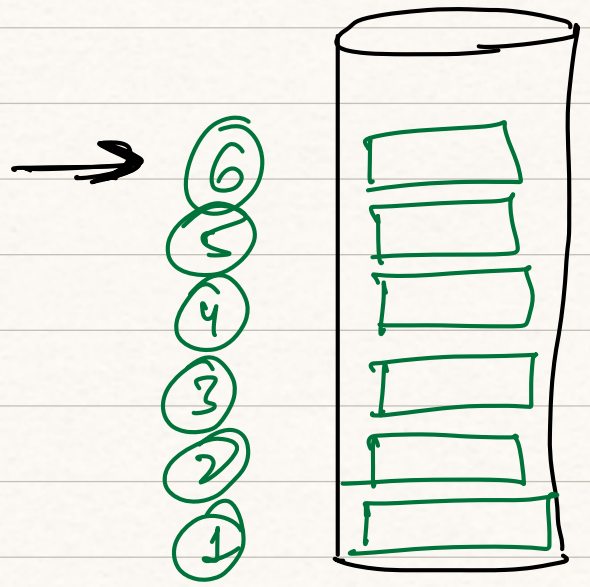
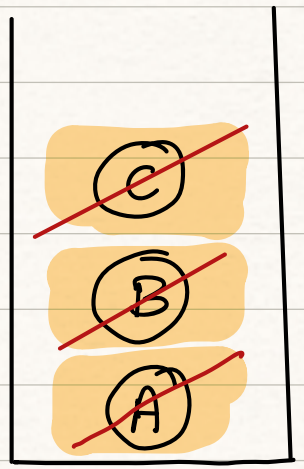
Stack →

push

remove





push A
push B
push C

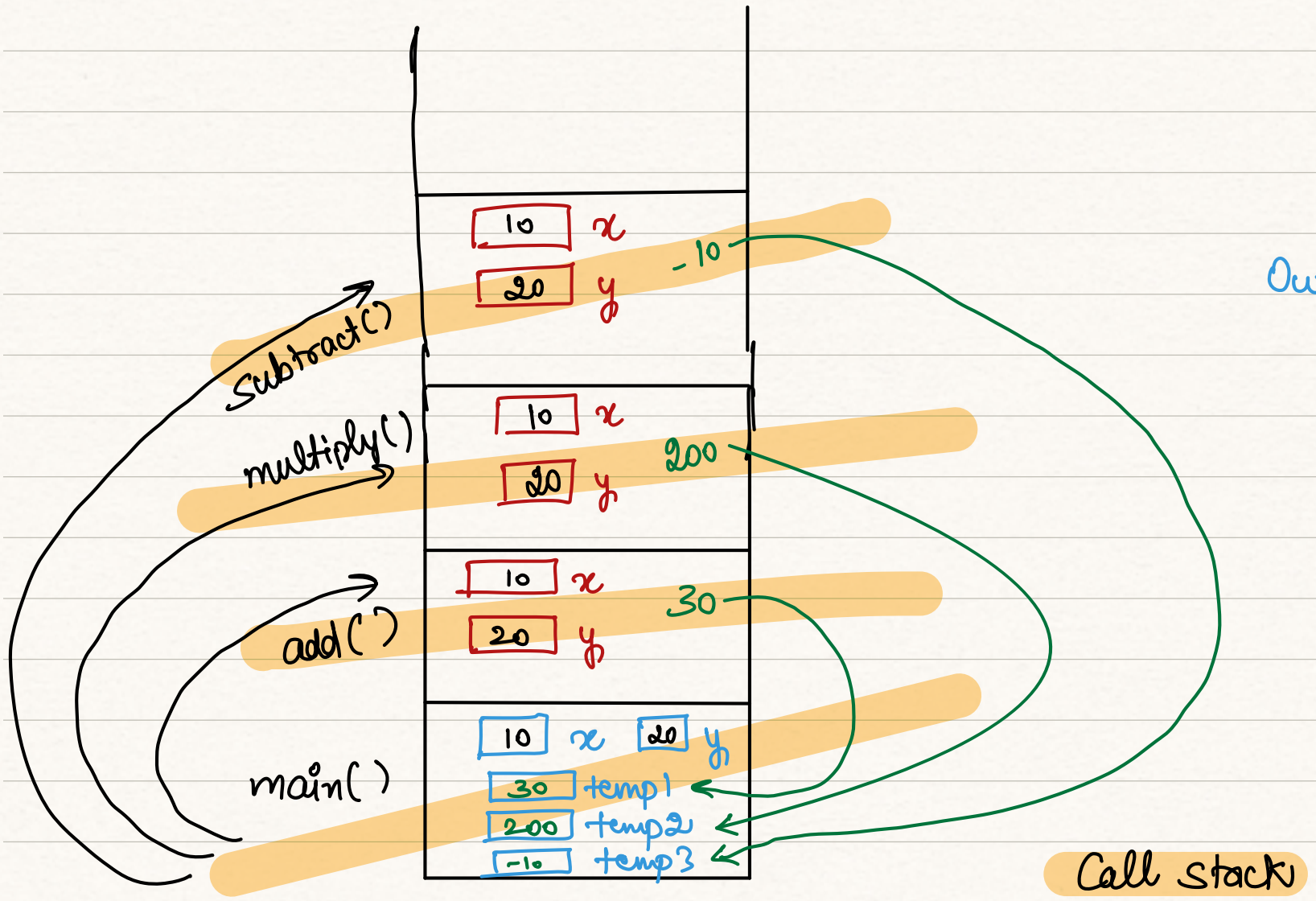
remove C
remove B
remove A

Last In First Out

Call stack

```
main () {
1   int x = 10;
2   int y = 20;
3   int temp1 = add (x, y);
4   int temp2 = multiply (x, y);
5   int temp3 = subtract (x,y);
6   S.O.P (temp3);
}
```

