

# Control Structures

# Control Statements

`if`

`for`

`while`

# Control Statements

`if`

**`for`**

`while`

This is called the **initialization statement** and is performed before the loop starts.

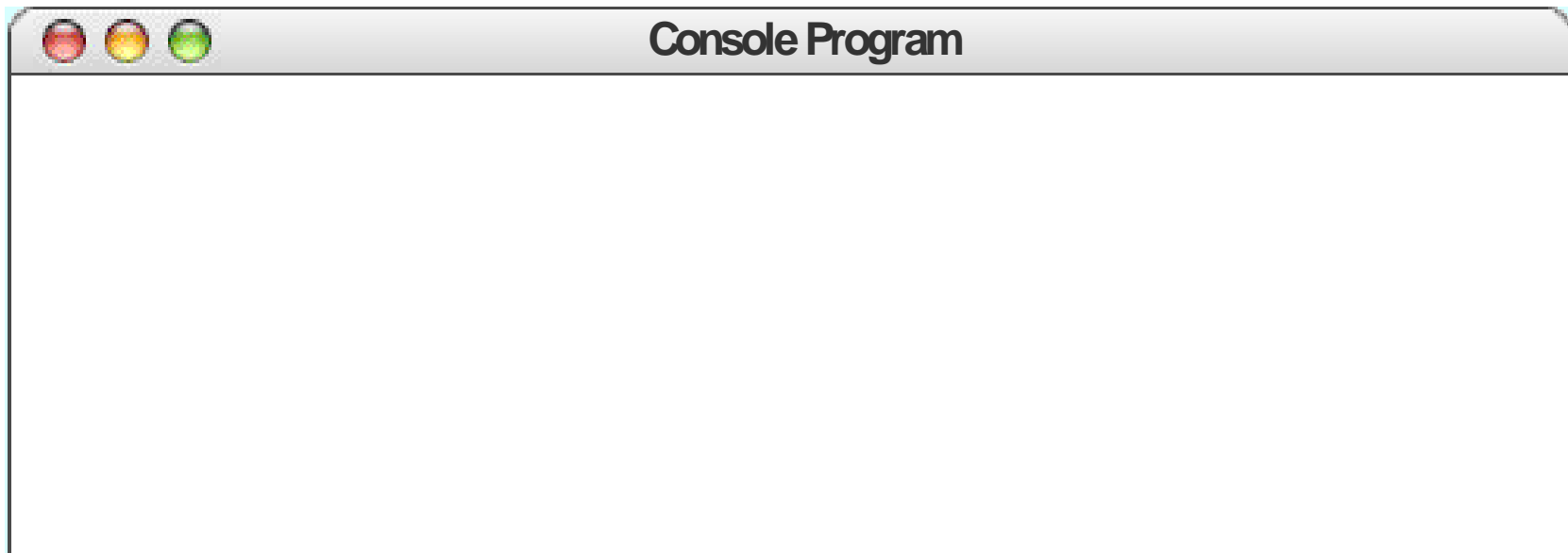
This is called the **step** or **increment** and is performed at the end of each loop iteration.

```
for (int i = 0; i < N; i++) {  
    ...  
}
```

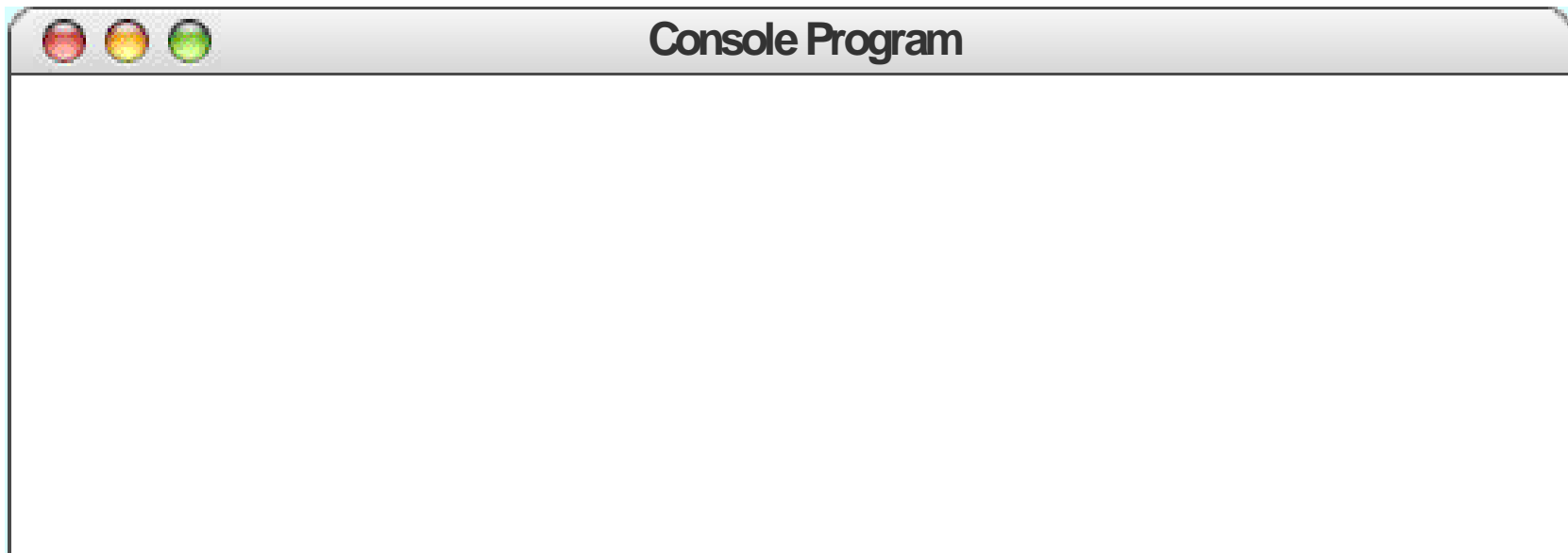
This is called the **loop condition** or **termination condition**. The loop will check whether this statement is true before each iteration of the loop.

```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```



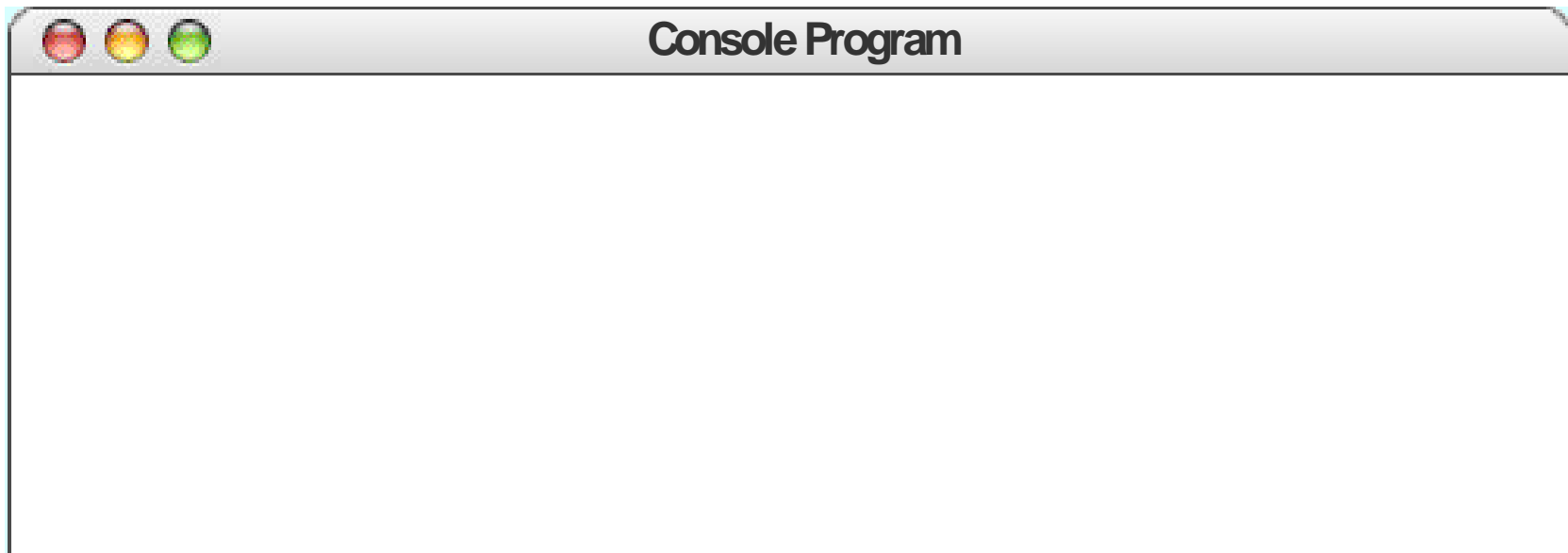
```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i
```

0

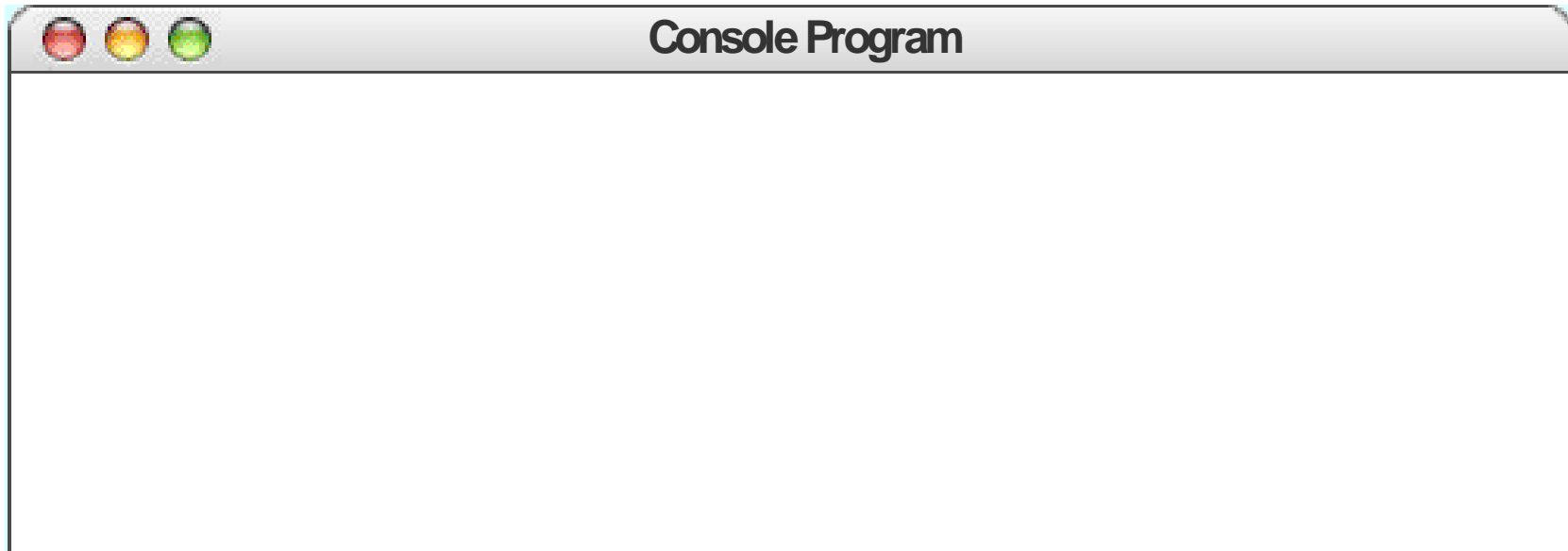




