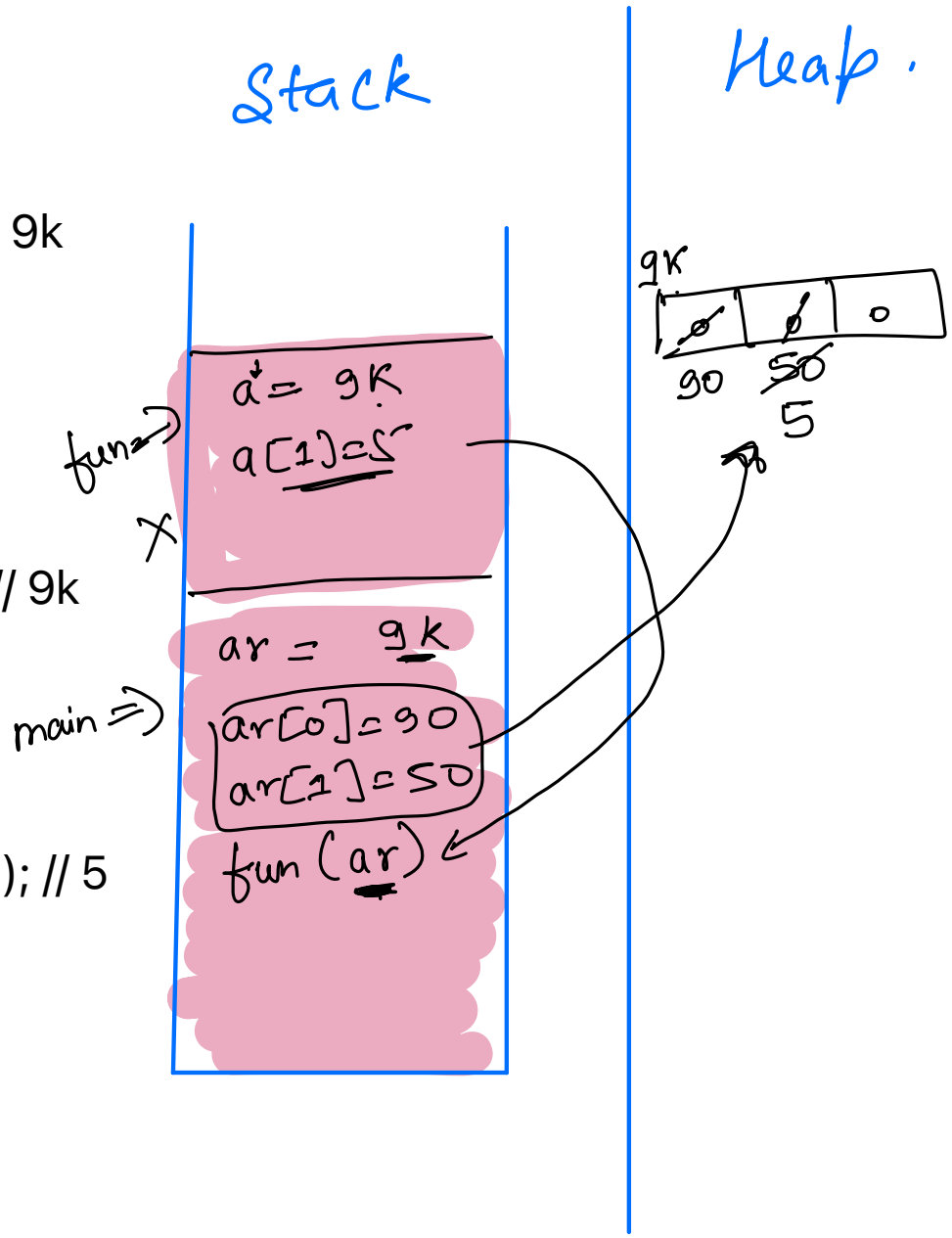
5K

7K



Example 4

```
static void fun(int[] a){
    System.out.println(a); // 9k
    a[1] = 5;
}
public static void main() {
    int[] ar = new int[3];
    System.out.println(ar); // 9k
    ar[0] = 90;
    ar[1] = 50;
    fun(ar);
    System.out.println(ar[1]); // 5
}
```



Output

9k

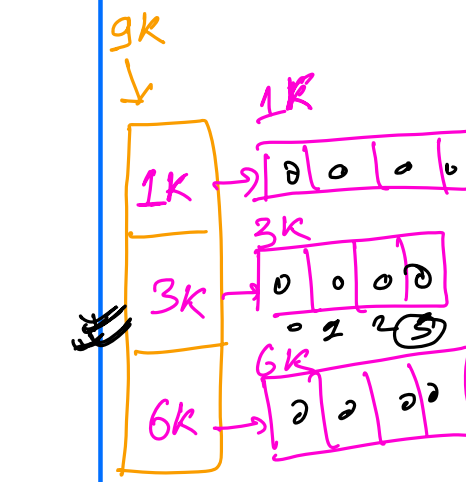
5

Example 5

```
public static void main() {
float y = 7.84f;
int[][] mat = new int[3][4];
System.out.println(mat); // 9k
System.out.println(mat[1]); // 3k
System.out.println(mat[1][3]); // 0
}
```

Stack

Heap.



