Lecture 5

11 September 2018

Admin Matters
Common C Mistakes
Unit 9: **Logical Expression**
Unit 10: **Assertion**
Unit 11: **Loops**
Unit 12: **Reasoning about Loops**

Online Quizzes

now available

Source code from **Lecture 4** now available

Tutorial 4

Problem Sets from Unit 10-12 Today

Assignment 1

Due this Friday 6pm

Assignment 2

Release this Friday

(to be graded on correctness and style)

Assignment 2

Everything
up to
loops

Midterm

Venue: MPSH 1 (B)

2 October

4pm - 6pm

Midterm

Open Book

Lecture 1 to 5

Previously on

CS1010..

Functions

A C program is a collection of functions.

A function is a black box with input(s) and output

```
long square(long x)
{
  return x*x;
}

double hypotenuse_of(long base, long height)
{
  return sqrt(square(base) + square(height));
}

int main()
{
   :
  hypotenuse = hypotenuse_of(base, height);
   :
}
```

```
long square(long x)
{
  return x*x;
}
```

function

```
double hypotenuse_of(long base, long height)
{
  return sqrt(square(base) + square(height));
}
```

function

```
int main()
{
   :
  hypotenuse = hypotenuse_of(base, height);
   :
}
```

function

```
long square(long x)
{
  return x*x;
}
```

```
double hypotenuse_of(long base, long height)
{
  return sqrt(square(base) + square(height));
}
```

```
double hypotenuse = hypotenuse_of(base, height);
```

not a function

```
int main()
{
  :
}
```

```
long square(long x)
{
  return x*x;
}
```

```
double hypotenuse_of(long base, long height)
{
  return sqrt(square(base) + square(height));
}
```

```
double hypotenuse;
```

not
a function

```
int main()
{
  hypotenuse = hypotenuse_of(base, height);
}
```

No function within a function

```
int main()
{
  double hypotenuse_of(long base, long height)
  {
    long square(long x)
    {
      return x*x;
    }
    return sqrt(square(base) + square(height));
  }

  double hypotenuse = hypotenuse_of(base, height);
}
```

Not a “black box”

```
long square(long x)
{
  return x*x;
}
```

function

```
double hypotenuse_of(long base, long height)
{
  hypotenuse = sqrt(square(base) + square(height));
}
```

function

```
int main()
{
  double hypotenuse;
  hypotenuse_of(base, height);
    :
}
```

function

example code in class

```
x = cs1010_read_long();
```

what (some of) you wrote

```
cs1010_read_long(x);
x = cs1010_input_long();
:
```

example code in class

```
cs1010_print_long(x);
```

what (some of) you wrote

```
cs1010_print_long(long x);
cs1010_print_long(“x”);
:
```

learn by example

and

“imitate” other code

when in doubt, see how it is done in the notes

Previously on

CS1010..

Lecture 4

4 September 2018

Admin Matters
Unit 8: **If Else**
Unit 9: **Logical Expression**
Unit 10: **Assertion**

The `bool` data type can take two values `true` or `false`

```c
#include <stdbool.h>

bool is_gen_z(long year)
{
  return ((year >= 1995) && (year <= 2005));
}

bool is_not_gen_z(long year)
{
  return ((year < 1995) || (year > 2005));
}
```

&&

||

!

Short-circuiting

```
a && b
```

```
a || b
```

```
if (number < 100000 && is_prime(number)) {
    .
}
```

```
if (is_prime(number) && number < 100000) {
    .
}
```

De Morgan's Law

```
!(e1 && e2)
```

is the same as

```
(!e1) || (!e2)
```

```
!(e1 || e2)
```

is the same as

```
(!e1) && (!e2)
```

```
!((year >= 1995) && (year <= 2005))
```