```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i
```

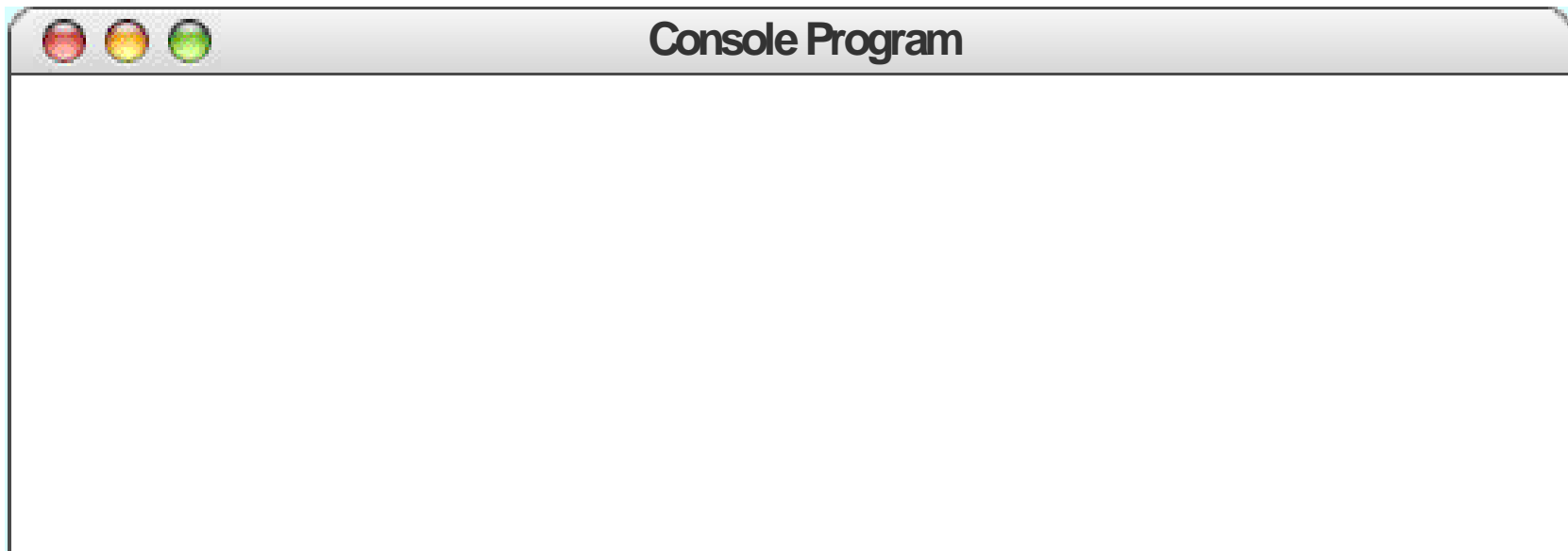
0



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i
```