main ⇒

y = 7.84f

mat = 9k

Output

9k

3k

0

mat[1] = 3k

mat[1][3]

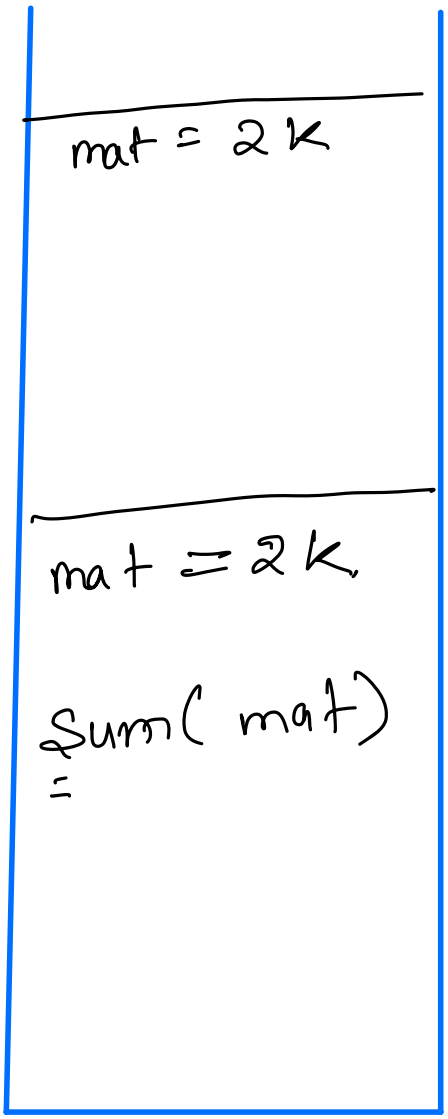
Example 6

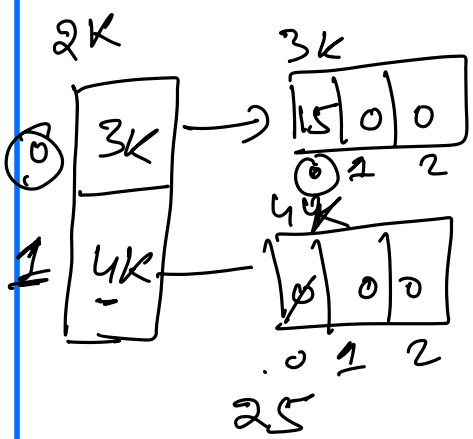
```
static void sum(int[][] mat){
System.out.println(mat); // 2k
System.out.println(mat[0][0] + mat[1][0]); // 40
}
public static void main() {
int[][] mat = new int[2][3];
mat[0][0] = 15;
mat[1][0] = 25;
sum(mat);
}
```

15 + 25

Stack

Heap.





