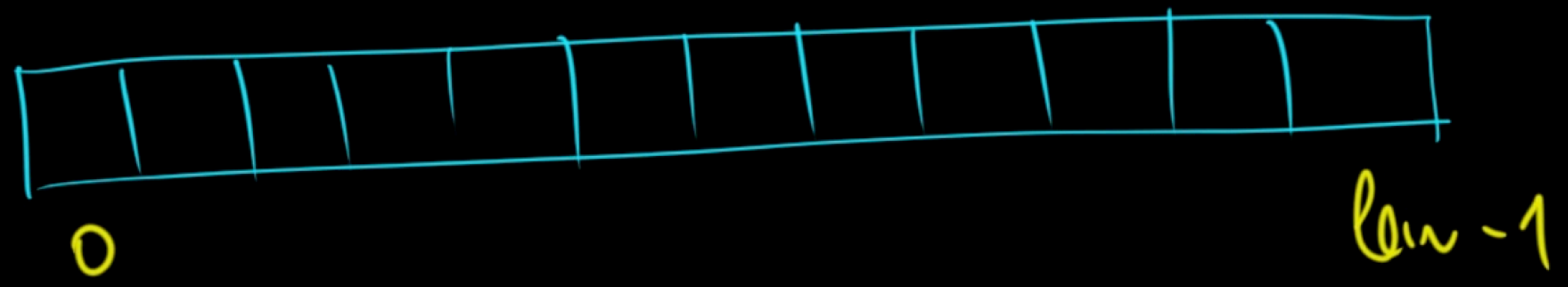
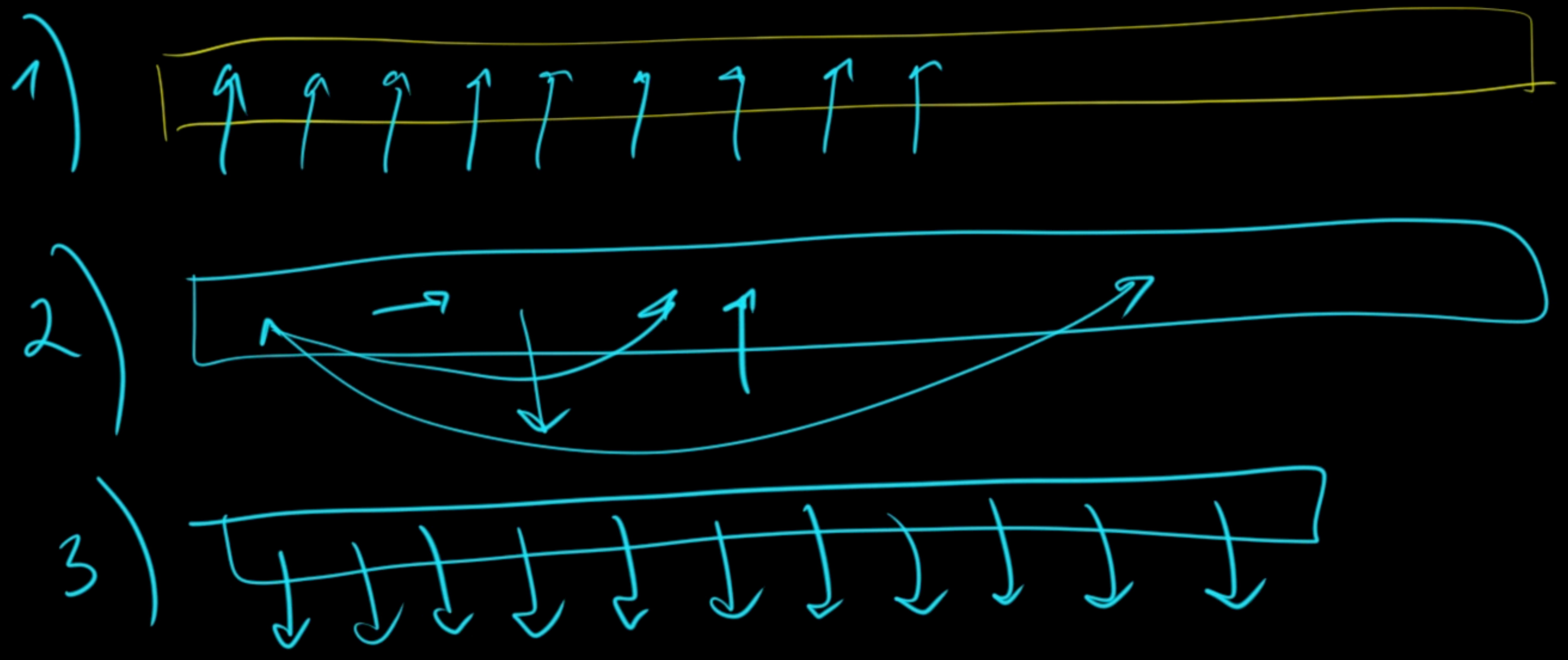
HEAD
commit1
commit2
commit online editor
right
main
left
ide
IDE
right
1) git pull
HEAD
2)
3 git commit
4 git push
remote