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

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

int subtract (int x, int y){
    return x-y;
}
```



Output box →

-10

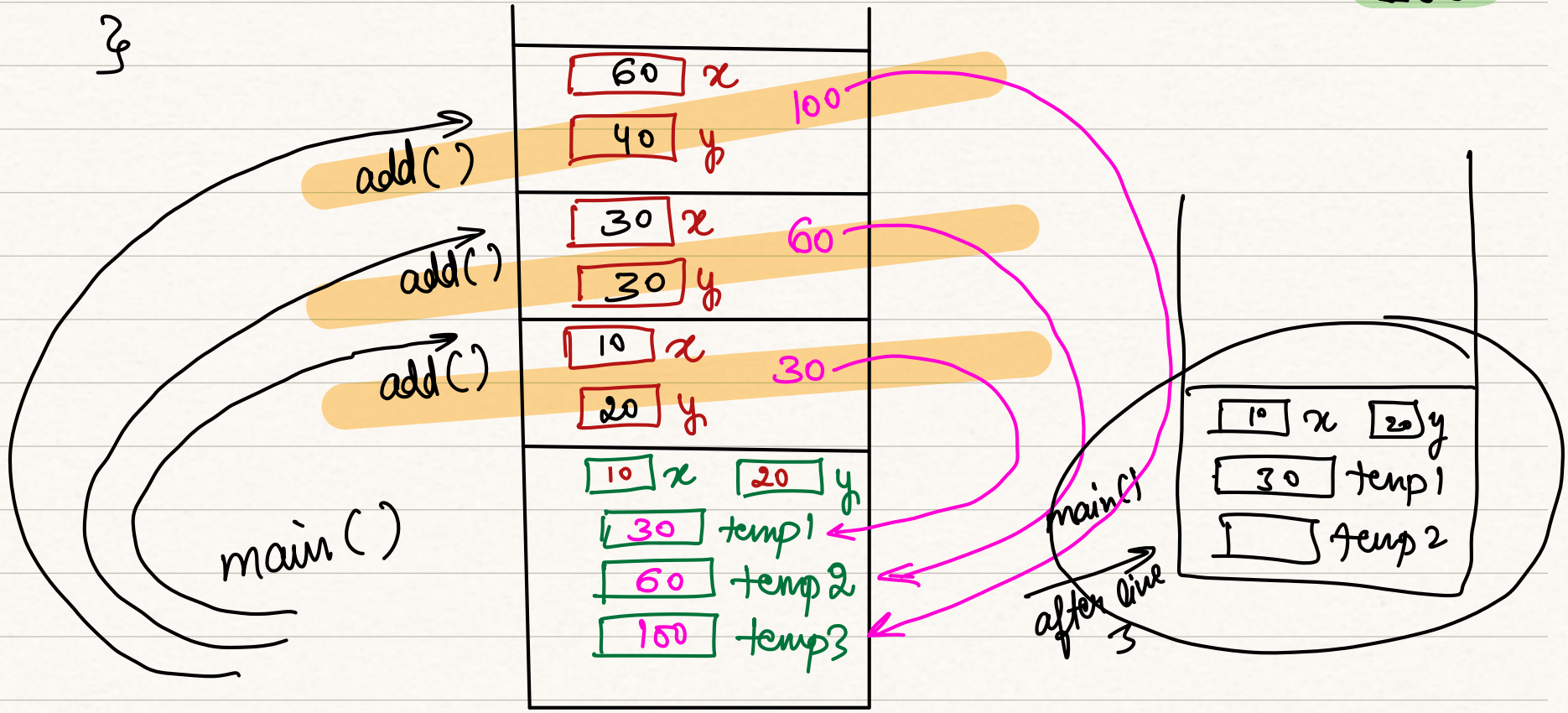
Call stack

```
main ( ) {
    int x = 10 ;
    int y = 20 ;
    int temp1 = add (x , y);
    int temp2 = add (temp1 , 30);
    int temp3 = add (temp2, 40);
    S.O.P (temp3);
}
```

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

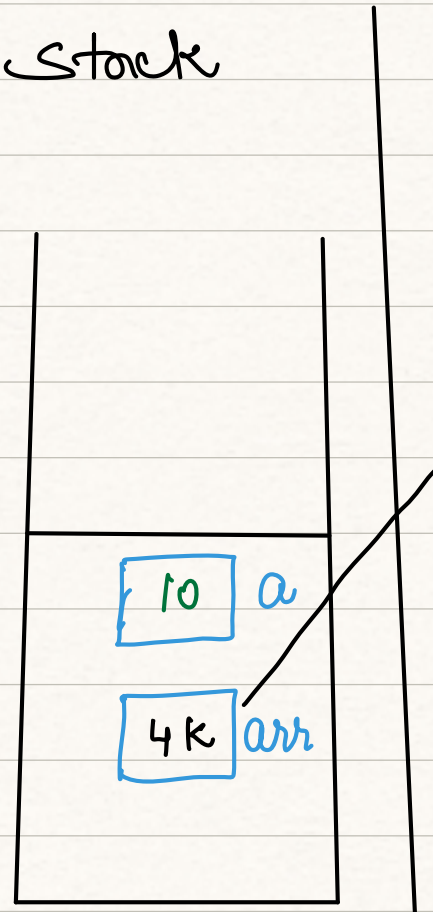
Output →

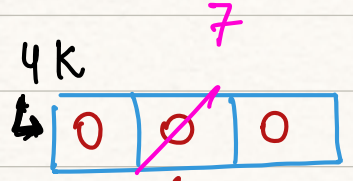
100



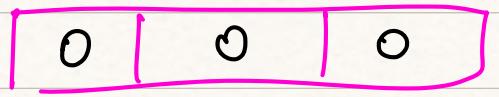
Memory management in JAVA