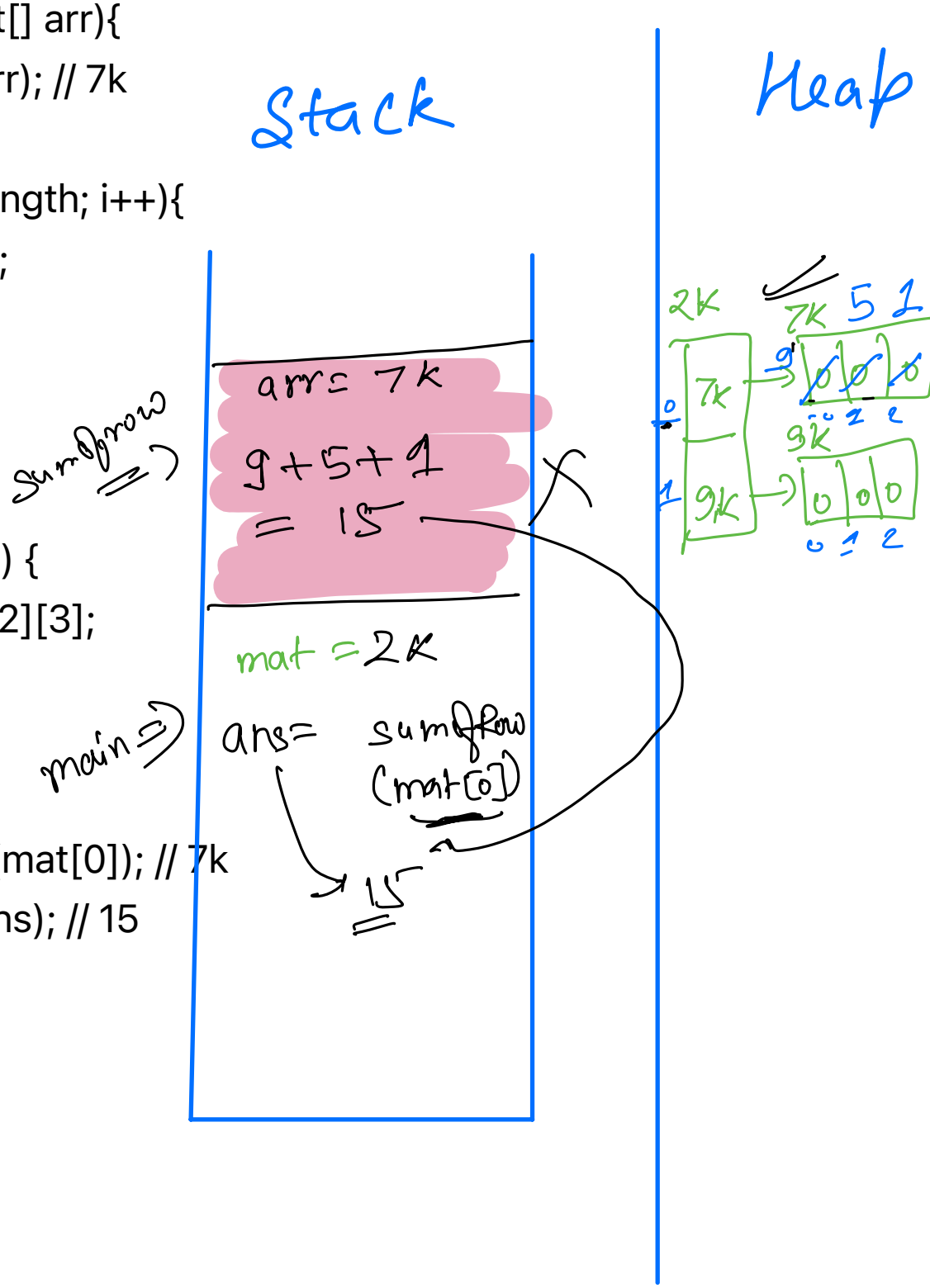
Output

2K

40

Example 7

```
static int sumOfRow(int[] arr){
    System.out.println(arr); // 7k
    int sum = 0;
    for (int i = 0; i < arr.length; i++){
        sum = sum + arr[i];
    }
    return sum;
}

public static void main() {
    int[][] mat = new int[2][3];
    mat[0][0] = 9;
    mat[0][1] = 5;
    mat[0][2] = 1;
    int ans = sumOfRow(mat[0]); // 7k
    System.out.println(ans); // 15
}
```

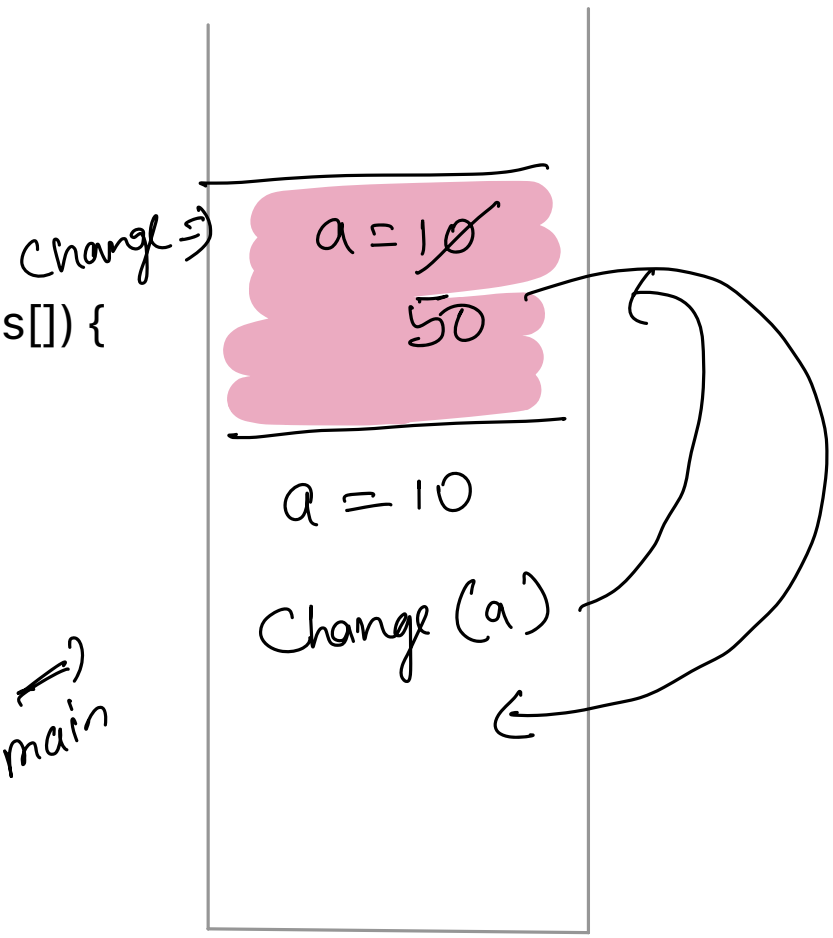


output

15

Quiz 1

```
static void change(int a) {
    a = 50;
}

public static void main(String args[]) {
int a = 10;
change(a);
System.out.println(a);
}
```