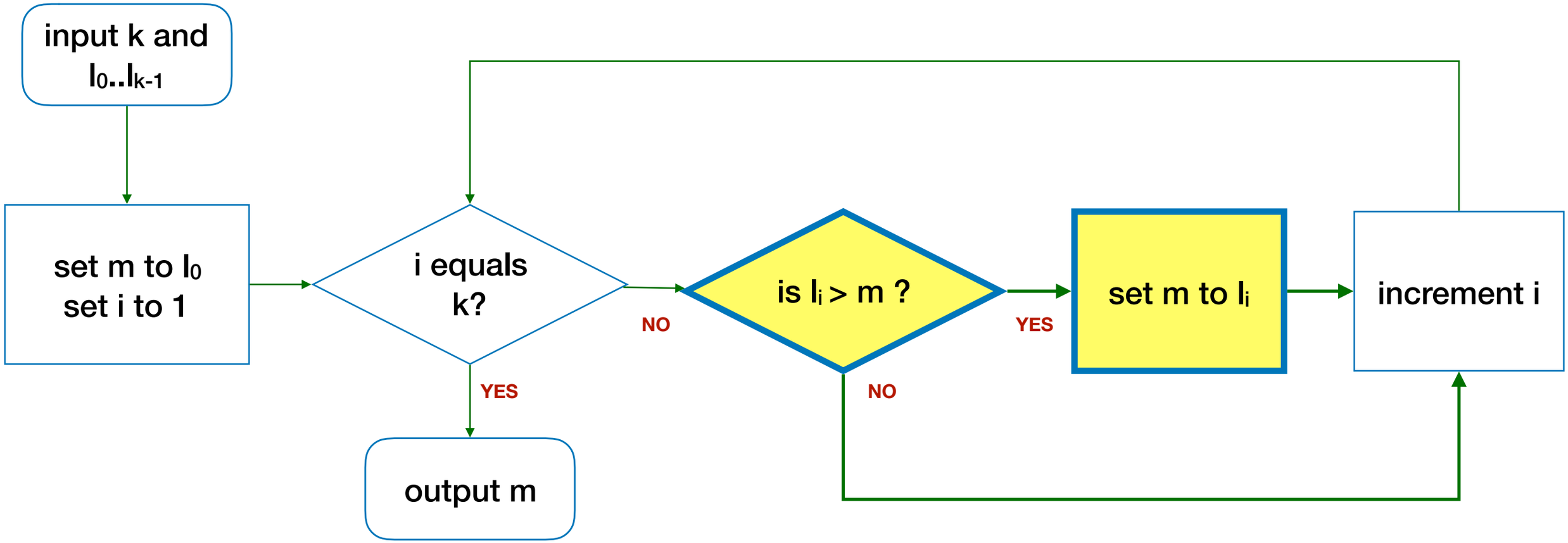
is the same as

4

questions for writing loops

1.

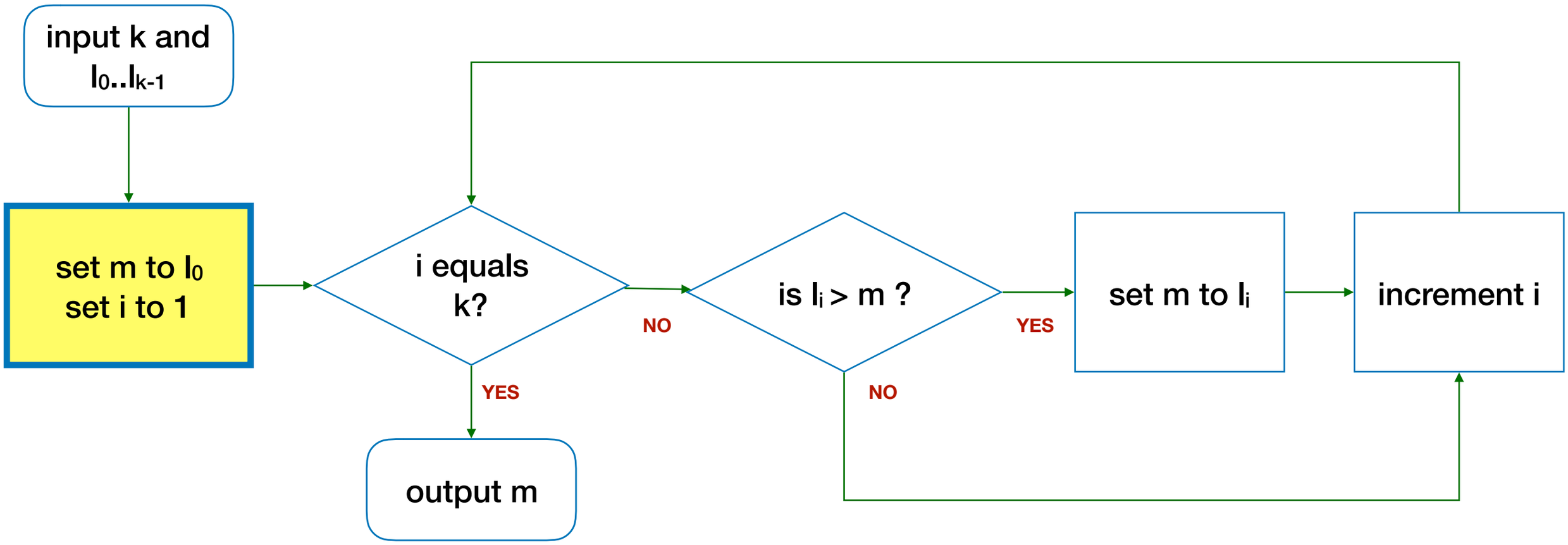
what do we want to repeat?