$x = 5$

$y = 7$

$k = 3.5$





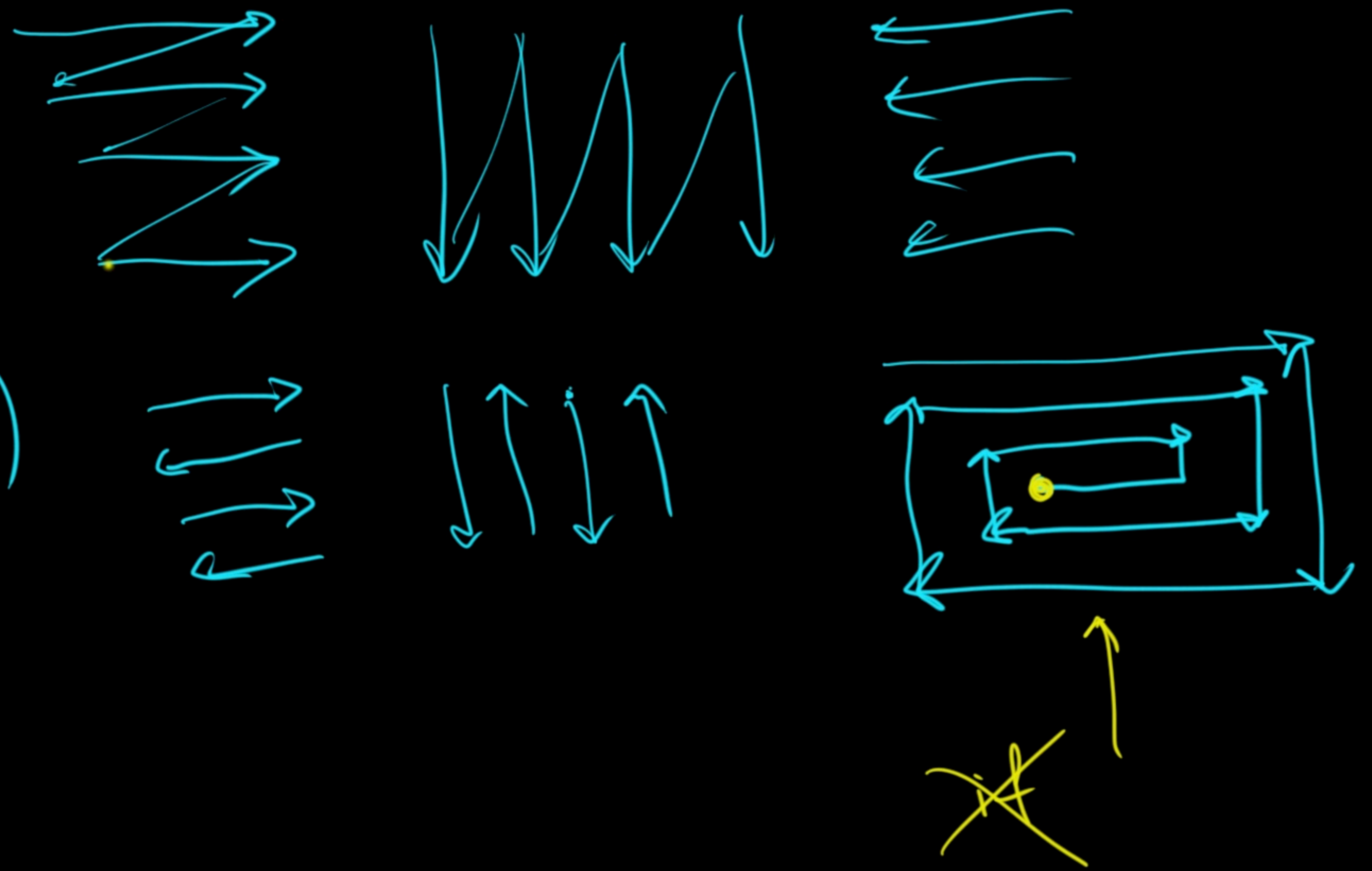
byte -128 .. 127
short

int
long

float
double

i8 -128 .. 127
u8 0 .. 255
i16
u16
i32
u32
i64
u64
i128
u128
f32
f64

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



for (i=1 ... 16)

all   x%2==0

int[] → int[]

```java