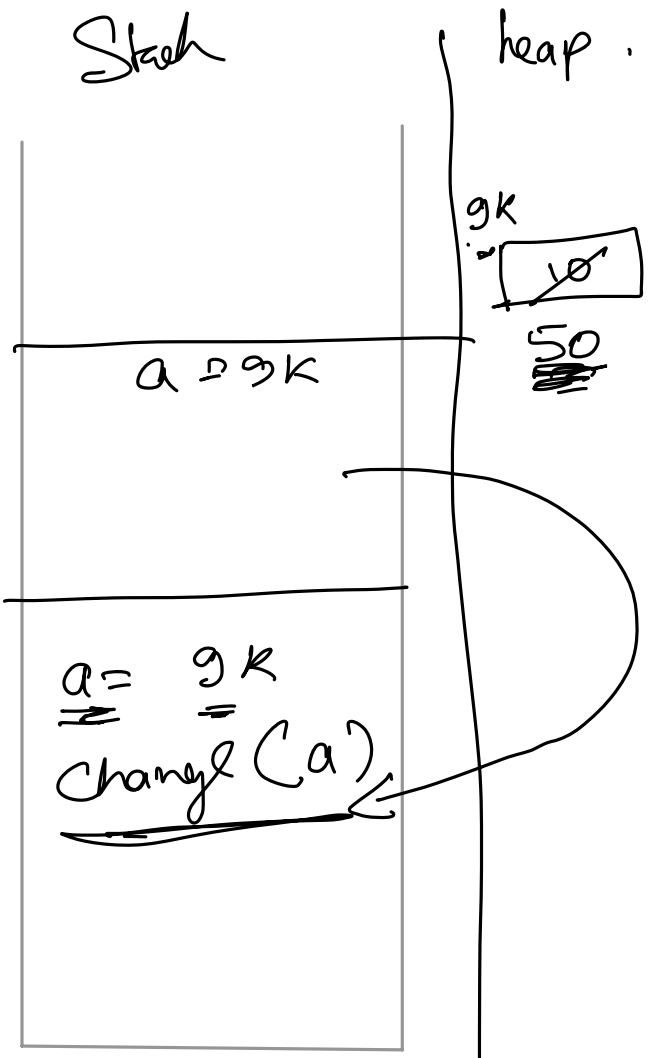


Output

10

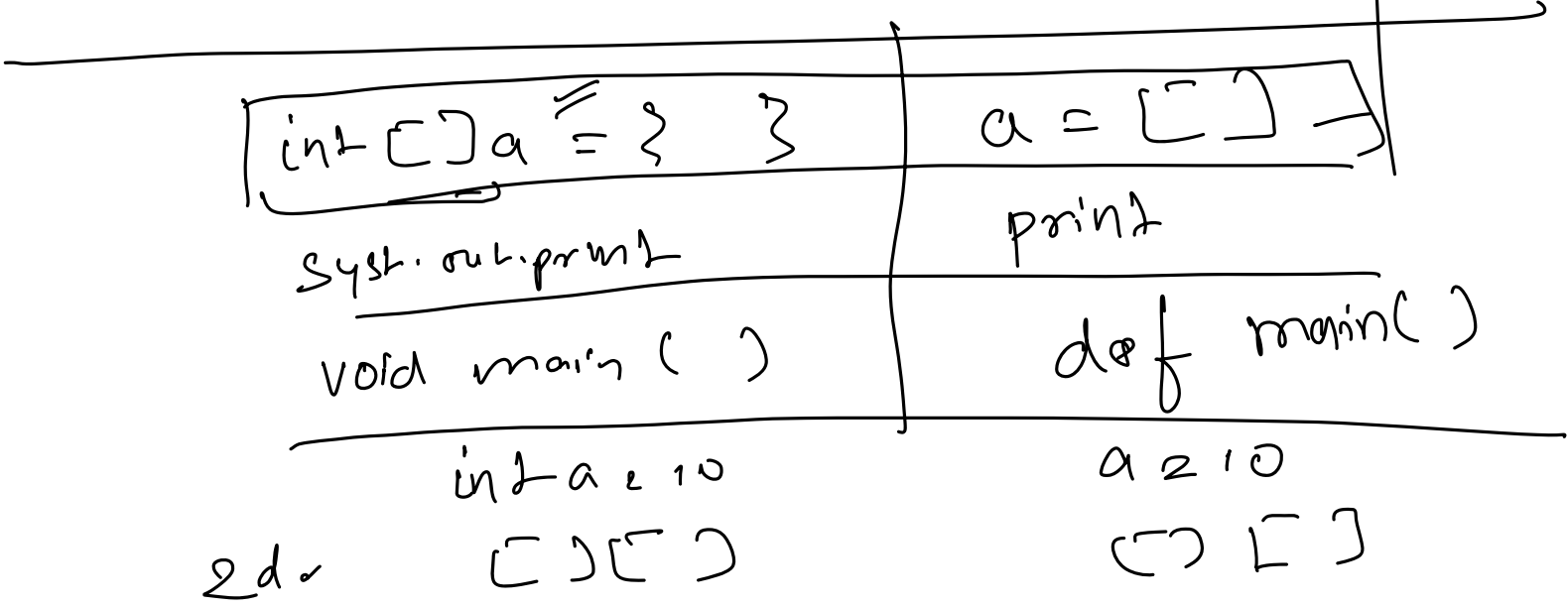
Quiz 2

```
static void change(int[]a) {
    a[0] = 50;
}

public static void main(String args[]) {
    int[]a = {10};
    change(a);
    System.out.println(a[0]);
}
```