what do we want to repeat?

2.

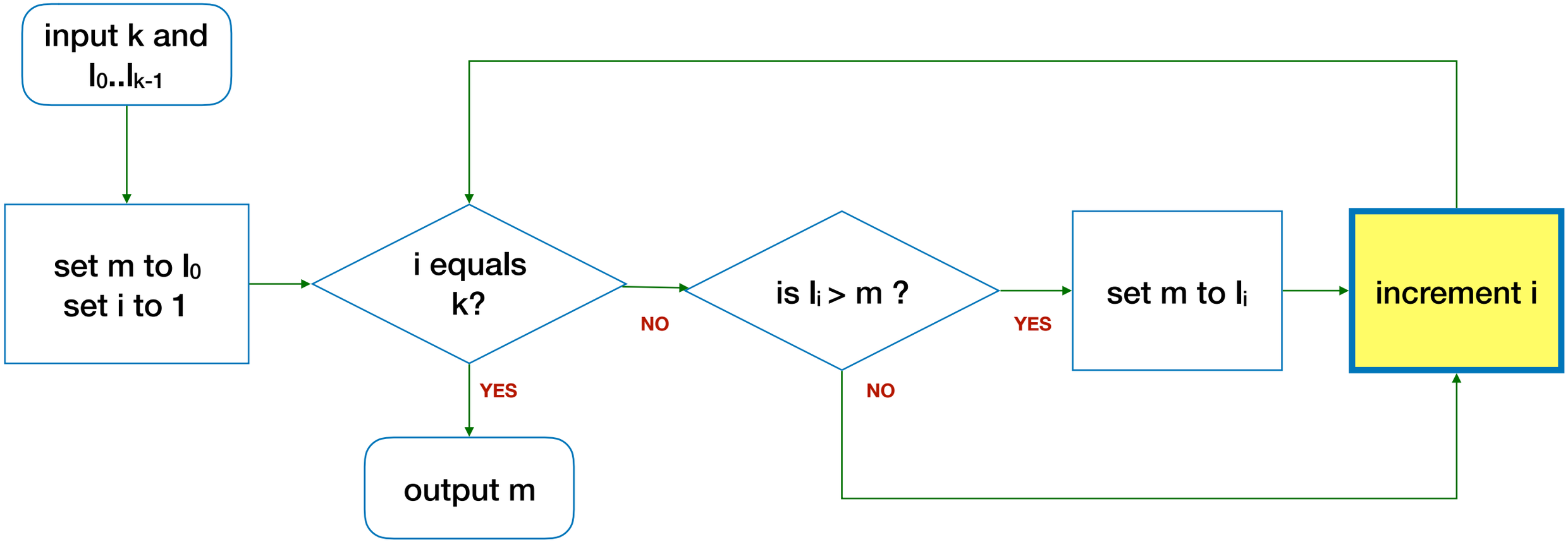
what do we need to set up before repeating?



what do we need to set up?