```
main () {
① int a = 10;
② int[] arr = new int[3];
   S.O.P (arr);

   arr[1] = 7;

}
```





arr[7] = 4K + 7x4
= 4K + 28
= 4028

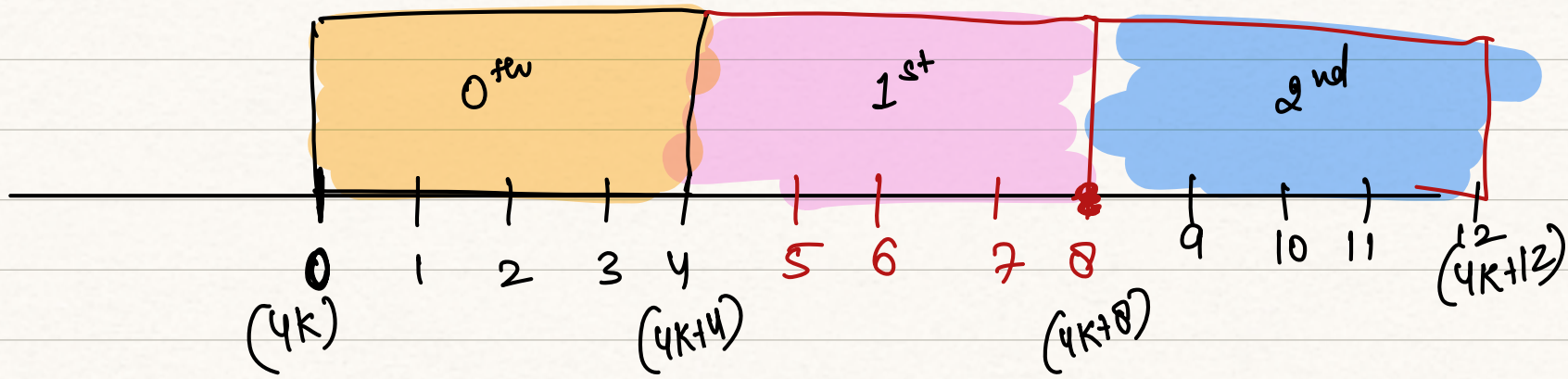


4 bytes

0 → 4K (4000)
1 → 4K+4bytes (4004)
2 → 4K+8 bytes (4008)

(assumption)

Address of the container is stored in stack and actual container is stored in heap.



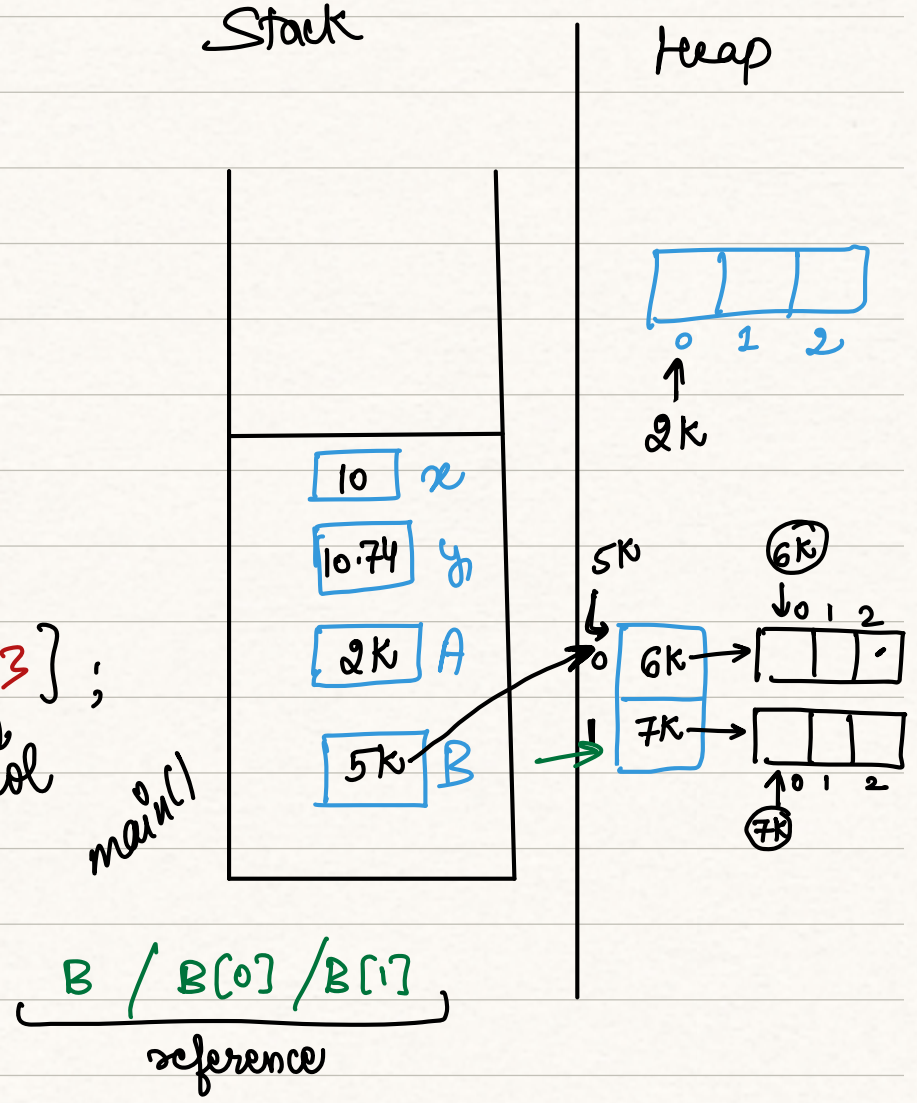
① primitive data types → int, long, float, double, boolean, char

[memory will be assigned in stack]

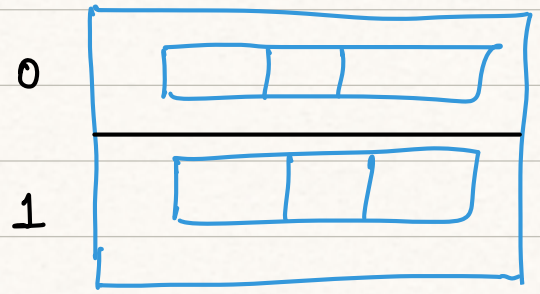
② the reference / address of containers will also store in stack

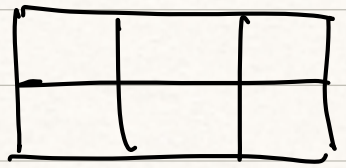
③ container [Array / Arraylist / HashMap / String] will be stored in heap.