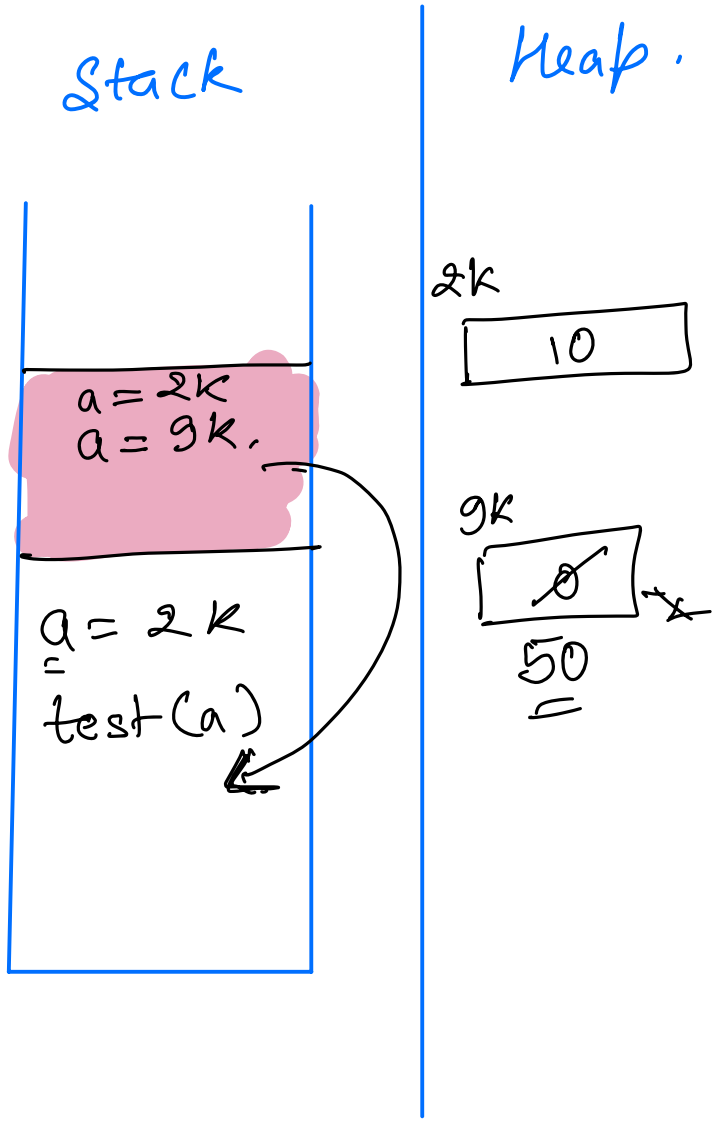
Output

50



Quiz 2

```
static void test(int[]a) {
    a = new int[1];
    a[0] = 50;
}

public static void main(String args[]) {
    int[]a = {10};
    test(a);
    System.out.println(a[0]);
}
```