3.

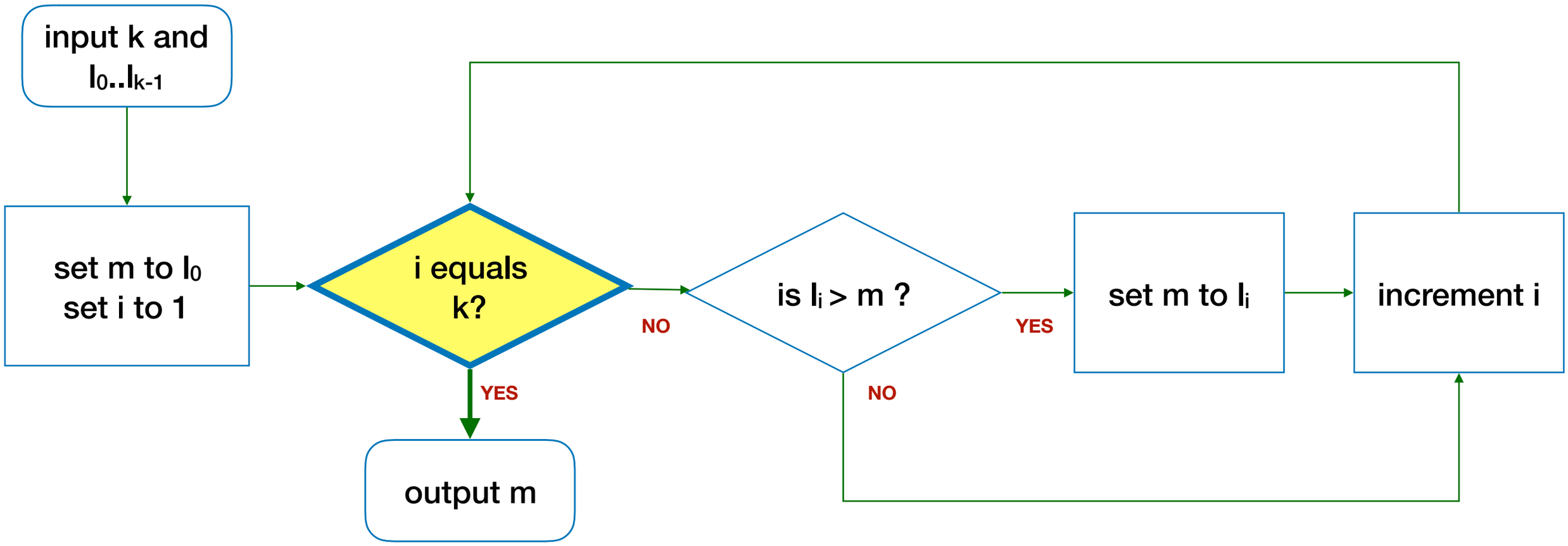
what changes from one repetition to another?