```
main() {
1  int x = 10;
2  float y = 10.74f;
3  int [] A = new int[3];
4  int [][] B = new int [2] [3];
                    ↓row  ↓col
}
```



array of arrays

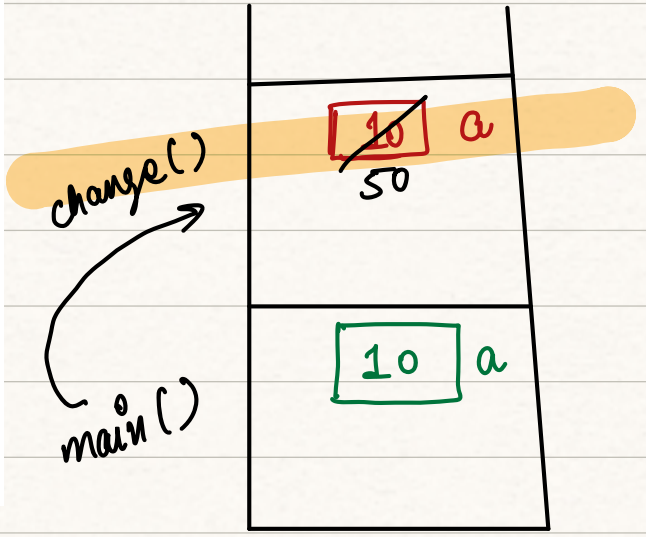




Quiz 1.

Predict the output :

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

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



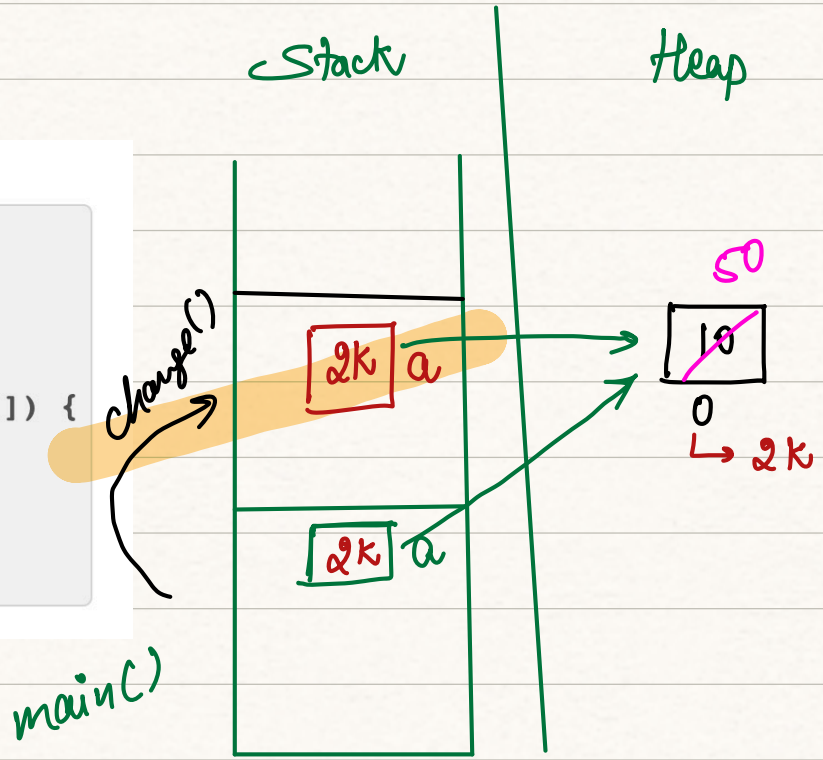
Output →

10

Quiz 2

Predict the output -

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

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

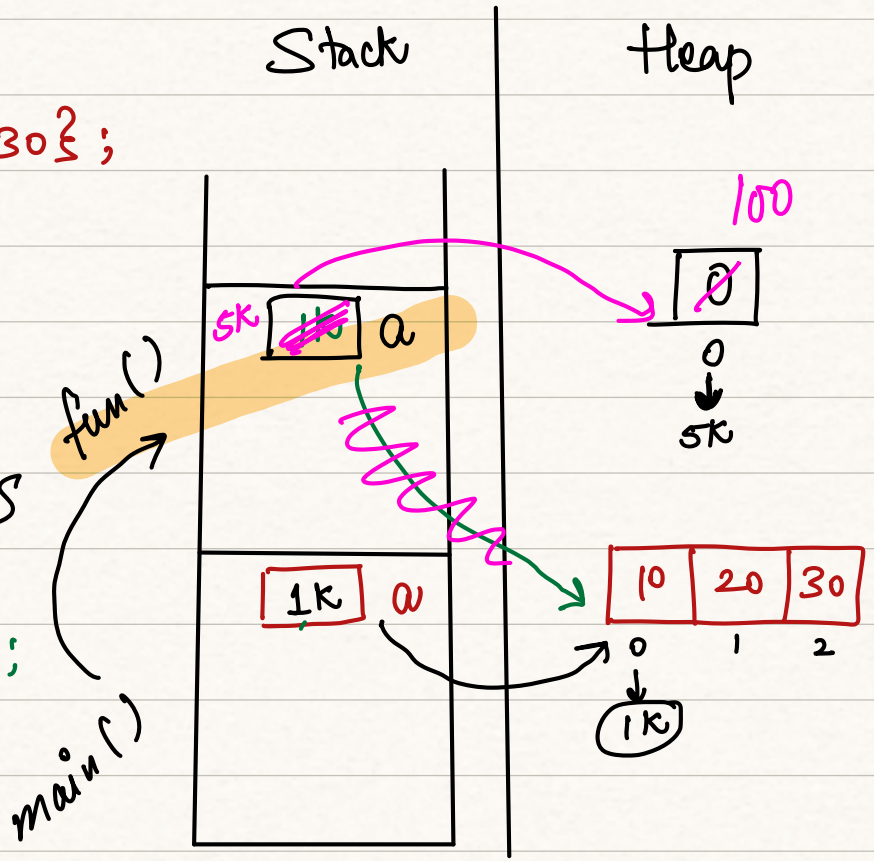


2K + 0*4 ⇒ (2K)

Output → 50

Quiz 3