test ⇒

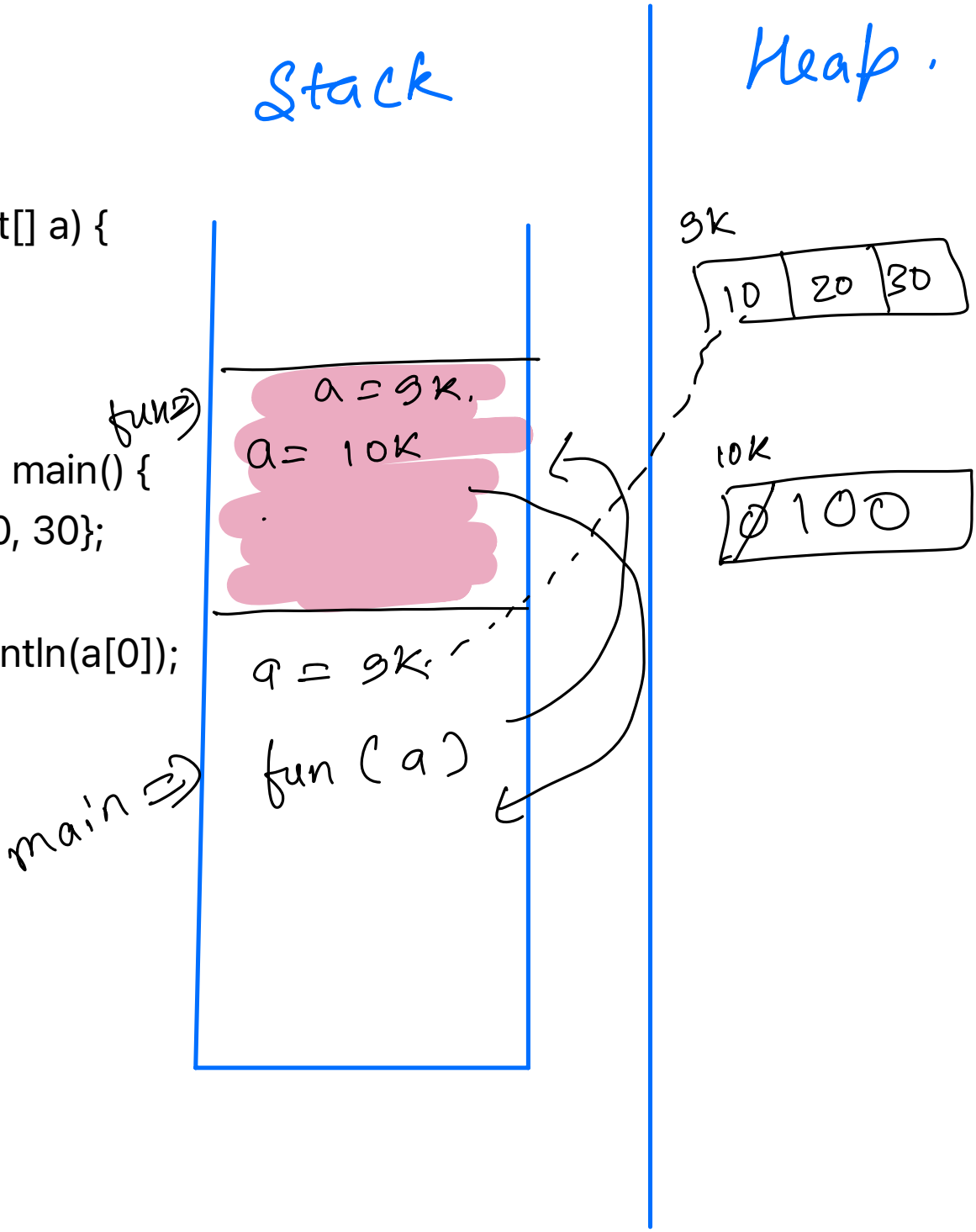
main ⇒

output a

10

Quiz 4

```
static void fun(int[] a) {
    a = new int[1];
a[0] = 100;
}
public static void main() {
 int[] a = {10, 20, 30};
fun(a);
System.out.println(a[0]);
}
```

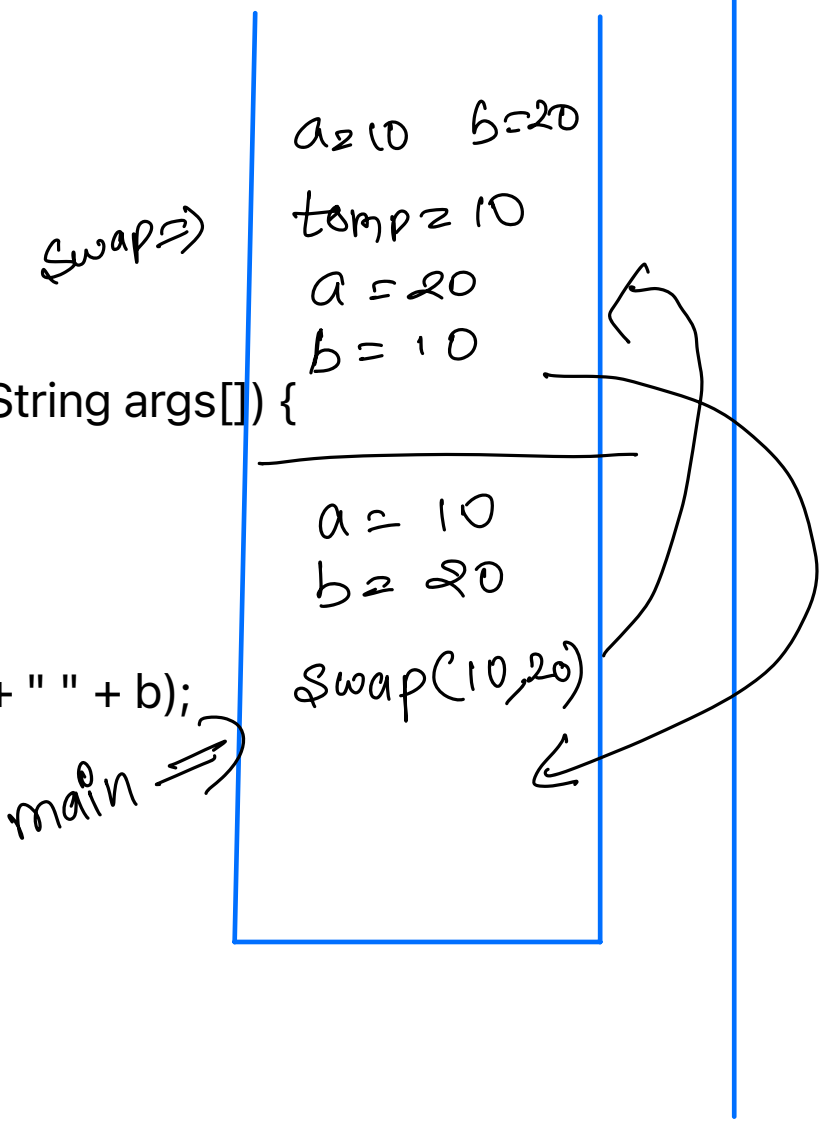


Output

10

Quiz

```
static void swap(int a,int b) {
    int temp = a;
    a = b;
    b = temp;
}

public static void main(String args[]) {
    int a = 10;
    int b = 20;
    swap(a,b);
    System.out.println(a + " " + b);
}
```