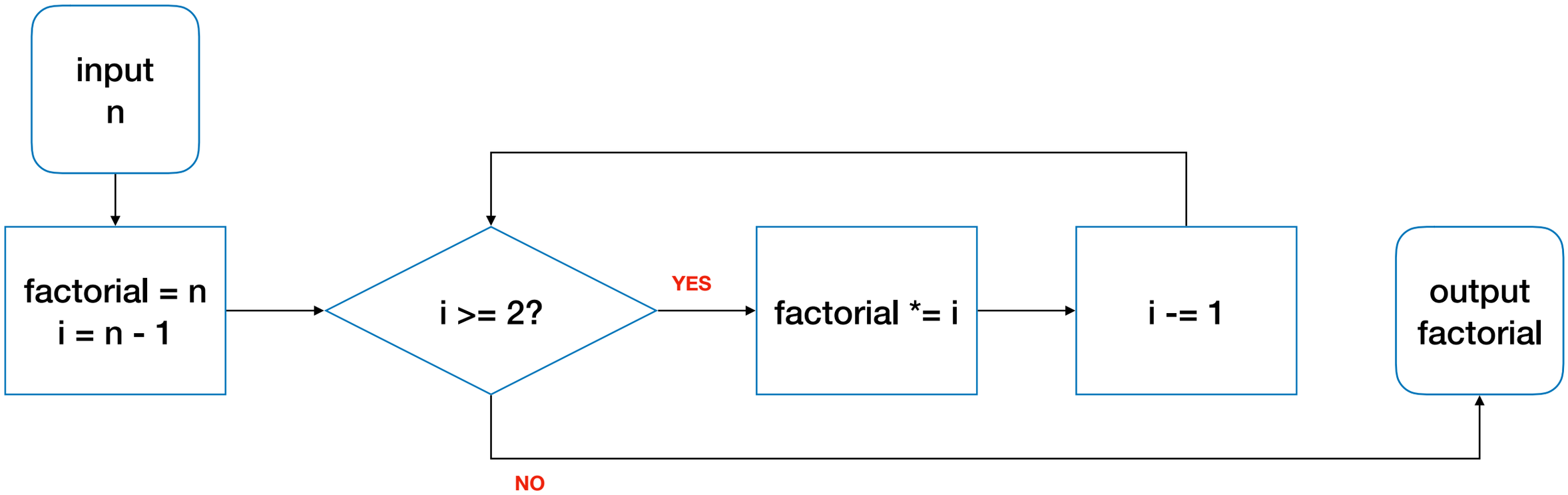
what changes?

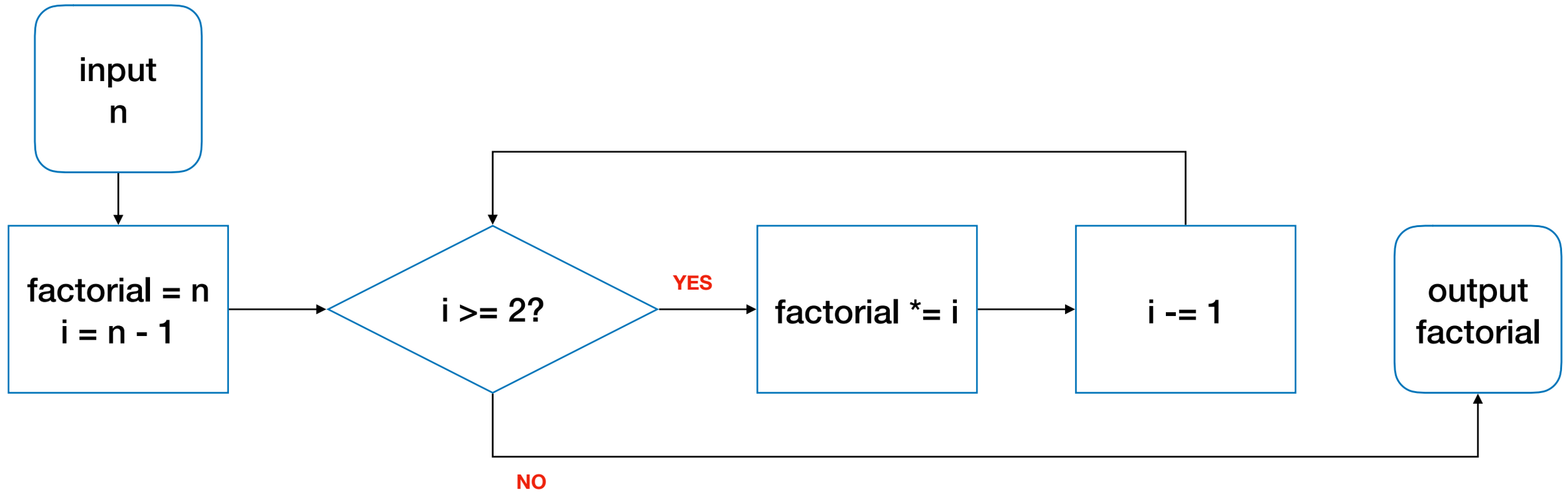
4.