for (int i = 0; i < as.length; i++) {
  System.out.printf("%d", as[i]);
  if (i != as.length-1)
    System.out.print(",");
}
```

```
-20,18,-40,-33,45,-22,26,-24,23,-1,-30,1,-27,2,46,22,36,-39,-9,-14
```

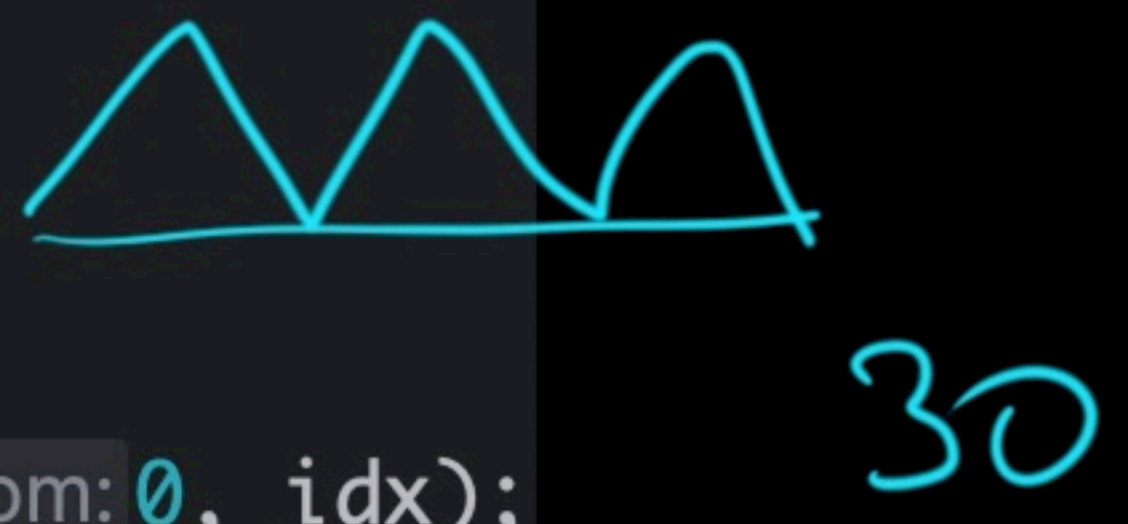
```
-20,18,-40,-33,45,-22,26,-24,23,-1,-30,1,-27,2,46,22,36,-39,-9,-14
```

