

CNAPS 12 A 4-15-10

11

Ex. Sorts 3 numbers.

```
int a, b, c, temp;
```

```
    // get a, b, c from user
```

```
    if (a > b) {  
        temp = a;  
        a = b;  
        b = temp;
```

```
    }
```

```
    if (b > c) {  
        temp = b;  
        b = c;  
        c = temp;
```

```
    }
```

```
    if (a > b) {  
        temp = a;  
        a = b;  
        b = temp;
```

```
    }
```

```
    // Print a, b, c
```

Trace : input 3 2 1
 ↓ ↓ ↓
 a b c

<u>a</u>	<u>b</u>	<u>c</u>	<u>temp</u>
3	2	1	-
3	2	1	3
2	2	1	3
2	3	1	3
2	3	1	3
2	1	1	3
2	1	3	3
2	1	3	2
1	1	3	2
1	2	3	2

output : 1 2 3

recall: Conditional evaluation

(cond)? (exp1):(exp2)

Ex. Prints absolute value of difference

```

int a, b, c;
:
// get a, b from user
:

```

```

{
  if (a > b)
    c = a - b;
  else
    c = b - a;
  System.out.println(c);
}

```



could replace this by :

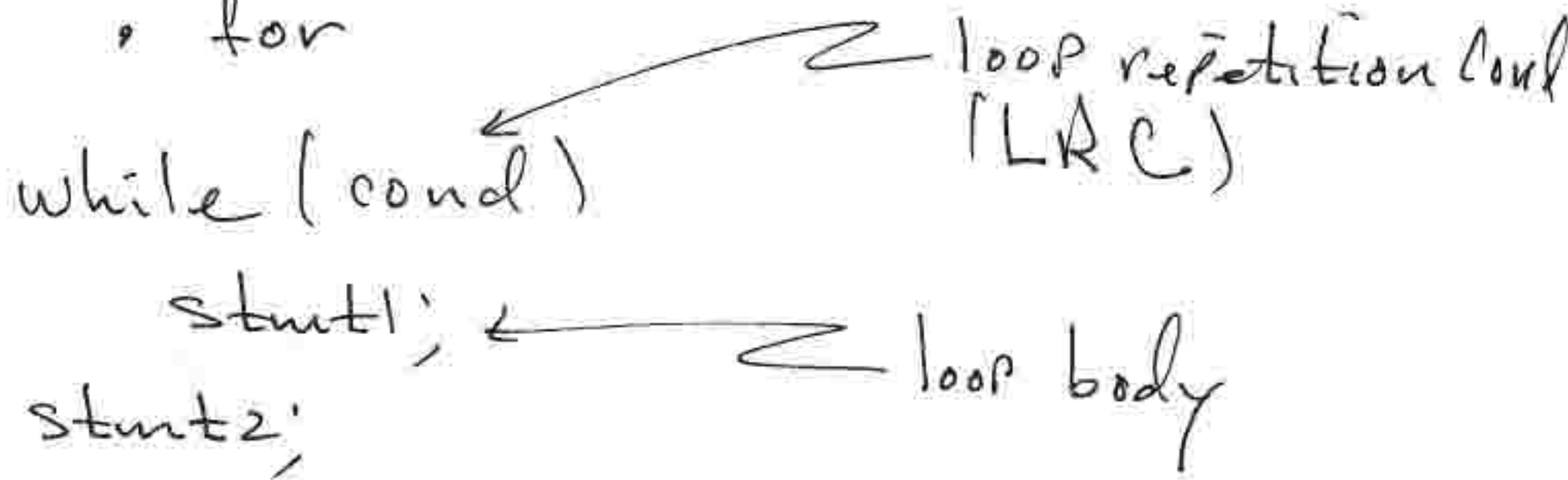
```

c = (a > b) ? (a - b) : (b - a);

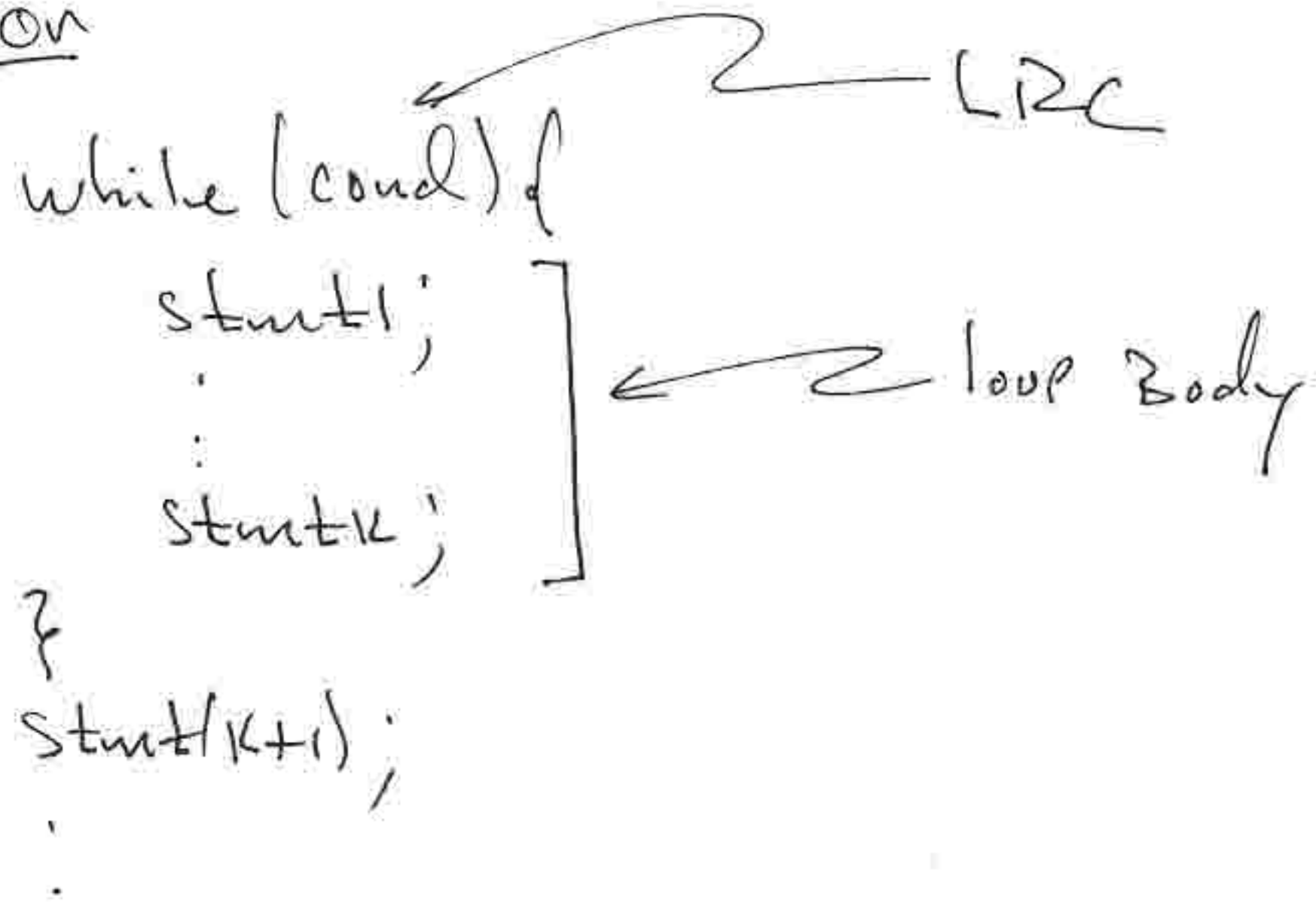
```