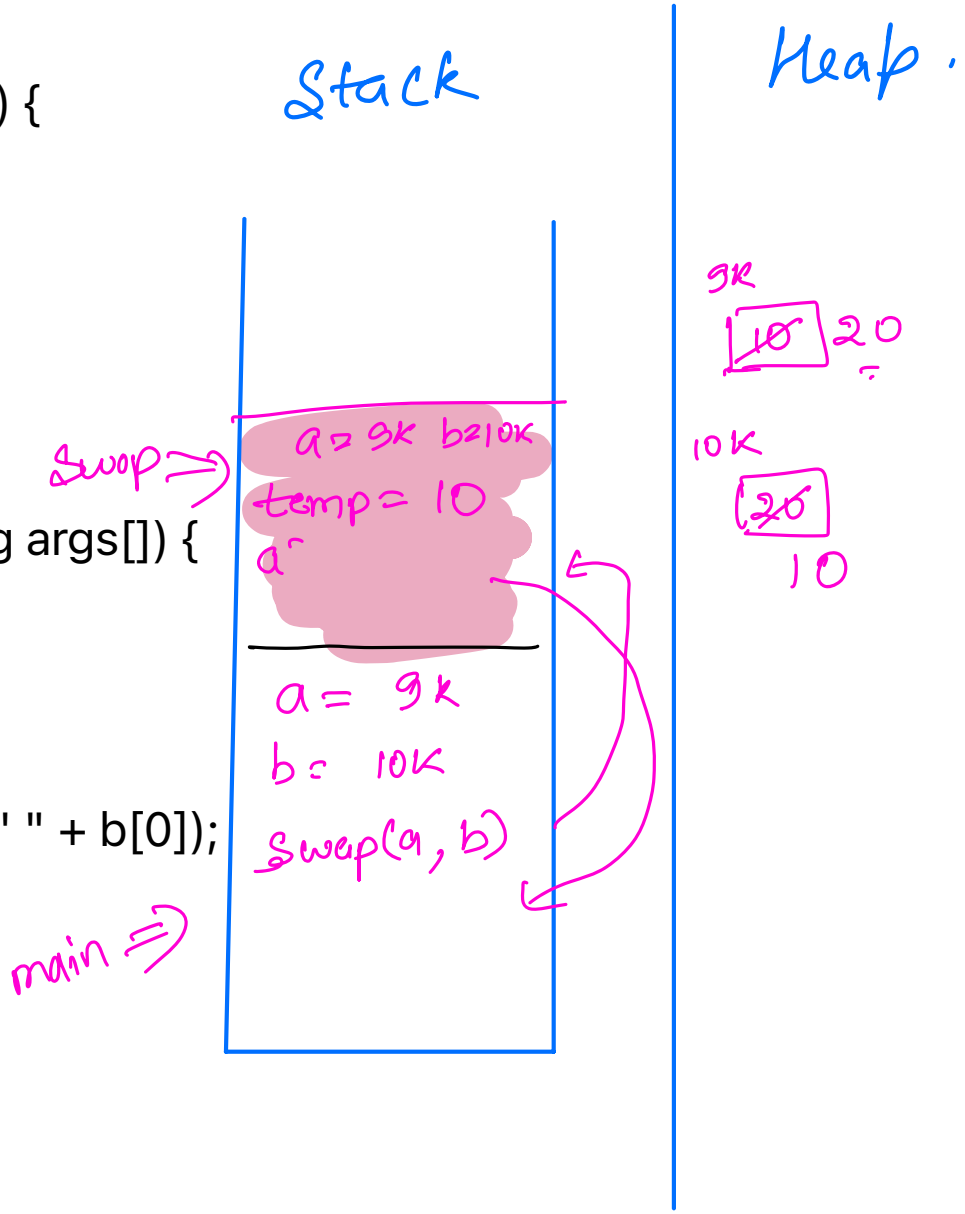


Output

10 20

Quiz6

```
static void swap(int[]a,int[]b) {
    int temp = a[0];
    a[0] = b[0];
    b[0] = temp;
}

public static void main(String args[]) {
    int[]a = {10};
    int[]b = {20};
    swap(a,b);
    System.out.println(a[0] + " " + b[0]);
}
```