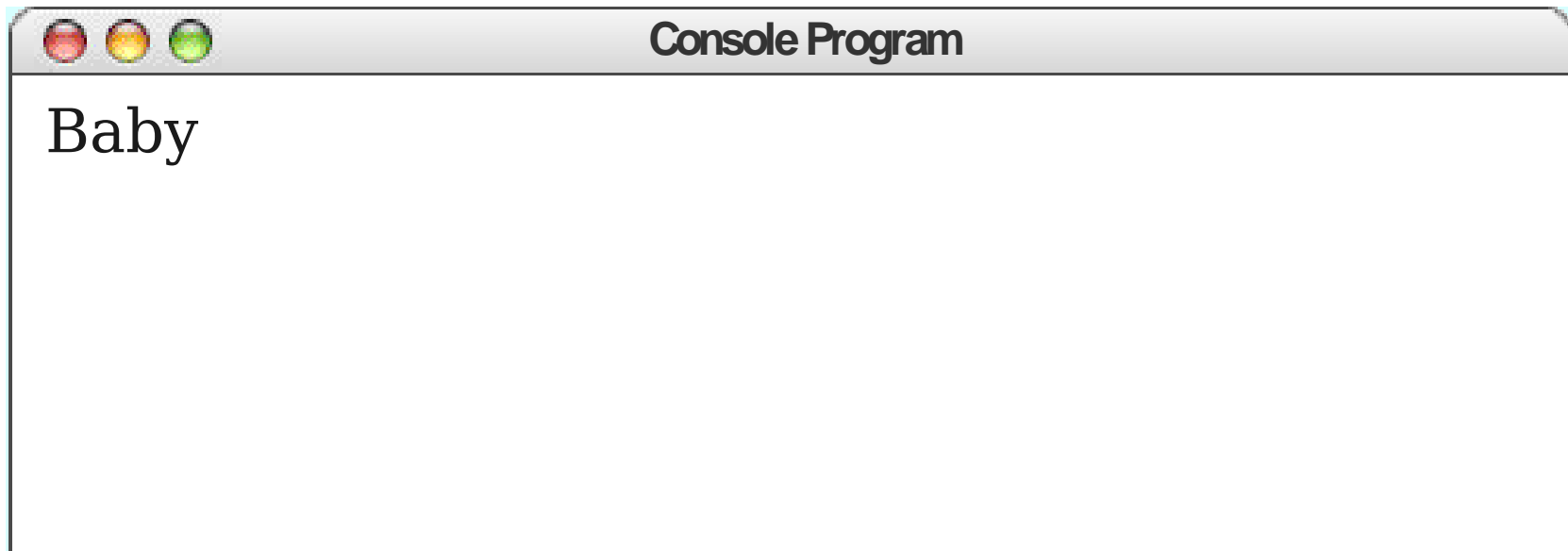
0



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i
```

0

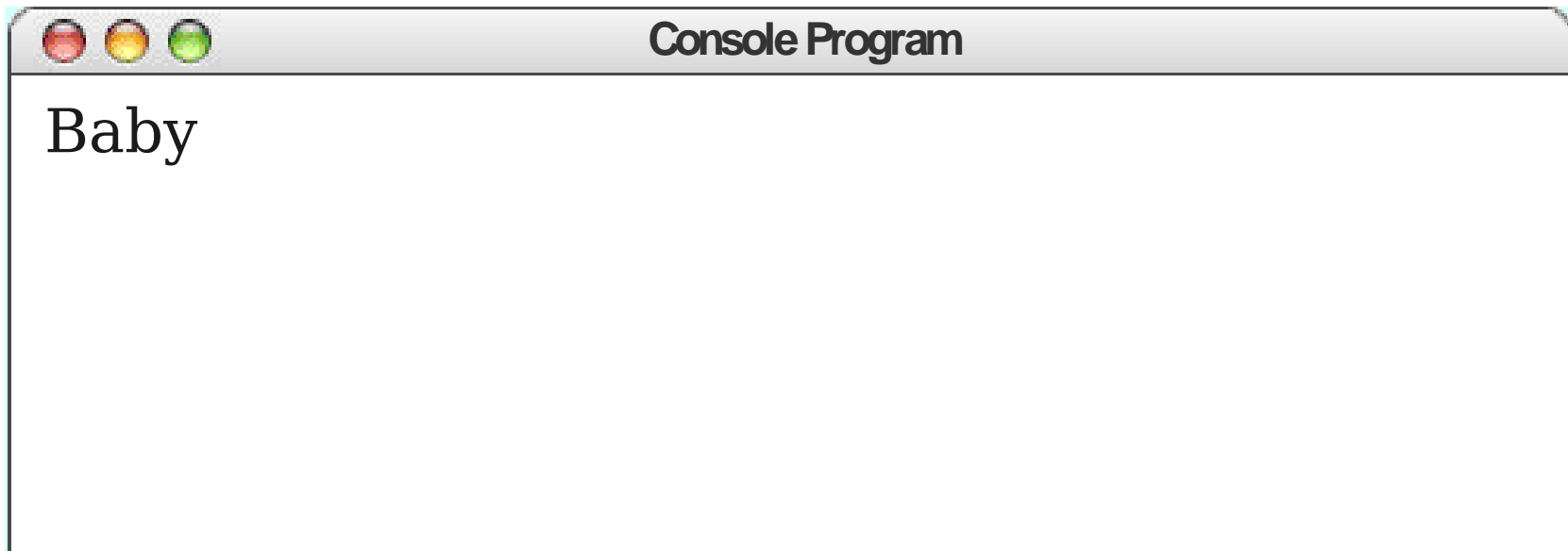


```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i
```



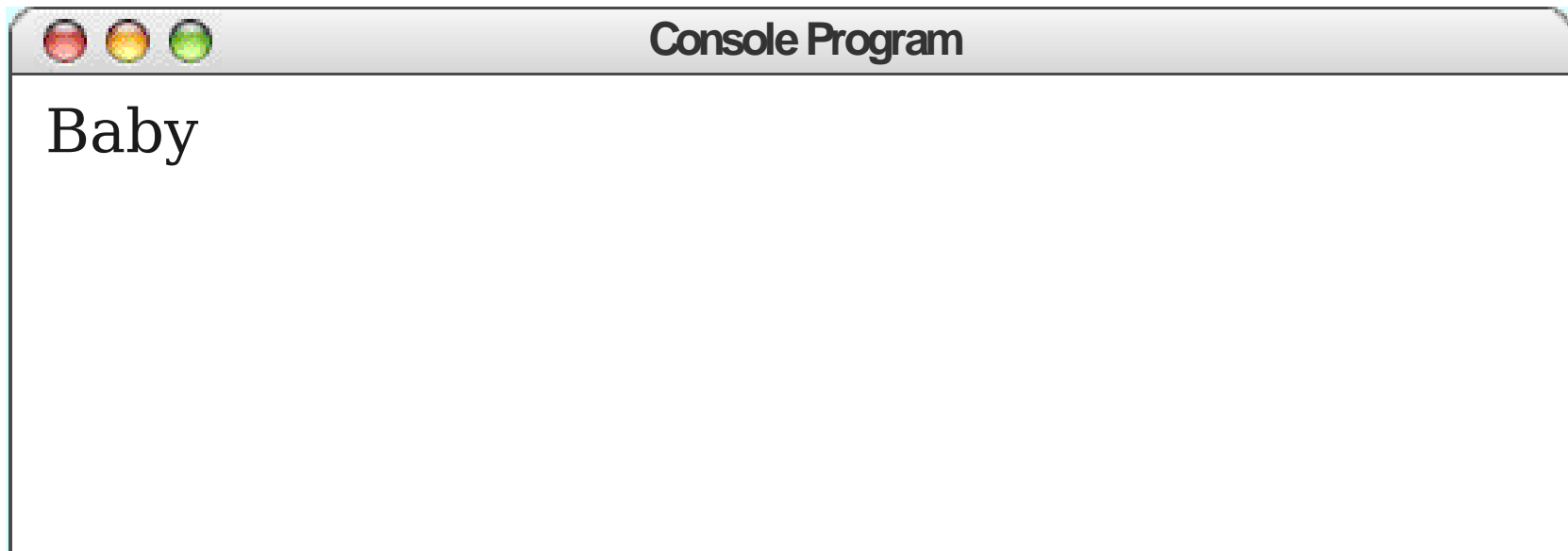
0



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i
```