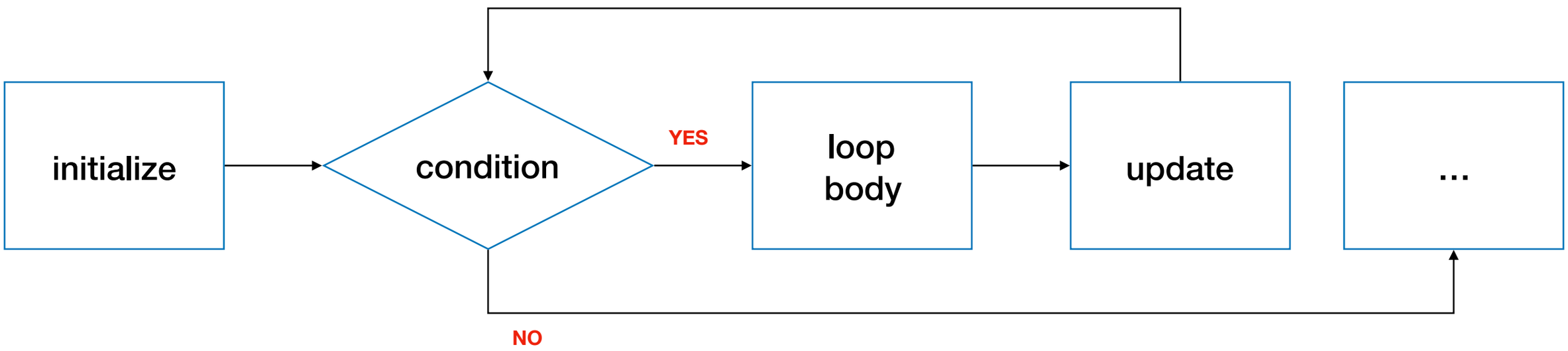
when to stop repeating?

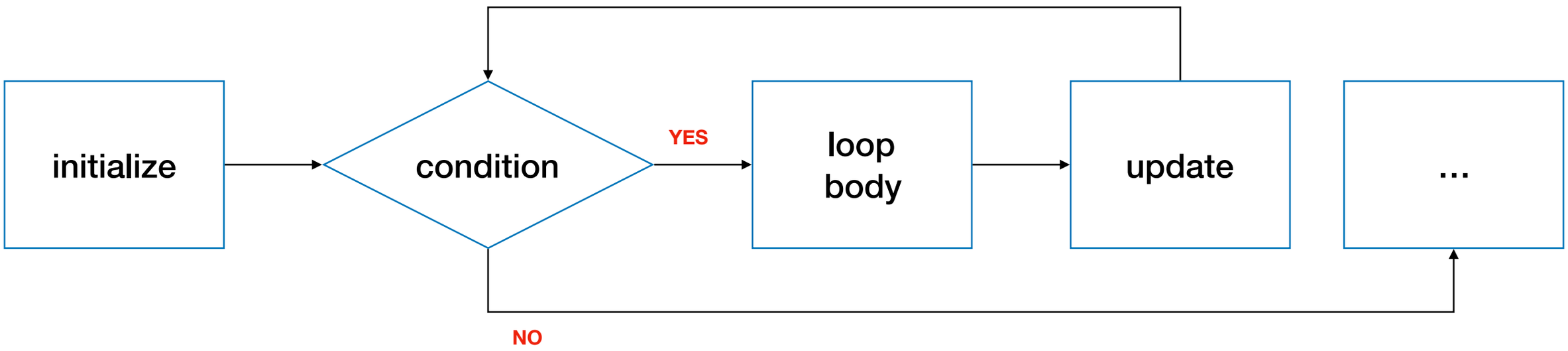

input k and
$l_0..l_{k-1}$
set m to l_0
set i to 1
i equals
k?
YES
output m
NO
is l_i > m ?
NO
YES
set m to l_i
increment i

when to stop?