Iterative (looping)

- while
- do-while
- for

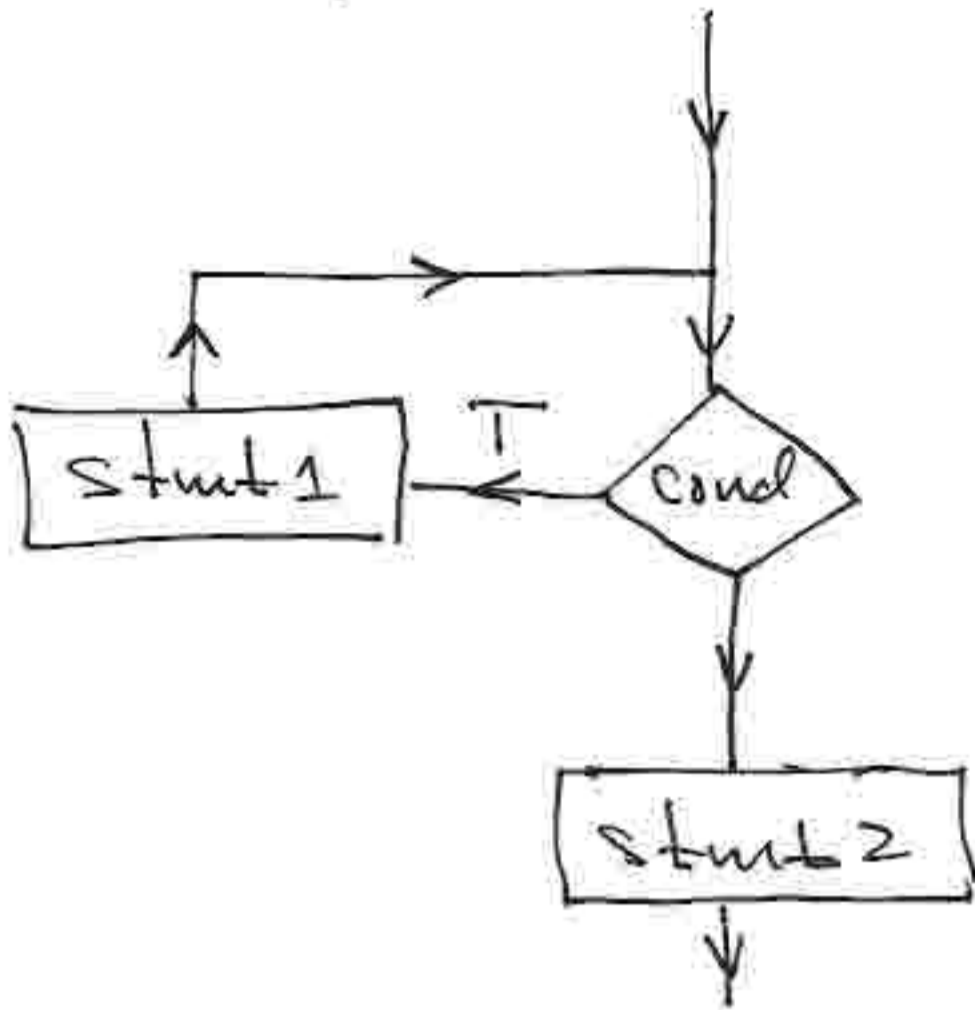


or



Flow Diagram

(5)



note: Possible that loop body is executed 0 times.

Ex. `int i, n = 7;`

`i = 1;`

`while (i <= n) {`

`System.out.println(i * i);`

`i += 1;`

`}`

`System.out.println("i = " + i);`

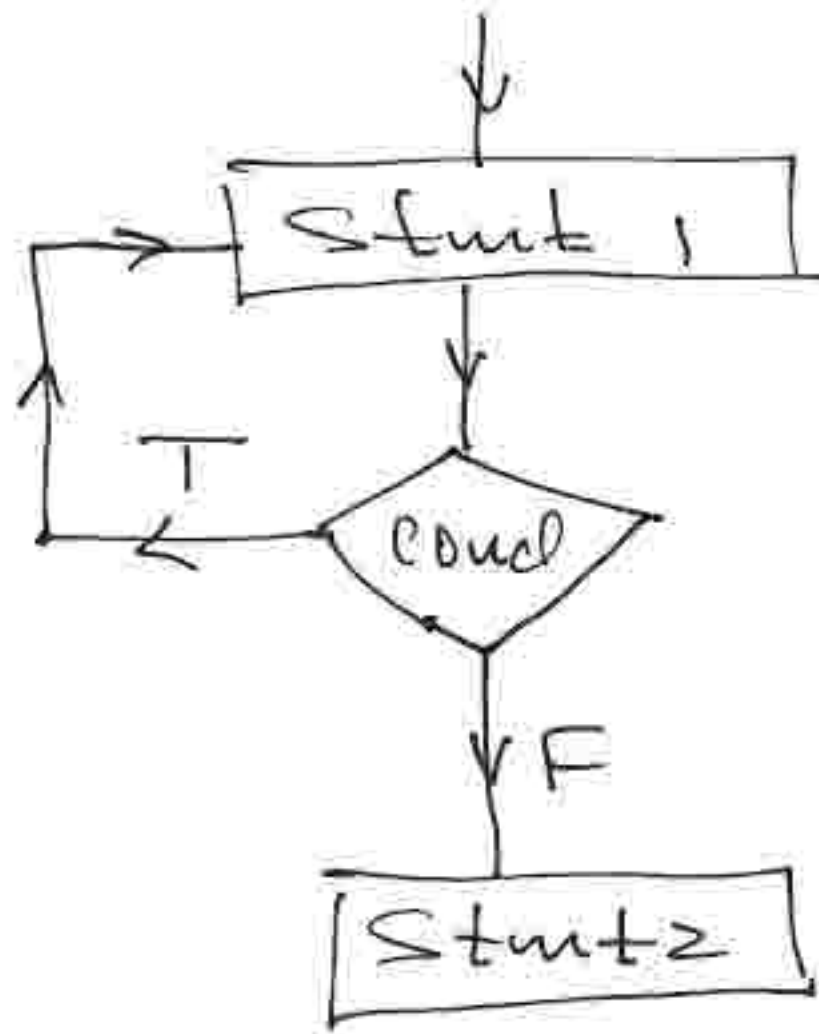
do
 stmt1; ← Body

while (cond); ← note semicolon
stmt2; ← LRC

or

do {
 stmt1;
 :
 stmtk; } ← Body
while (cond);
stmt(k+1);

Flow Diagram :



note:

loop body must execute at least once.