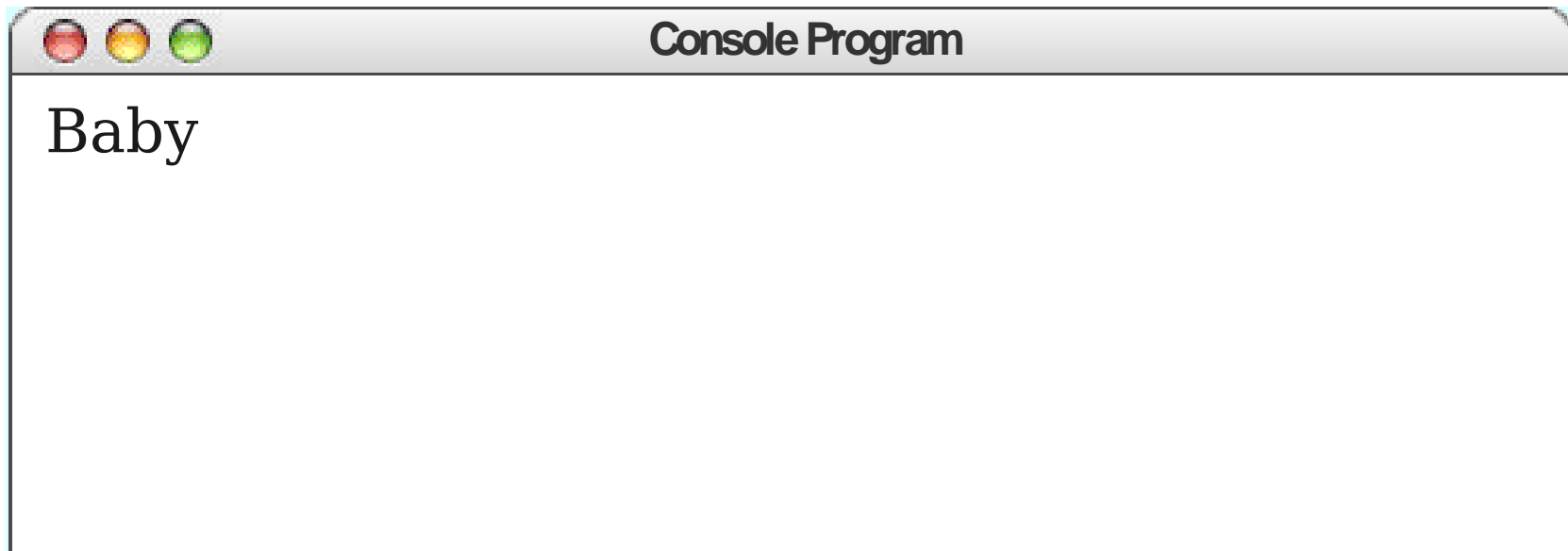
1



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i
```

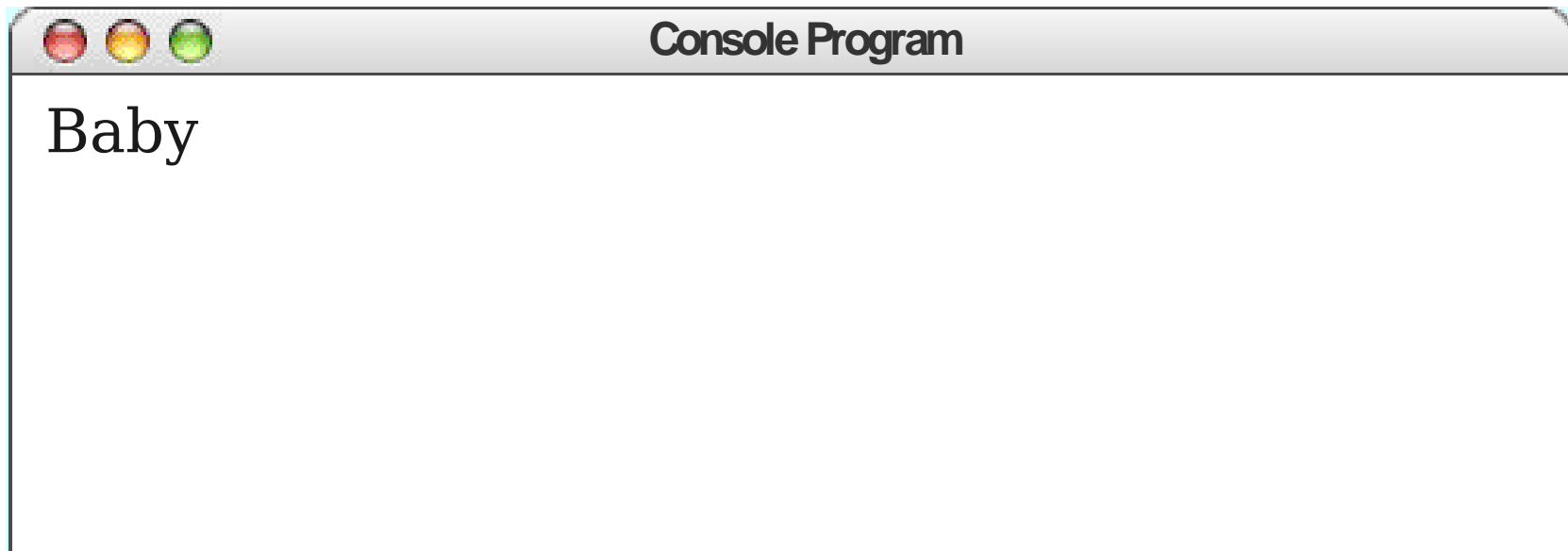
1



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i
```