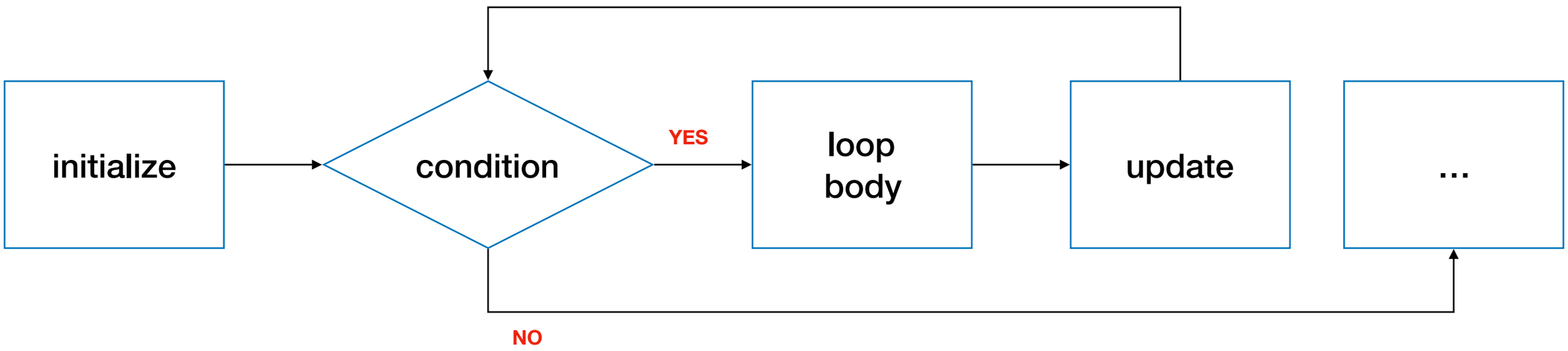

input
n
factorial = n
i = n - 1
i >= 2?
YES
NO
factorial *= i
i -= 1
output
factorial


input
n
factorial = n
i = n - 1
i >= 2?
YES
NO
factorial *= i
i -= 1
output
factorial

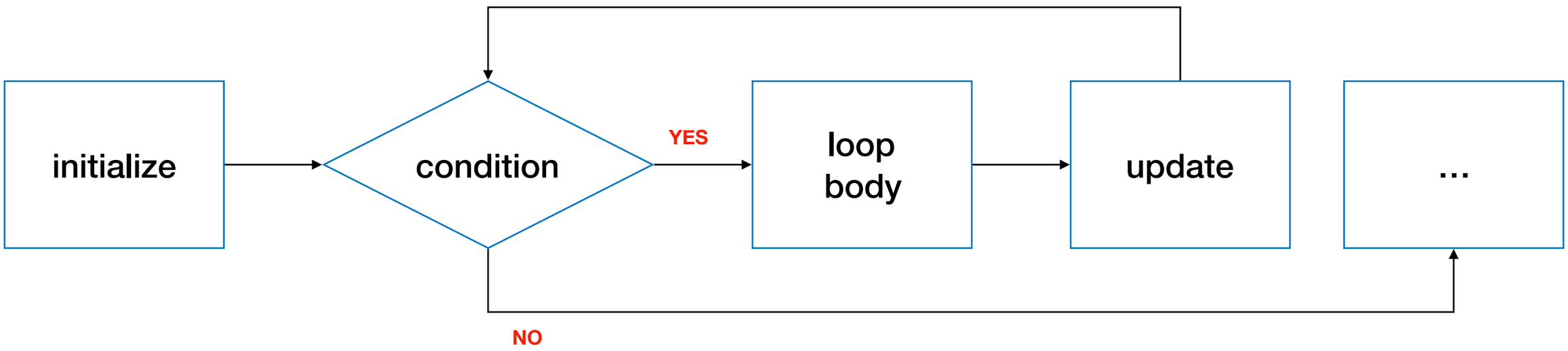

initialize
condition
YES
NO
loop
body
update
...