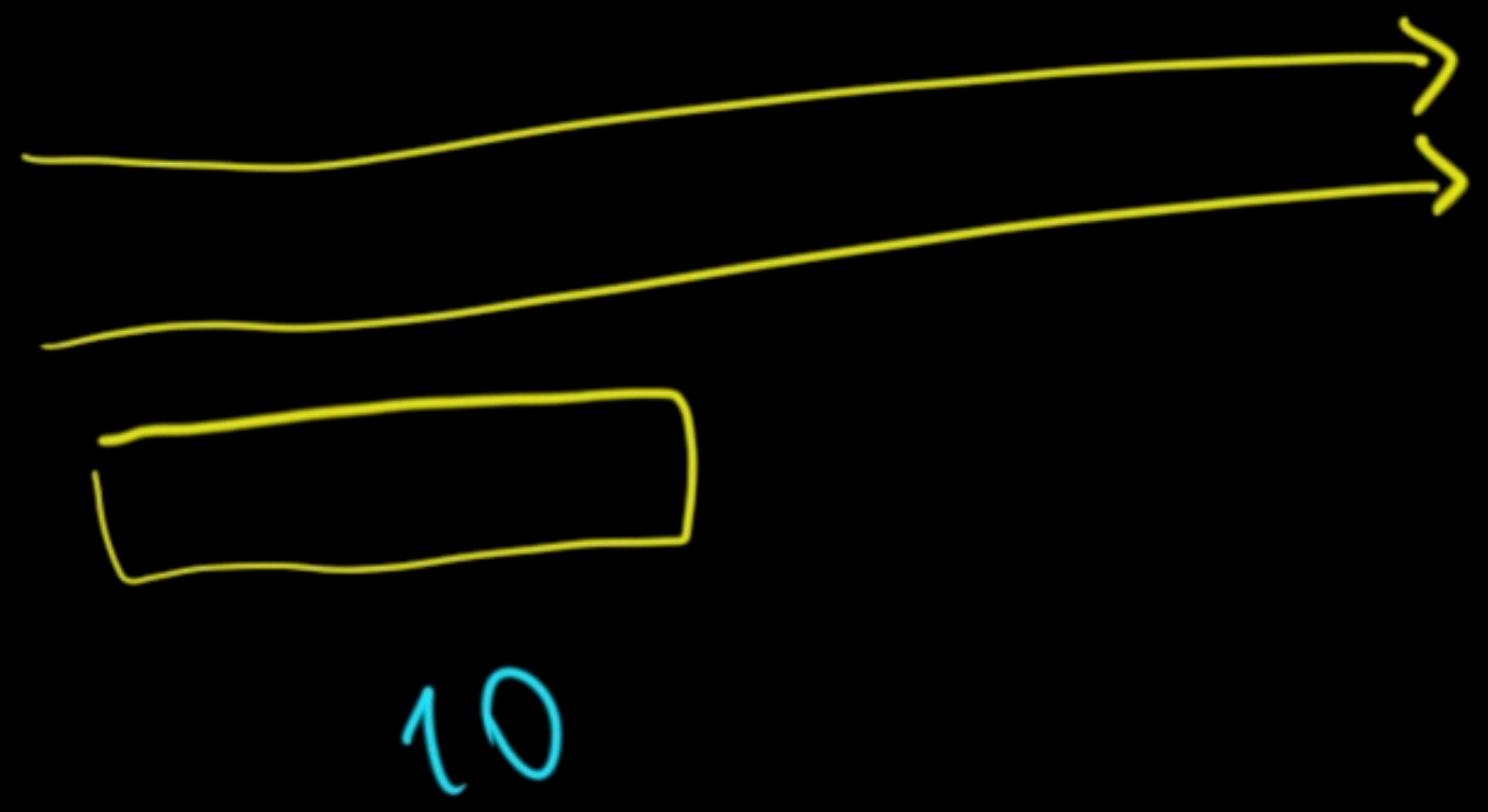
```java
for (int i = 0; i < as.length; i++) {
  if (i > 0) System.out.print(",");
  System.out.printf("%d", as[i]);
}
```