```
main () {
// int [] a = {10, 20, 30};
// fun (a);
// S.O.P (a[0]);
}
void fun (int [] a) {
// a = new int [1];
// a[0] = 100;
}
```



Output → 10

Swap two variables [using a third variable]

a = 10 b = 20

```
temp = a
a = b
b = temp
```

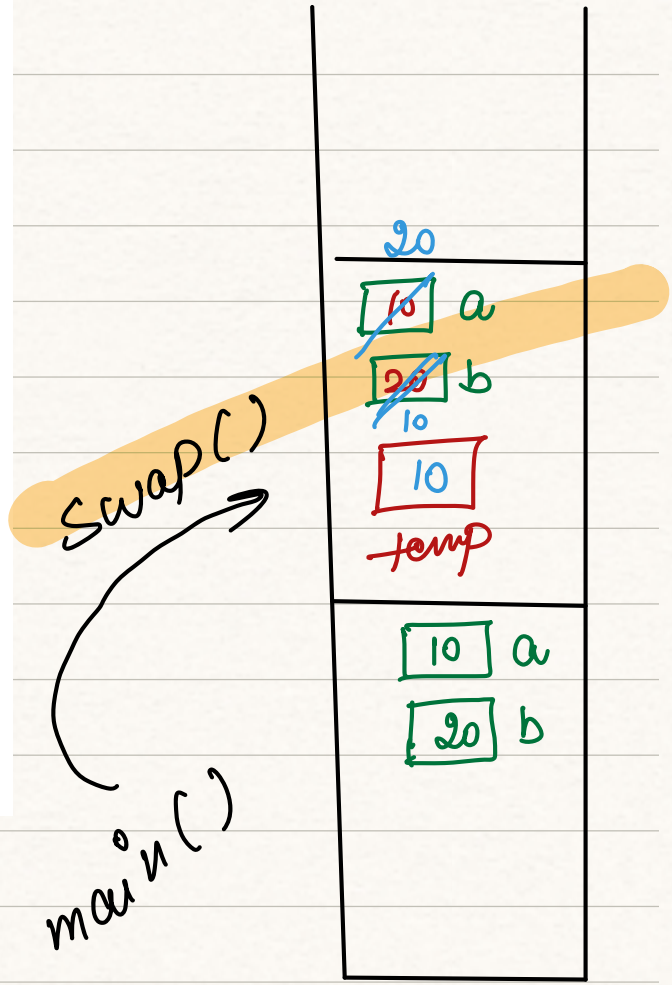
a = 20
b = 10

temp = 10
a = 20
b = 10

Quiz 4.

Predict the output -