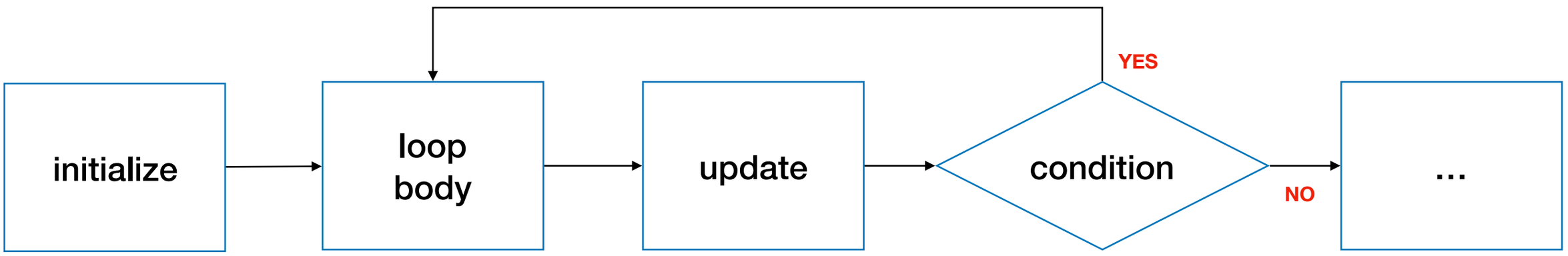

```
for (<initialize>; <condition>; <update>) {
    <body>
}
```



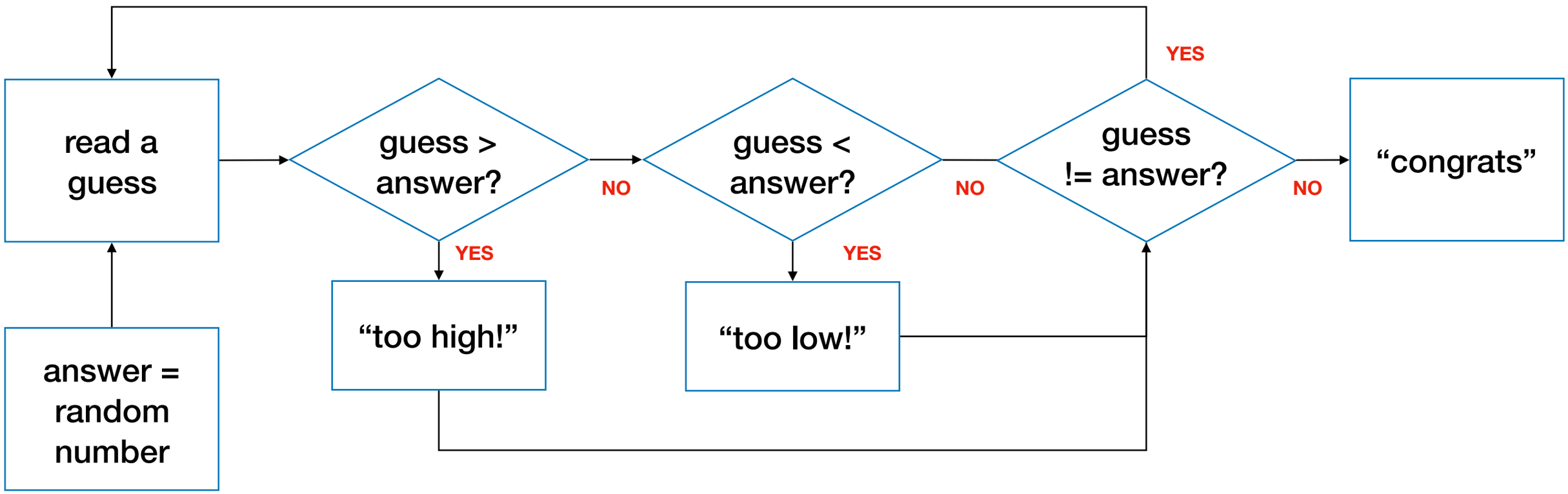

```
while (<condition>) {
      <body>
}
```




```
<initialize>
while (<condition>) {
    <body>
    <update>
}
```




```
<initialize>
do {
    <body>
    <update>
} while (<condition>);
```


read a guess
guess > answer?
NO
guess < answer?
NO
guess != answer?
NO
“congrats”
YES
YES
YES
“too high!”
“too low!”
answer = random number

Assertion

// { … }

Expression that must be true at a given point in the program.

```
long x = 1;
// { x == 1 }
```

(obvious)

```
x = 3/2;
// { x == 1.5 }
```

(maybe not so obvious)