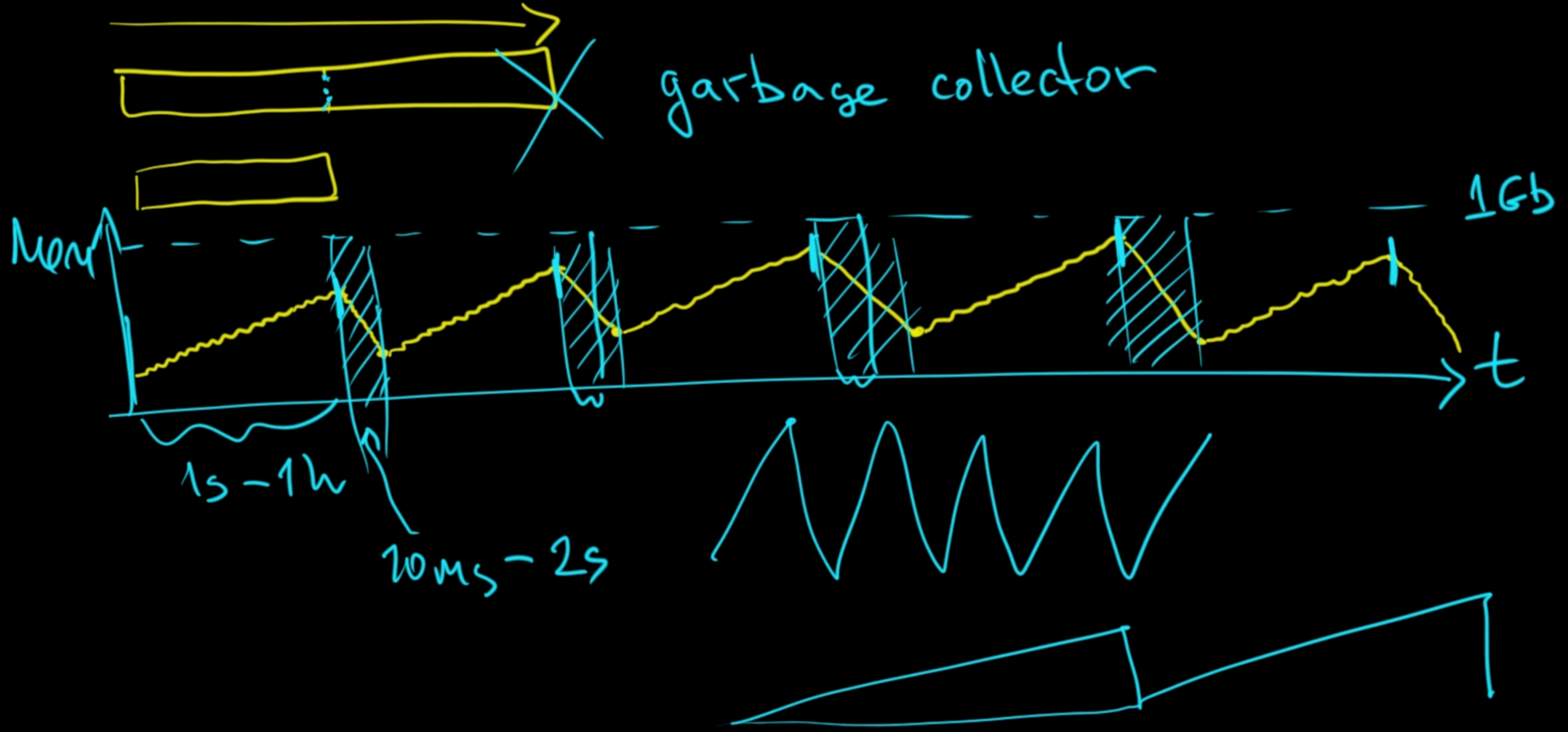
```
[30, 0, 35, -32, 6, -2, 9, 38, 42, 34, 50, -6, 46, 22, 27, 11, -26, -19, 12, 37]
```

```java
static int[] collectNegatives1(int[] xs) {
  // count negatives
  int nc = 0;
  for (int i = 0; i < xs.length; i++) {
    if (xs[i] < 0) nc++;
  }
  // collect negatives
  int[] outcome = new int[nc];
  int idx = 0;
  for (int i = 0; i < xs.length; i++) {
    if (xs[i] < 0) {
      outcome[idx] = xs[i];
      idx++;
    }
  }
  return outcome;
}
```

```java
static int[] collectNegatives2(int[] xs) {
  int[] outcome = new int[xs.length];
  int idx = 0;
  for (int i = 0; i < xs.length; i++) {
    if (xs[i] < 0) {
      outcome[idx] = xs[i];
      idx++;
    }
  }
  return Arrays.copyOfRange(outcome, from: 0, idx);
}
```







```java
static int[] collectNegatives2(int[] xs) {
  int[] outcome = new int[xs.length];
  int idx = 0;
  for (int i = 0; i < xs.length; i++) {
    if (xs[i] < 0) {
      outcome[idx] = xs[i];
      idx++;
    }
  }
  return Arrays.copyOfRange(outcome, from: 0, idx);
}
```

```java
static int[] collectPositives(int[] xs) {
  int[] outcome = new int[xs.length];
  int idx = 0;
  for (int i = 0; i < xs.length; i++) {
    if (xs[i] >= 0) {
      outcome[idx] = xs[i];
      idx++;
    }
  }
  return Arrays.copyOfRange(outcome, from: 0, idx);
}
```