```
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);
}
```



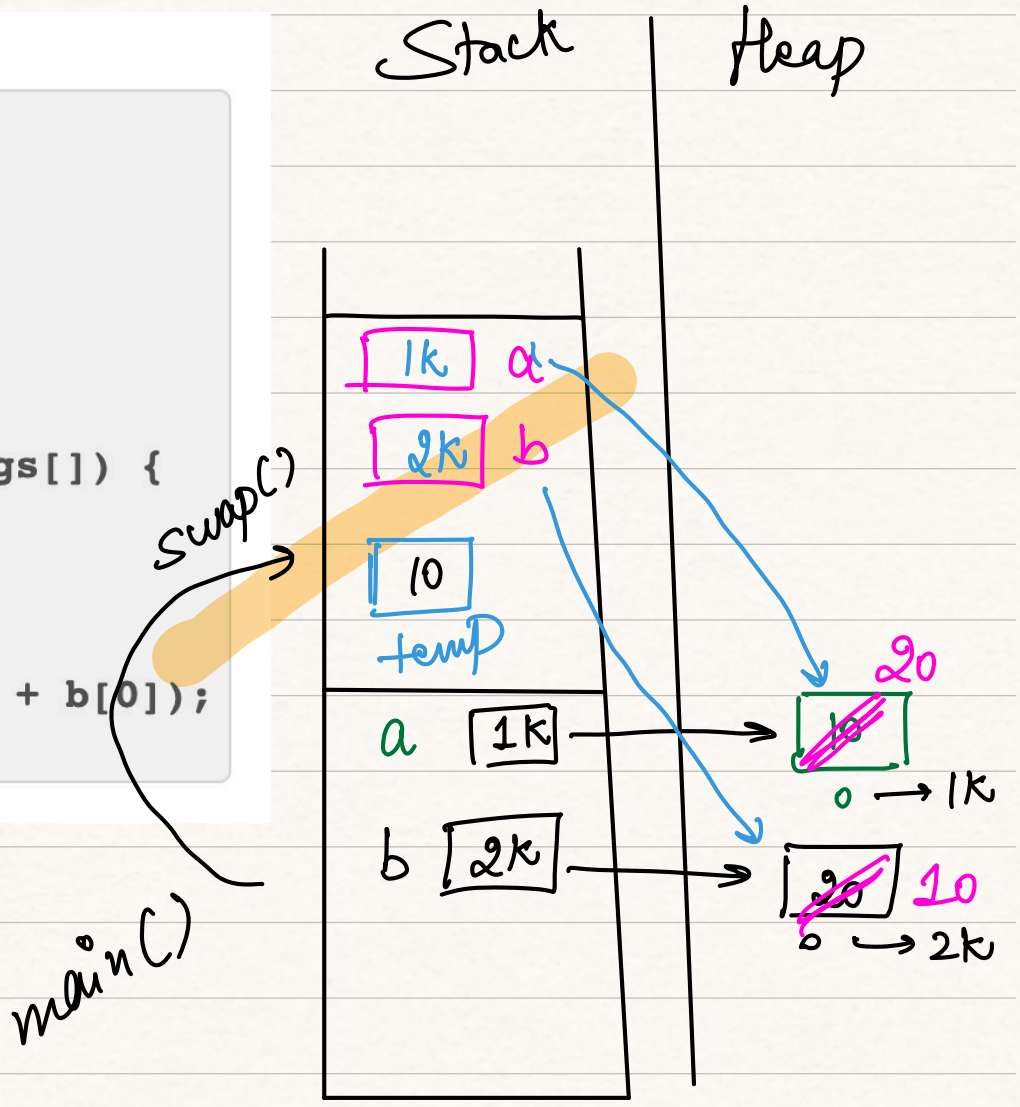
Output →

10 20

Quiz 5

Predict the output -