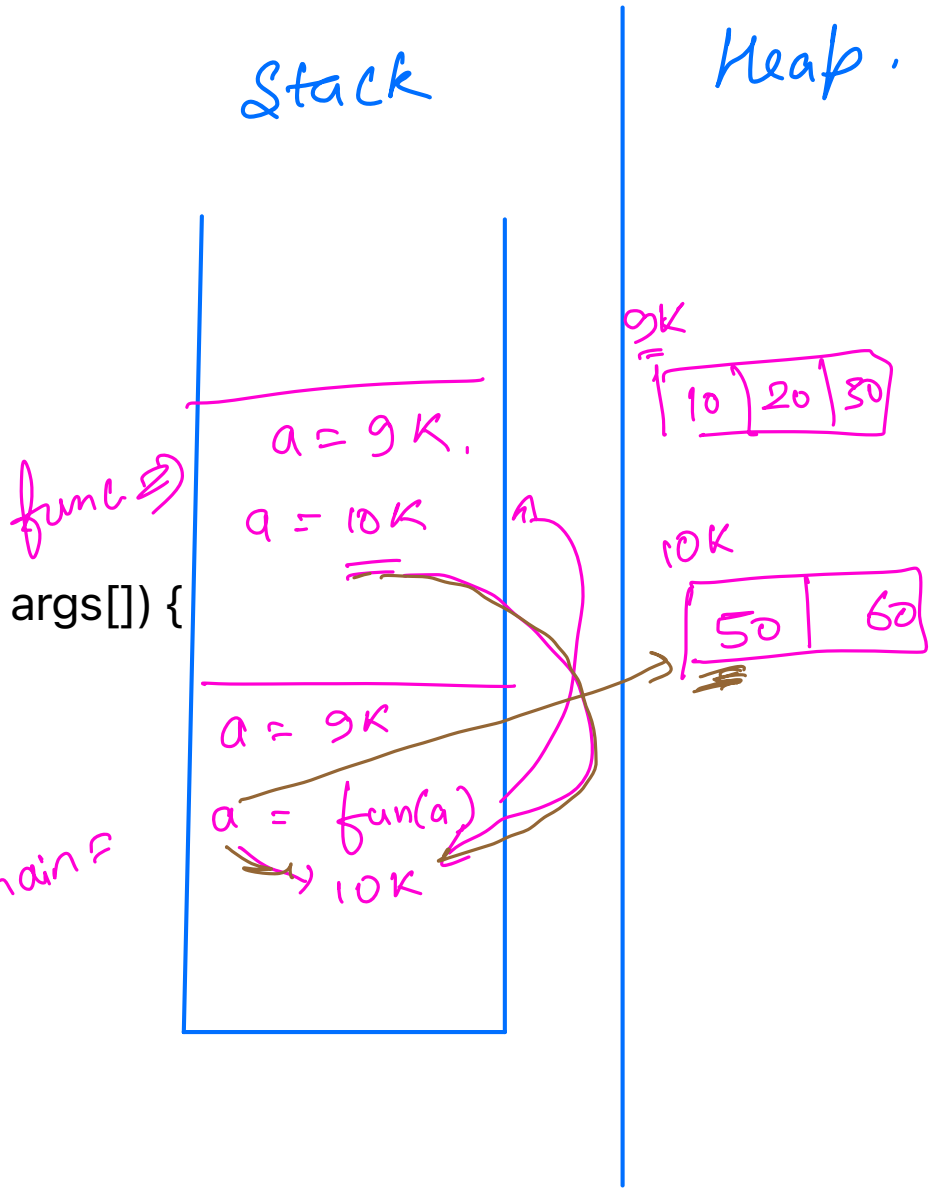


Output

20 10

Quiz 7

```
static int[] fun(int[]a) {
    a = new int[2];
    a[0] = 50; a[1] = 60;
    return a;
}

public static void main(String args[]) {
    int[]a = {10,20,30};
    a = fun(a);
    System.out.println(a[0]);
}
```



Output

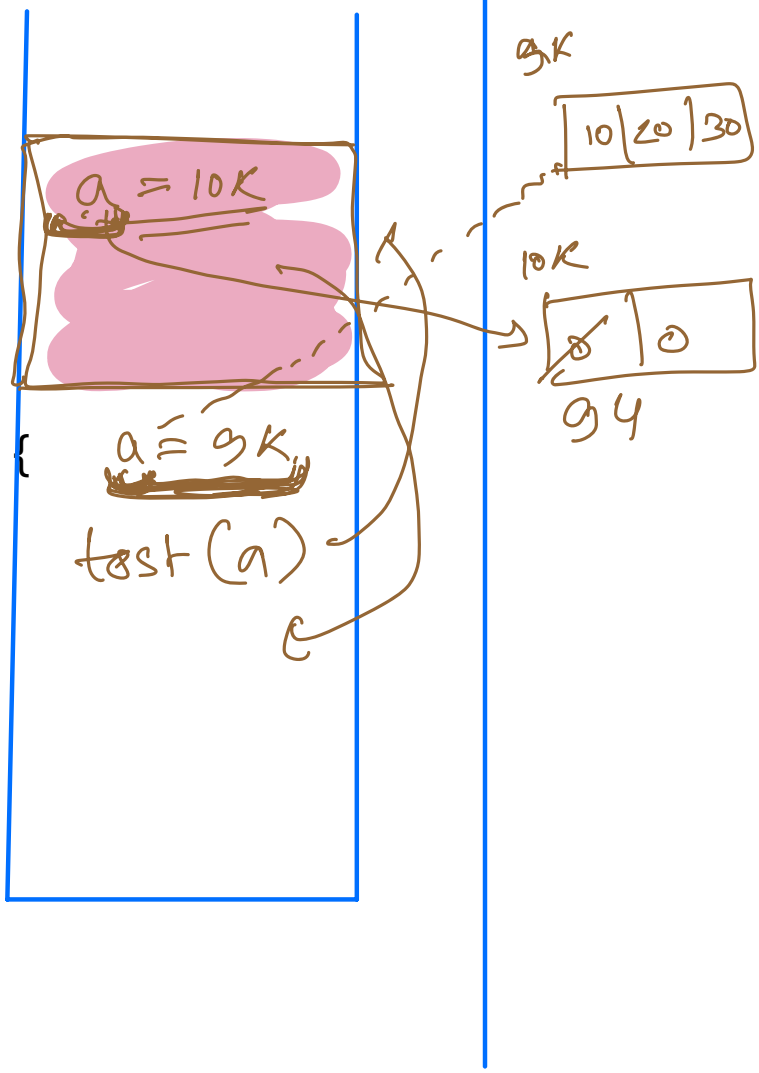
50

Quiz 8

```
static void test(int[]a) {
    a = new int[2];
    a[0] = 94;
    return a;
}

public static void main(String args[]) {
    int[]a = {10,20,30};
    test(a);
    System.out.println(a[0]);
}
```

test

main



output

10