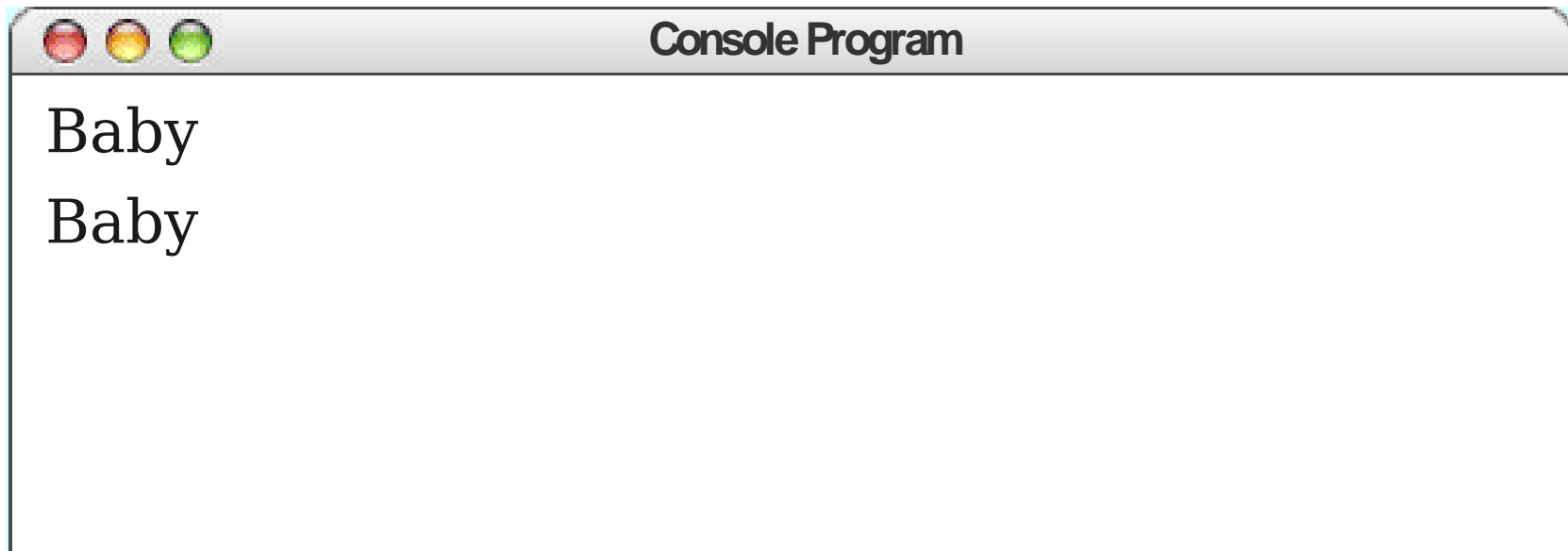
1



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i
```

1

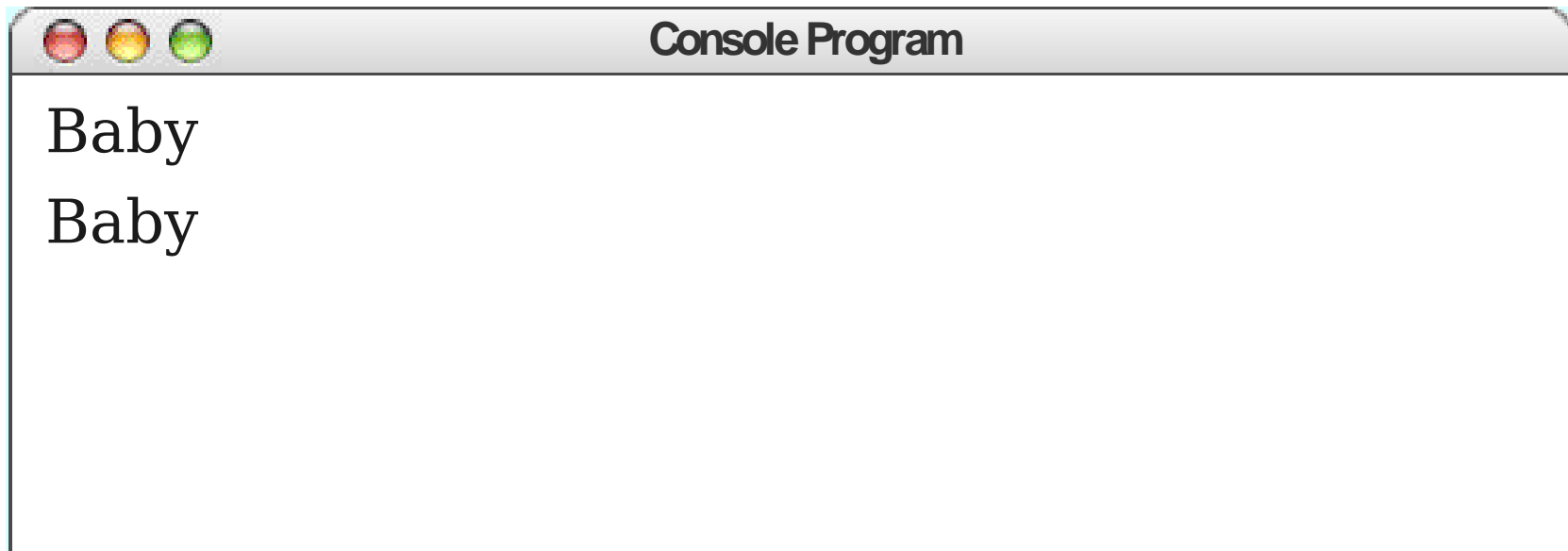