```
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]);
}
```



a[0]

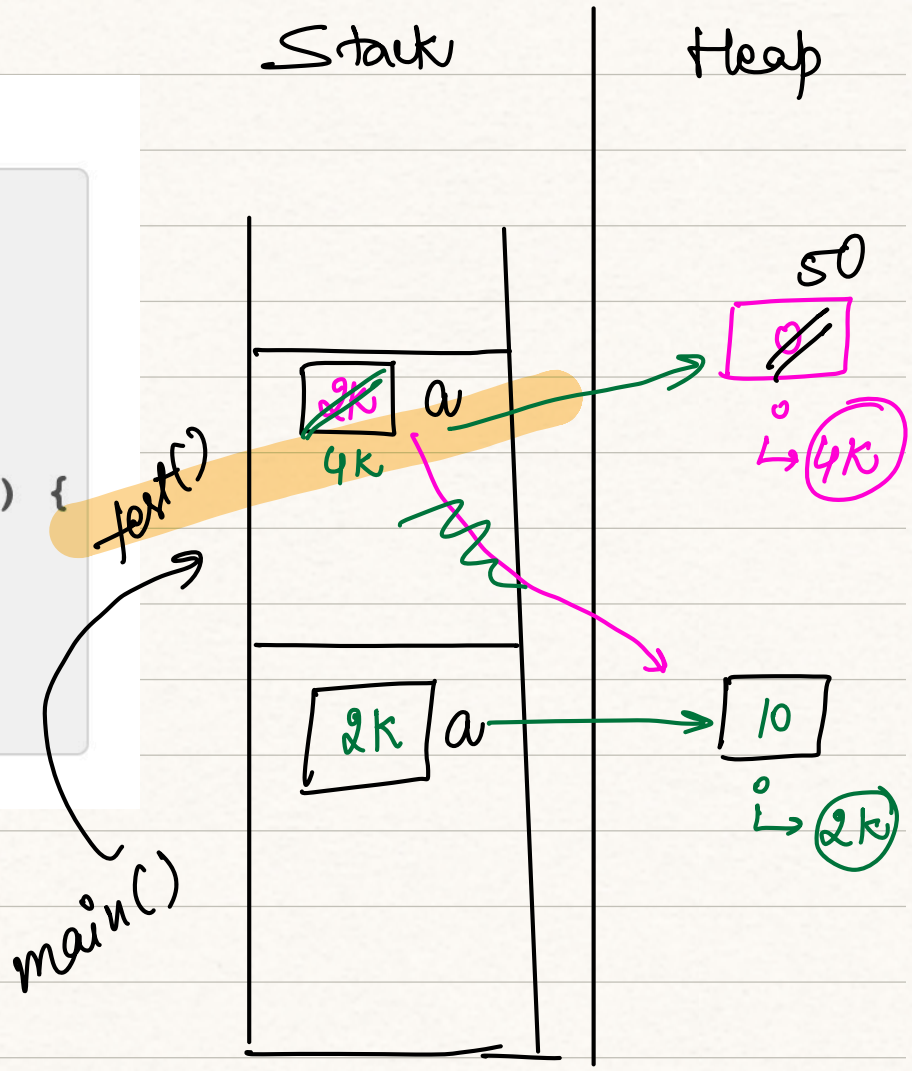
Output →

20 10

Quiz 6.

Predict the output -

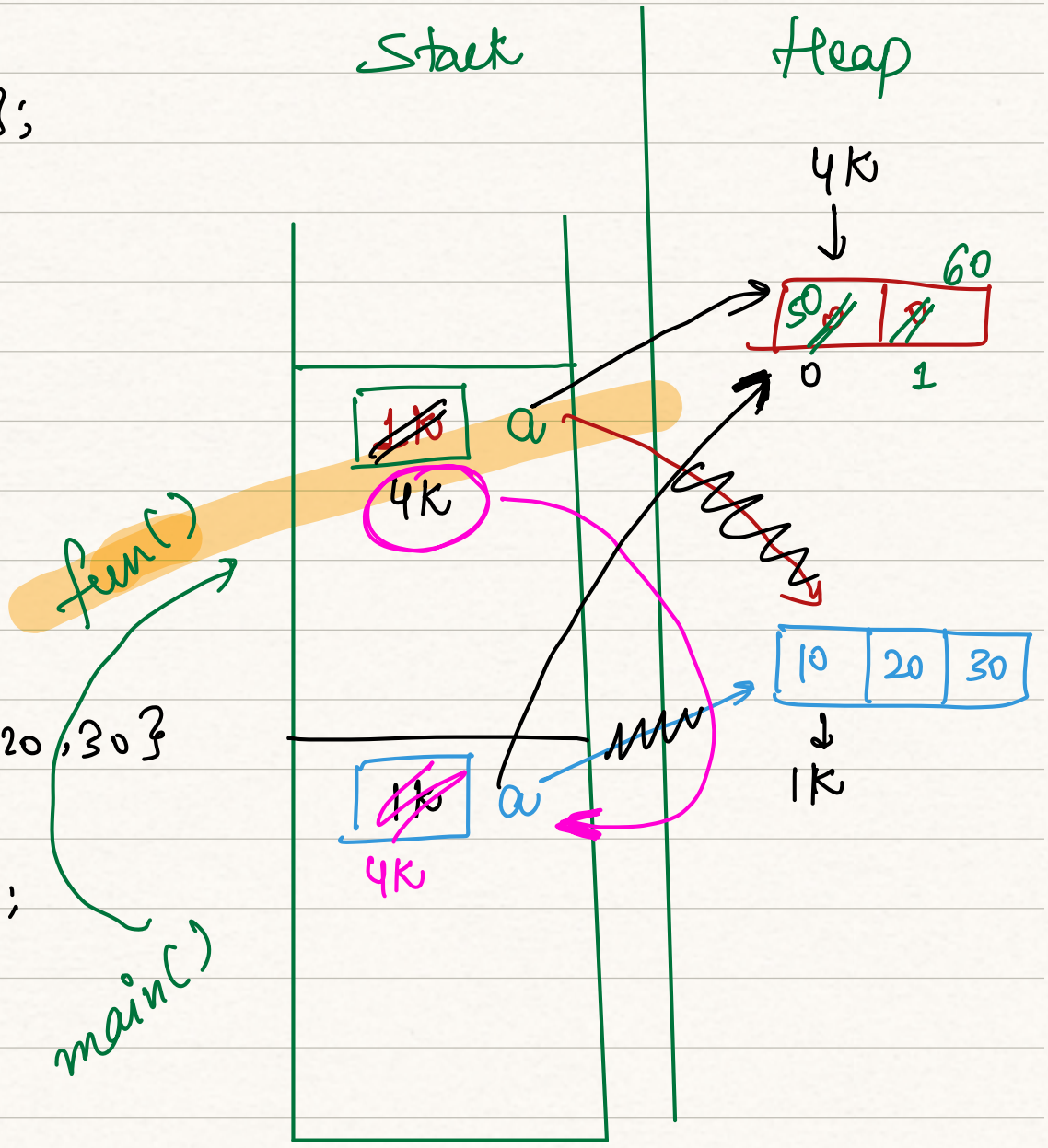
```
static void test(int[]a) {
    a = new int[1];
    a[0] = 50;
}