Ex. `int i, n = 7;`

`i = 1;`

`do {`

`System.out.println(`

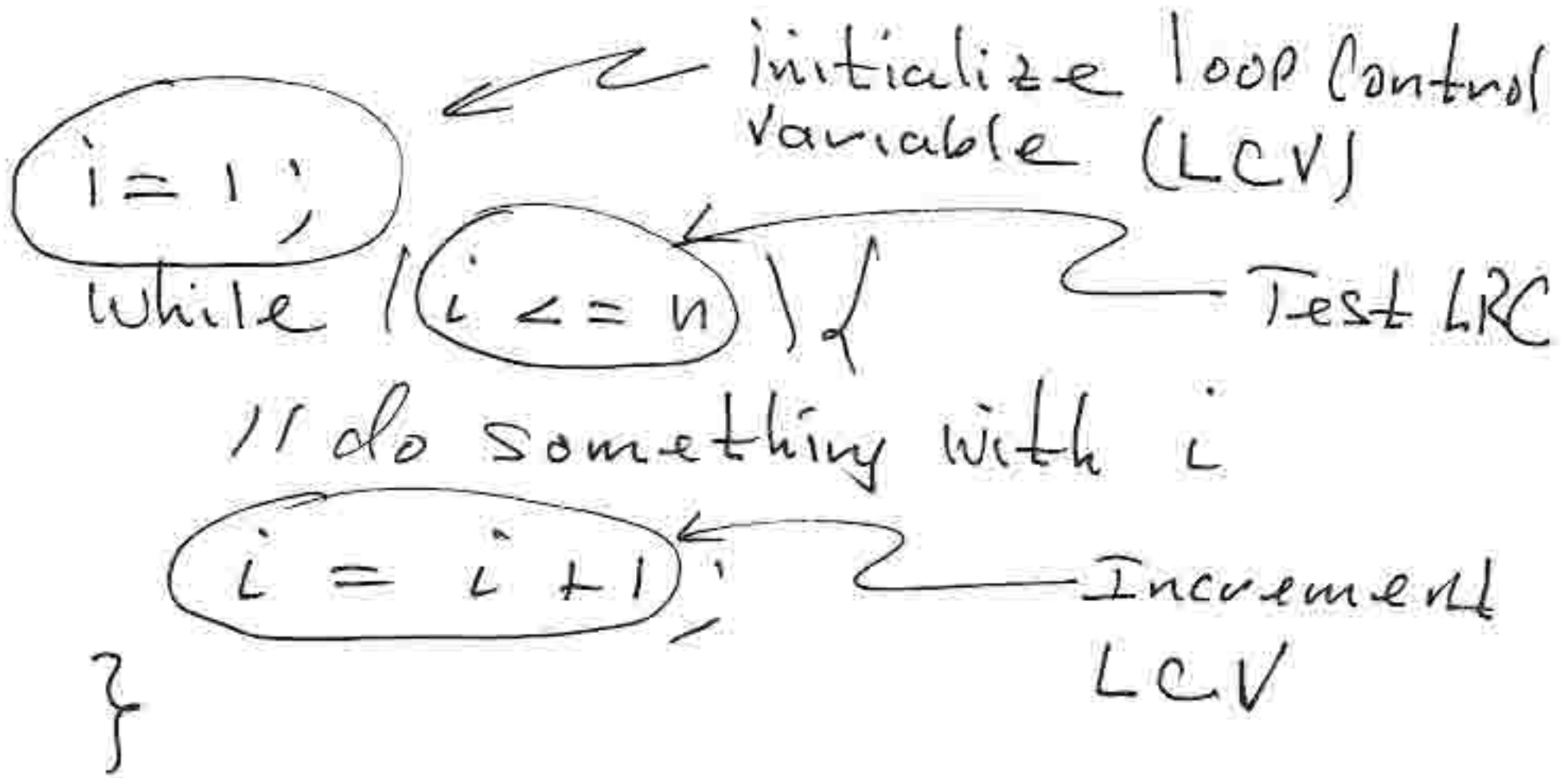
`(int) Math.Pow(i, 2));`

`i++;`

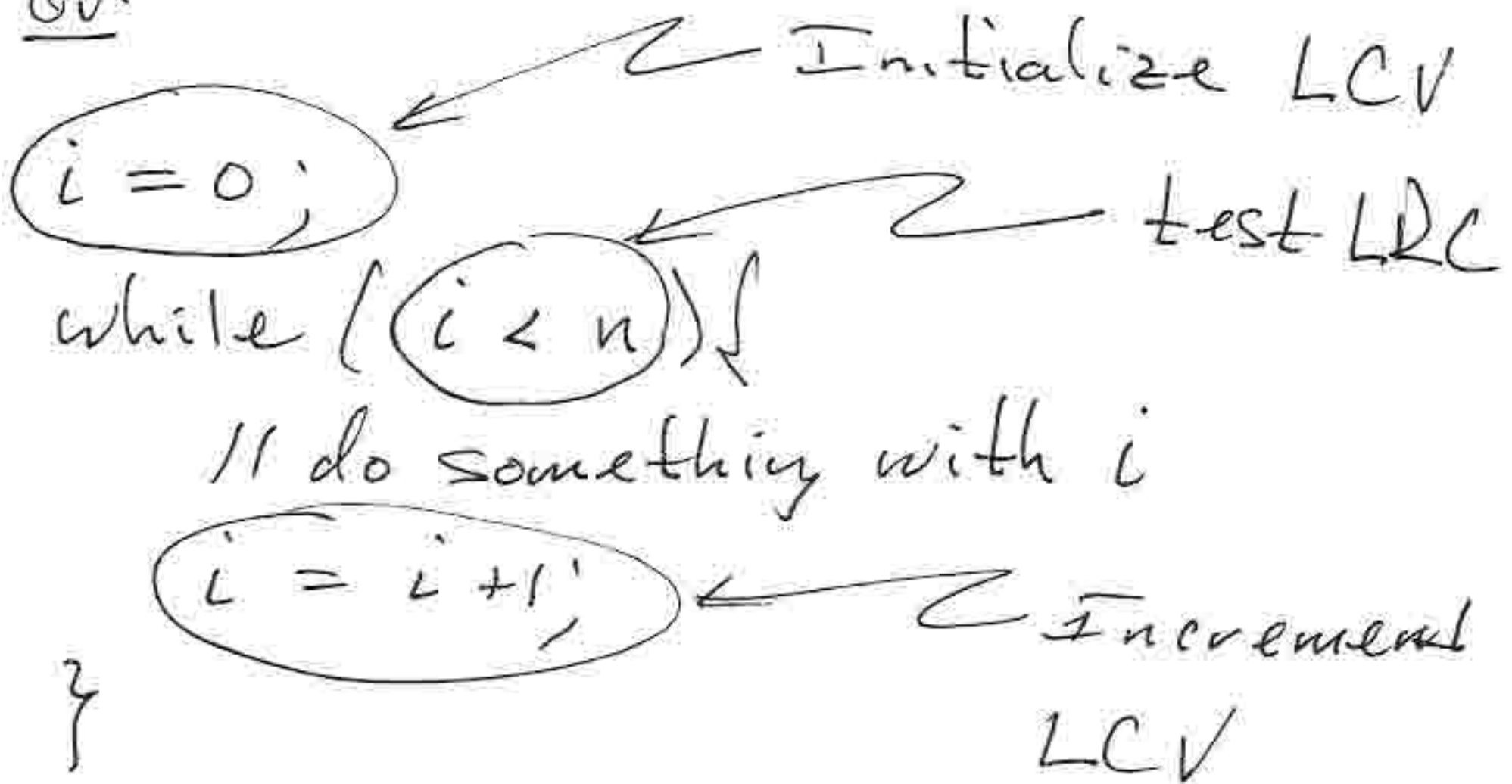
`} while (i <= n);`

`System.out.println("i = " + i);`

In while:



or



for loop places all 3 tasks in
same place.

L9

```
for (i=1; i<=n; i++) {  
    // do something  
}
```

or

```
for (i=0; i<n; i++) {  
    // do something  
}
```

Ex.

```
int i, n = 7;
```

```
for (i = 1; i <= n; i++)
```

```
    System.out.println(i * i);
```

```
    System.out.println("i = " + i);
```

Ex. Average example with for loop

```
// same Declarations
```

```
// same Prompt
```

```
for (item = sc.nextDouble(); item != 0;
```

```
    item = sc.nextDouble();
```

```
{
```

```
    sum += item;
```

```
    count++;
```

```
}
```

```
average = (count > 0) ? (sum / count) : 0;
```

```
// same output
```