```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

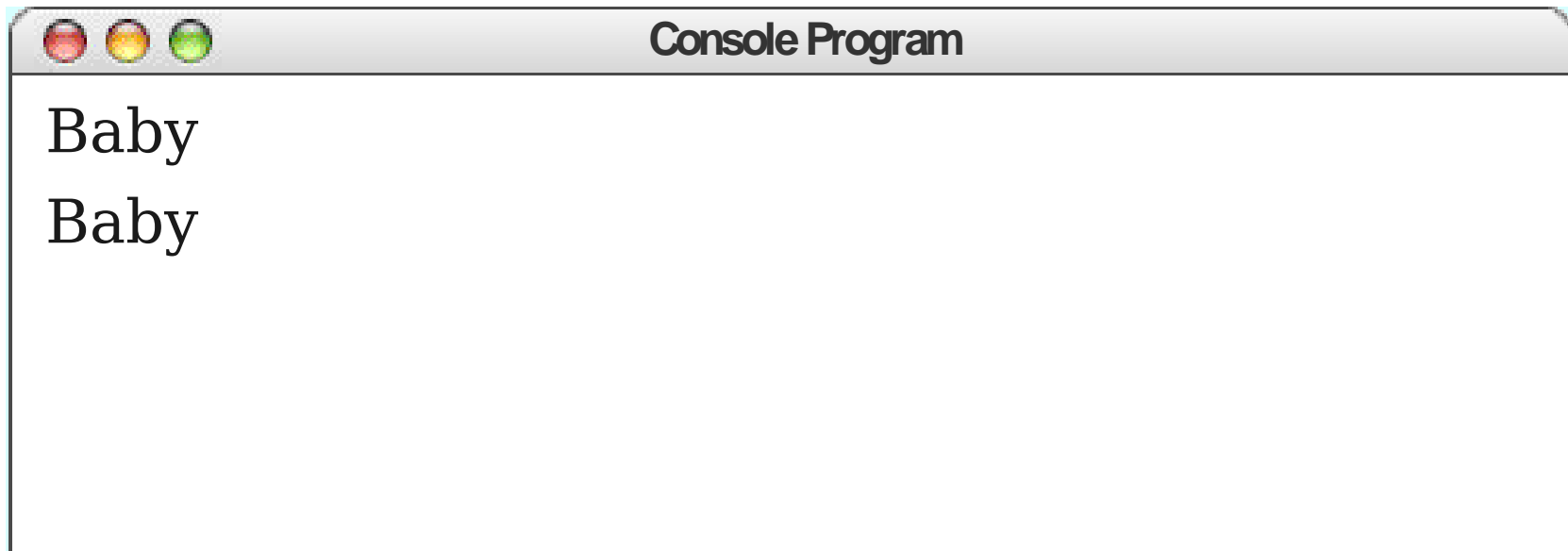
```
int i
```

1



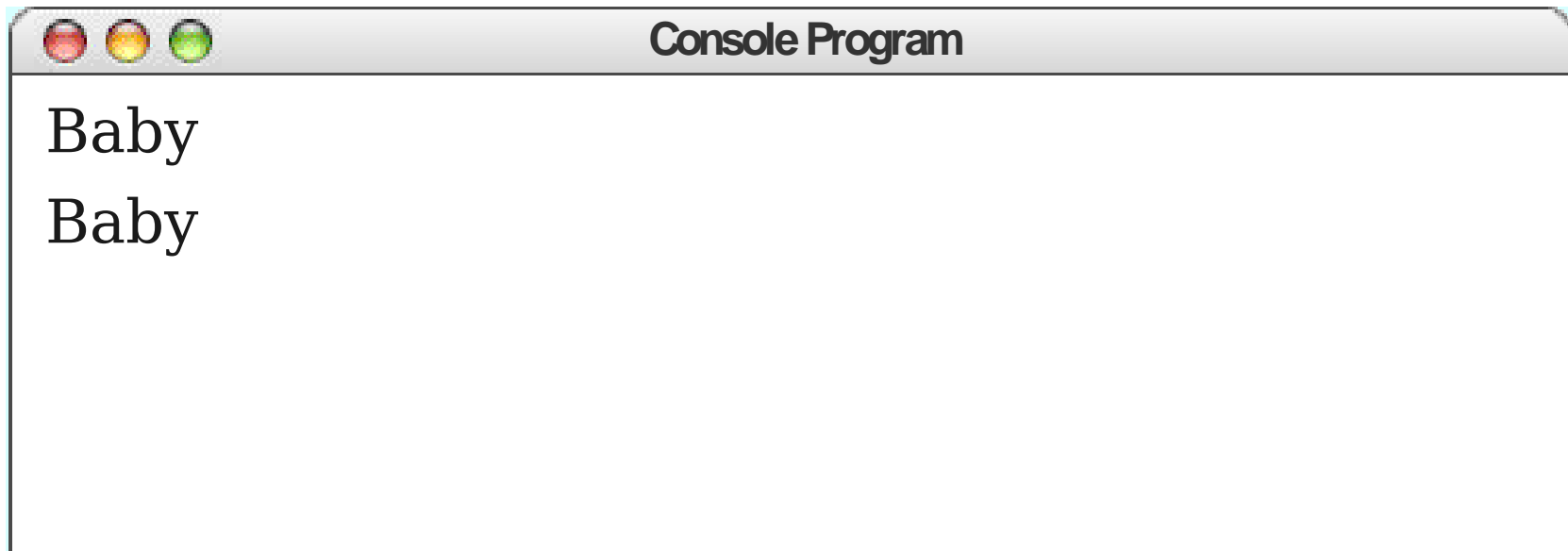
```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i 2
```