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



Output →

10

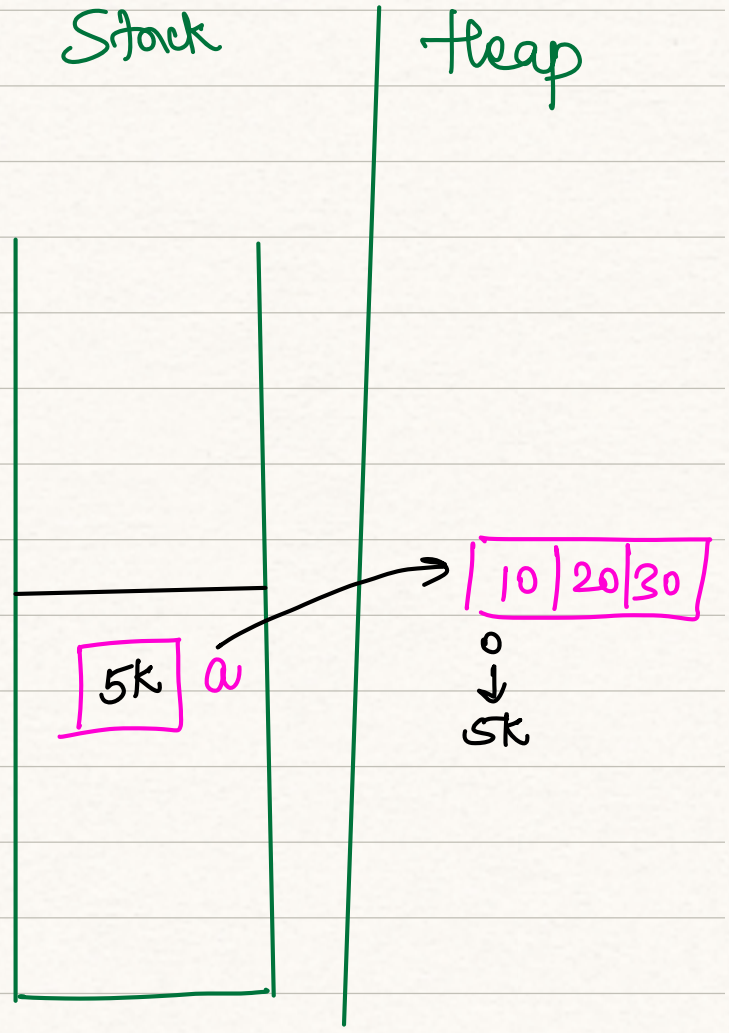
```
int[] fun (int[]a) {
    a = new int[2];
    a[0] = 50;
    a[1] = 60;
    return a;
}

main() {
    int[] a = {10, 20, 30};
    a = fun(a);
    S.O.P (a[0]);
}
```



Output :

50

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

main() {
    // int[] a = {10, 20, 30};
    // S.O.P. (a[0]);
}
```



Output → 10

(long)(s*s*s)

int * int → int

$10^5 * 10^5$

10^{10}

overflow

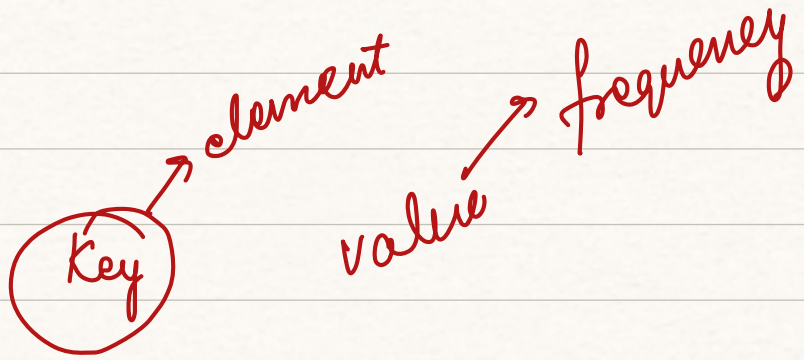
(long)s *s *s

solve (double celsius)

C → F

double ans = celcius *9 / 5 +32

Math. round (ans *100) / 100d




```
HashMap <Integer, Integer> map = - - - -

for (int i=0; i< A.length ; i++ ) {

    if (map.containsKey (A[i]) == true) {
            int oldfreq = map.get (A[i]);
            map.put (A[i], oldfreq+1);

    } else {
            map.put (A[i] , 1) ;

    }
```

```
int count = 0;

for (int x : map.keySet()) {

    if (map.get(x) == 1) {

        count++;
    }
}
return count;
```

```
Str = "India";
      0 1 2 3 4

int ch = (int) str.charAt(0);
                  I
```

ch = 73

S.O.P (ch) → 73

```
int [] ans = new ans [B.length];

for( int i=0 : i< B.length; i++) {

        ans [i] = map.get (B[i]);
}
return ans;
```