$x \% 2 == 0$

$x \% 2 != 0$

$xs[i] < 0$

$xs[i] > 0$

$xs[i] == 0$

$xs[i] \% 2 == 0$

boolean

t f

$f : \text{int} \Rightarrow \text{boolean}$

|||

```java
interface FilterFn {
  boolean filter(int x);
}
```

```java
static int[] collectNegatives2(int[] xs) {
  int[] outcome = new int[xs.length];
  int idx = 0;
  for (int i = 0; i < xs.length; i++) {
    if (xs[i] < 0) {
      outcome[idx] = xs[i];
      idx++;
    }
  }
  return Arrays.copyOfRange(outcome, from: 0, idx);
}
```

```java
interface FilterFn {
  boolean test(int x);
}
```

```java
static int[] collect(int[] xs, FilterFn fn) {
  int[] outcome = new int[xs.length];
  int idx = 0;
  for (int i = 0; i < xs.length; i++) {
    if (fn.test(xs[i])) {
      outcome[idx] = xs[i];
      idx++;
    }
  }
  return Arrays.copyOfRange(outcome, from: 0, idx);
}
```

```java
static int[] collectNegatives2(int[] xs) {
  return collect(xs, x -> x < 0);
}
```

λ function

int

boolean

```java
static int[] collectPositives(int[] xs) {
  return collect(xs, x -> x >= 0);
}
```

```java
static int[] collectNegatives3(int[] xs) {
  var fn = new FilterFn() {
    @Override
    public boolean test(int x) {
      return x < 0;
    }
  };
  return collect(xs, fn);
}
```