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

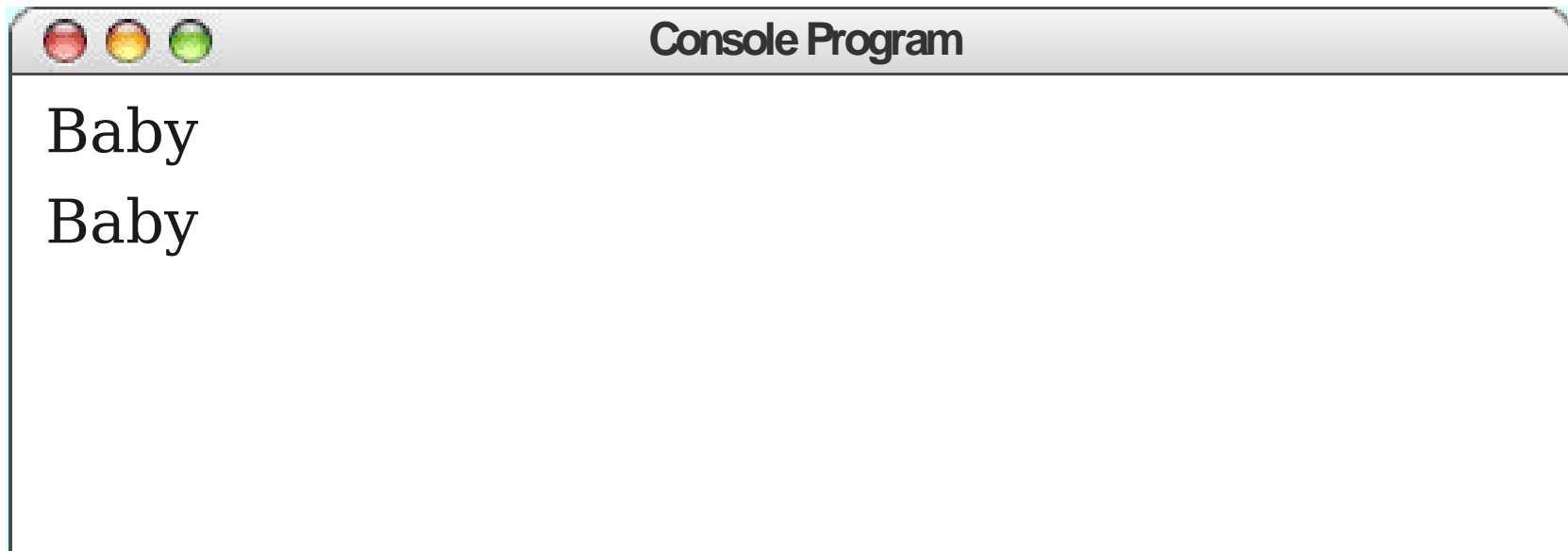
```
int i 2
```



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

int i

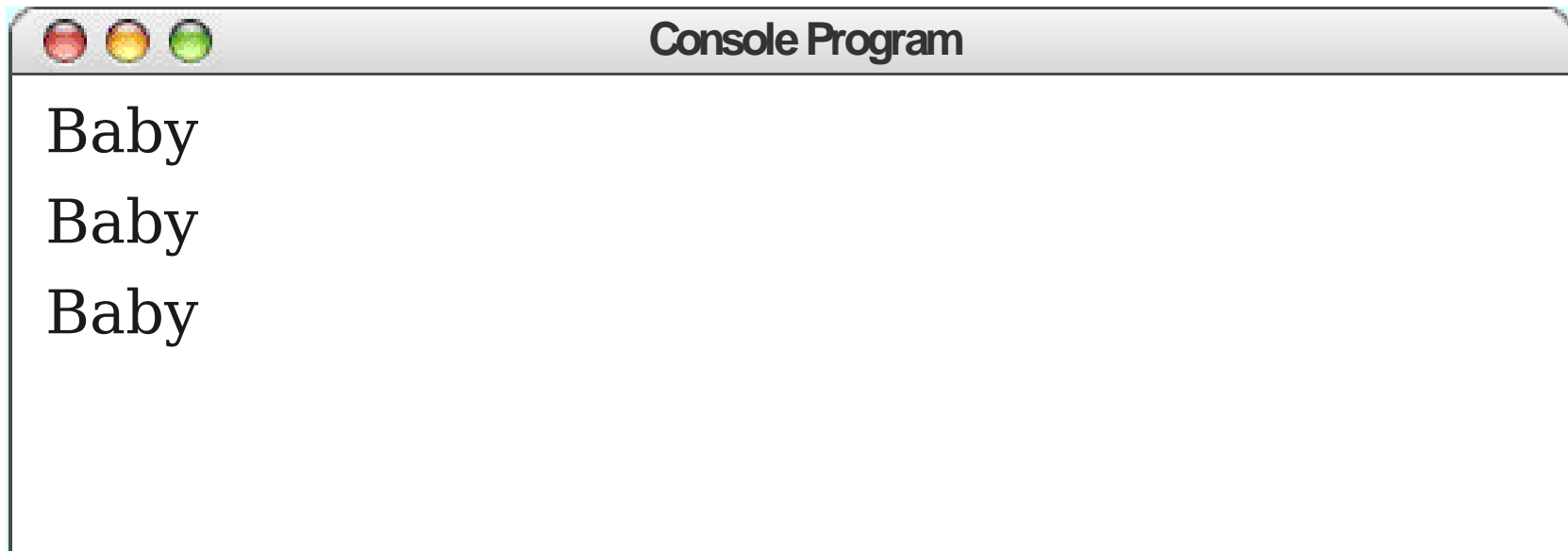
2



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

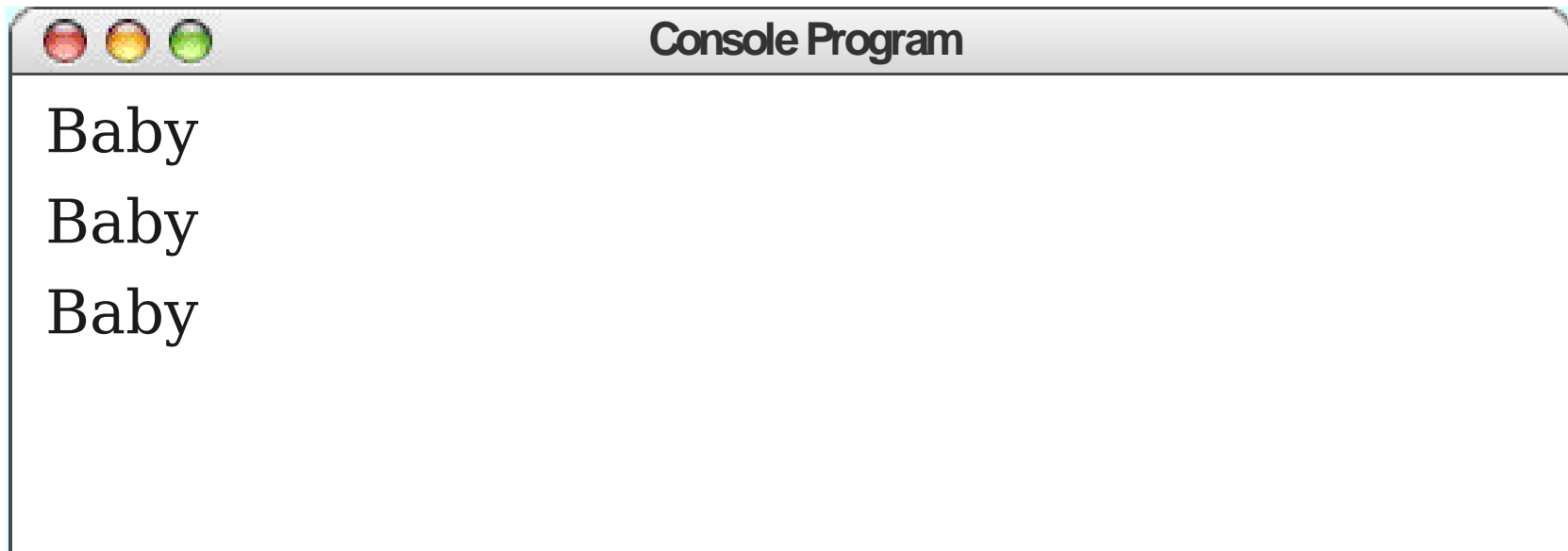
int i

2



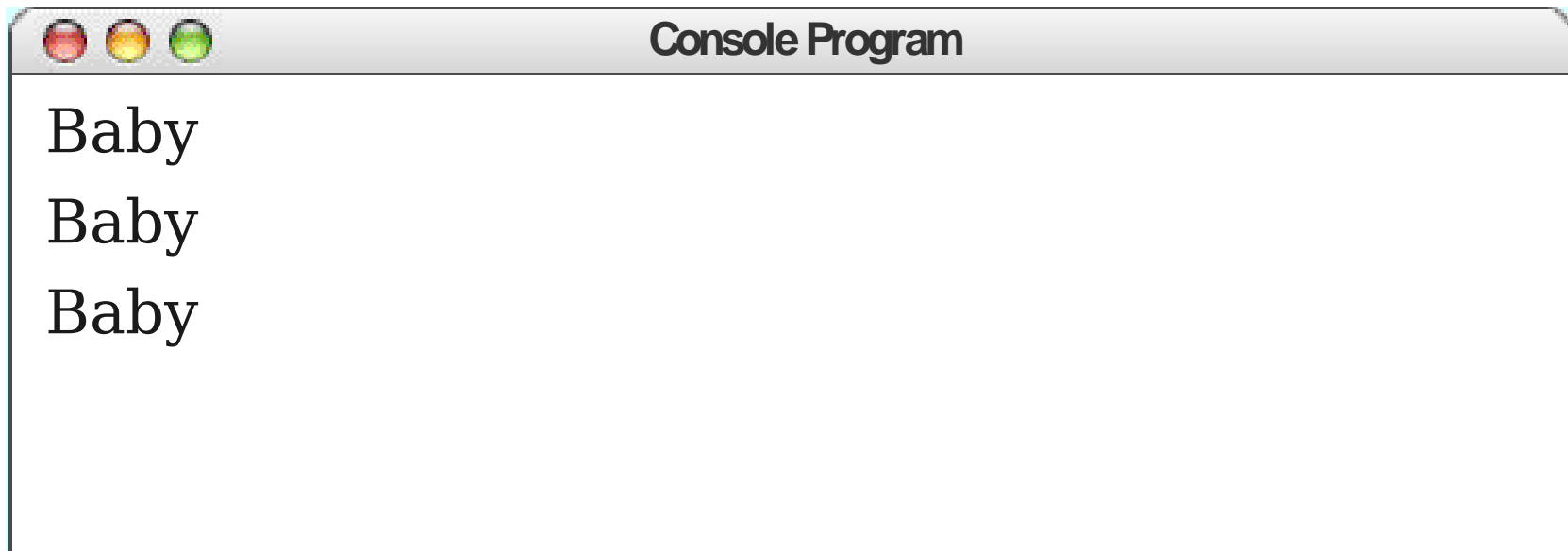
```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i 2
```