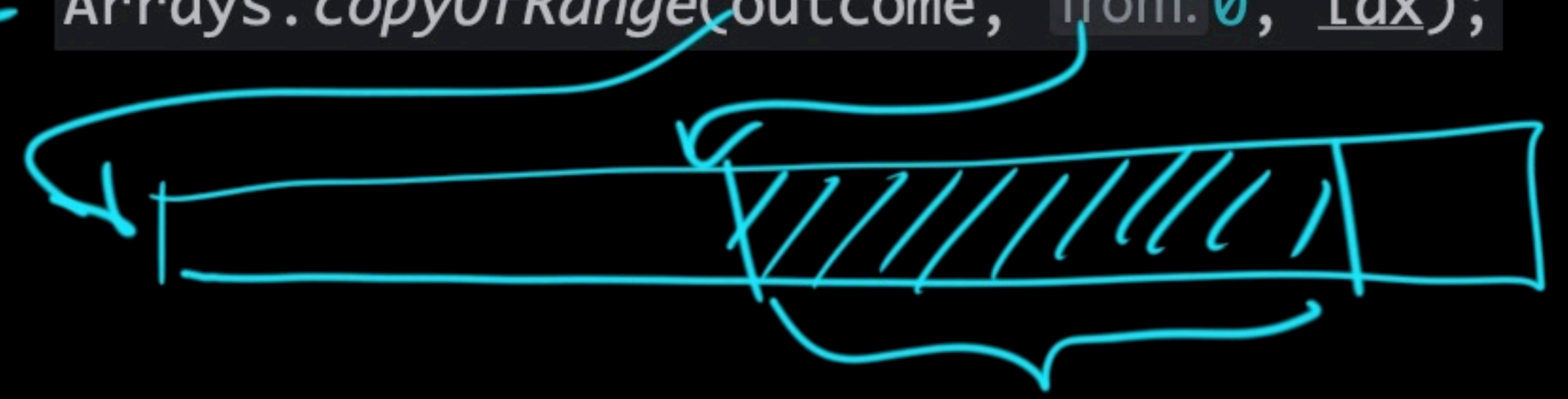
()

println (x)

$f: A \Rightarrow B$

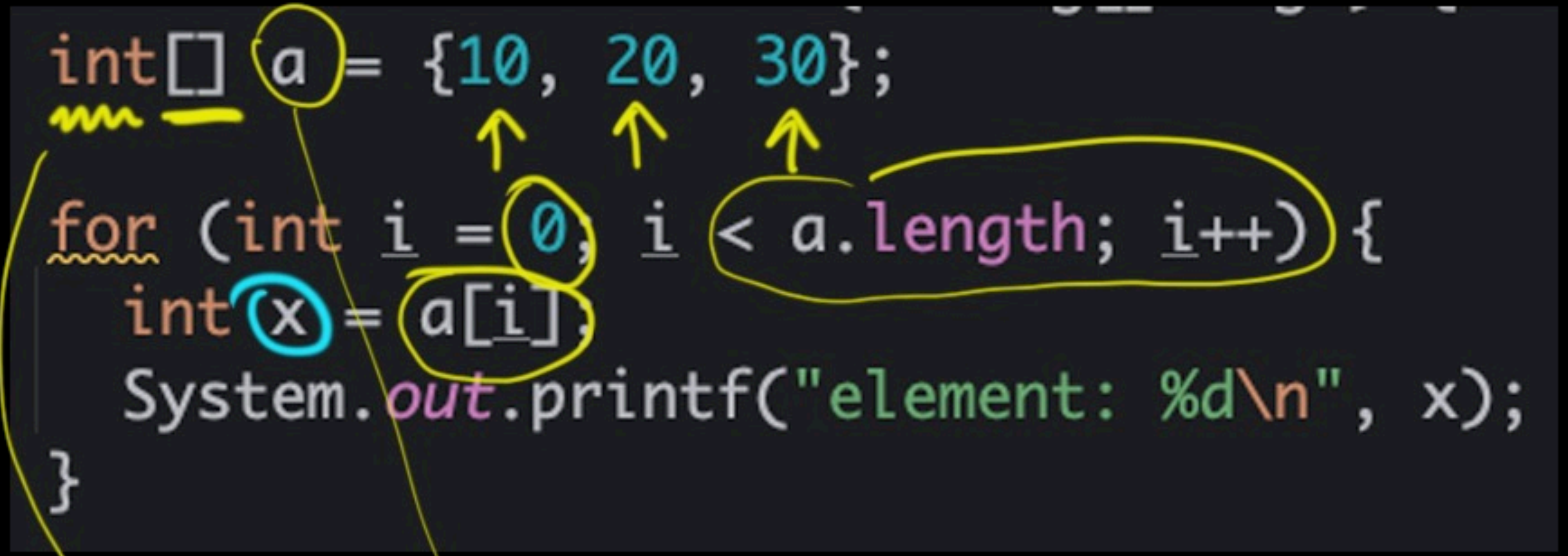
$inc: x \to x+1$

xs= `Arrays.copyOfRange(outcome, from: 0, idx);`



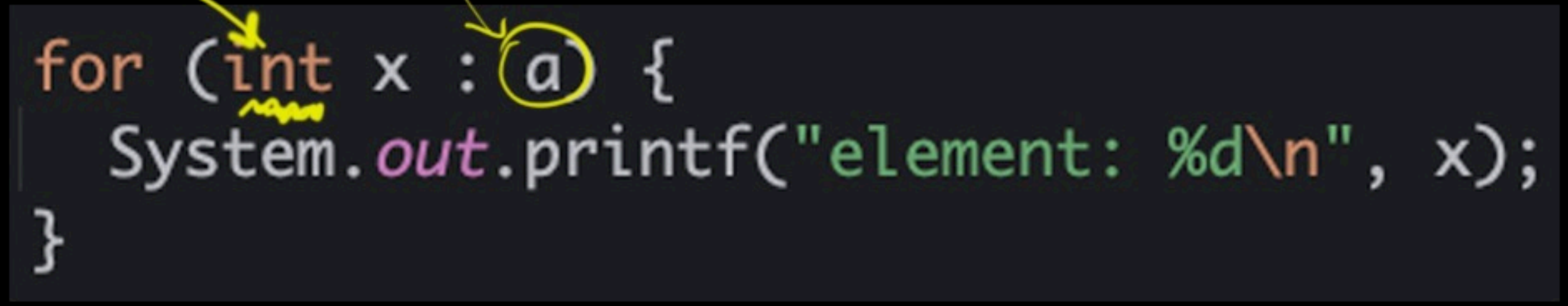
s= `Arrays.toString(as)`

[1,2,3] $\to$ String

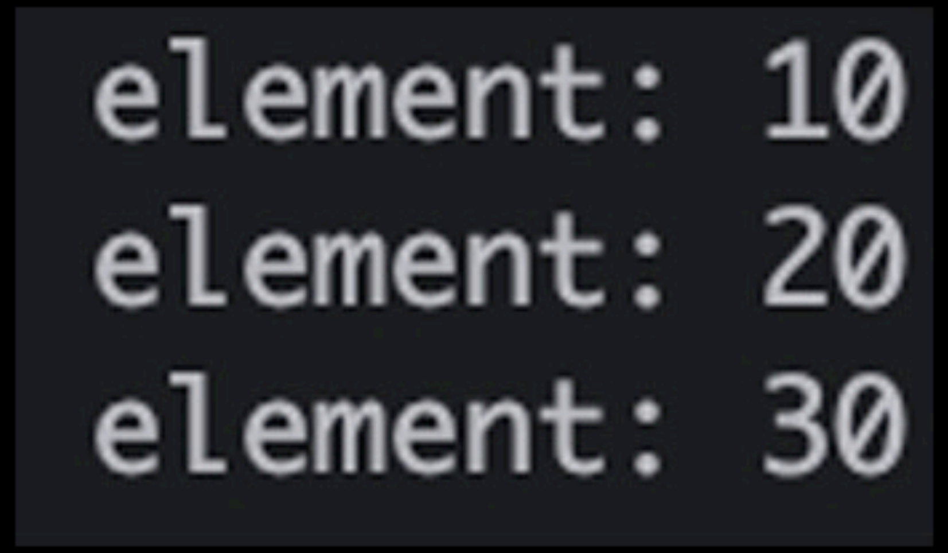


```
int[] a = {10, 20, 30};

for (int i = 0; i < a.length; i++) {
  int x = a[i];
  System.out.printf("element: %d\n", x);
}
```



```
for (int x : a) {
  System.out.printf("element: %d\n", x);
}
```



```
element: 10
element: 20
element: 30
```

10 , 20 , 30

int x