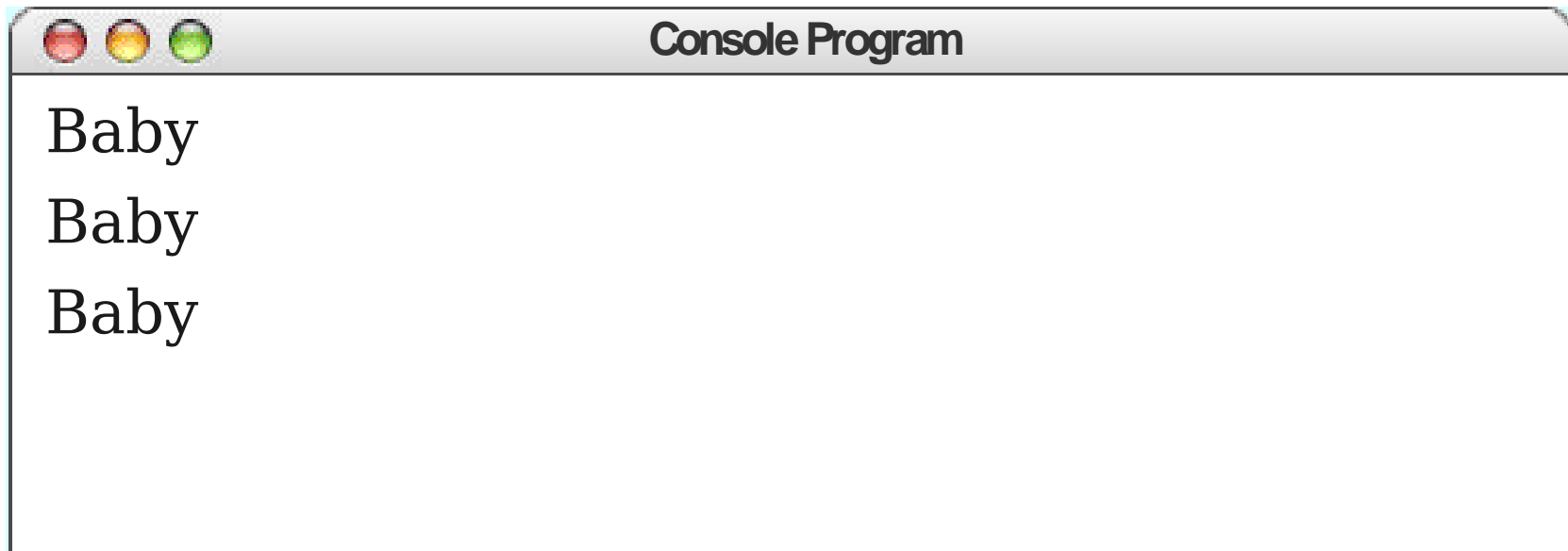
```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i 3
```



```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

```
int i 3
```

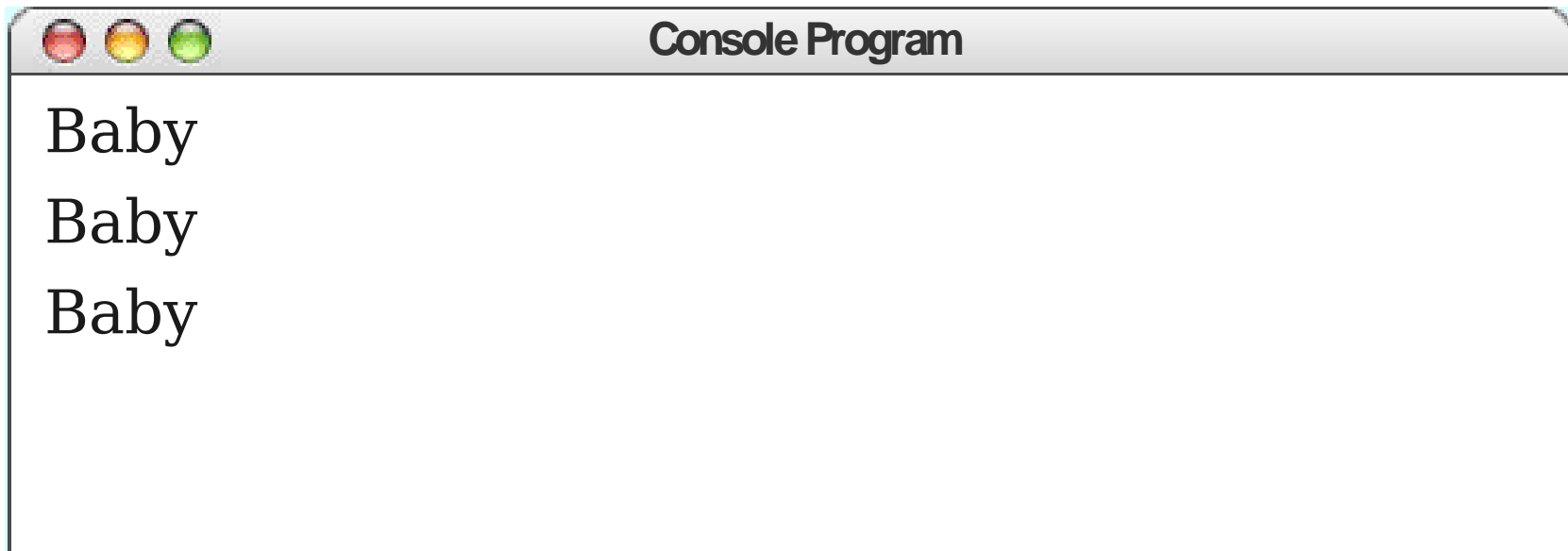




```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}
```

```
println("Whoa");
```

```
int i
```



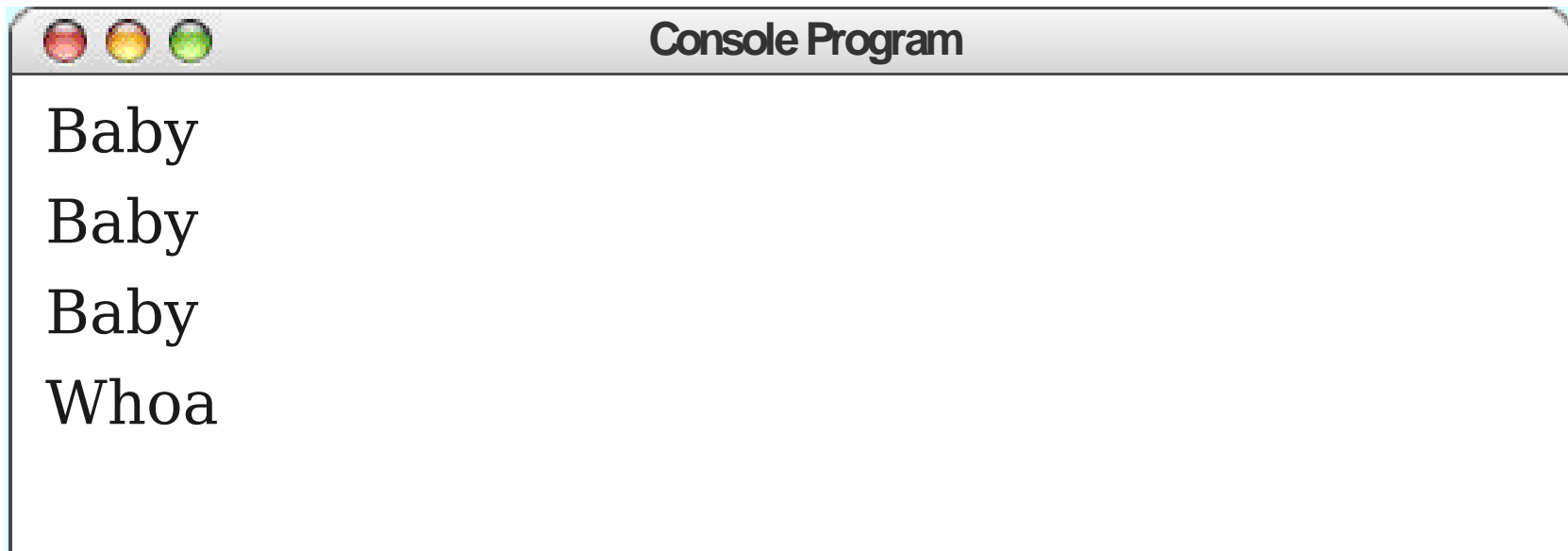
```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}
```

```
println("Whoa");
```

```
int i
```



3

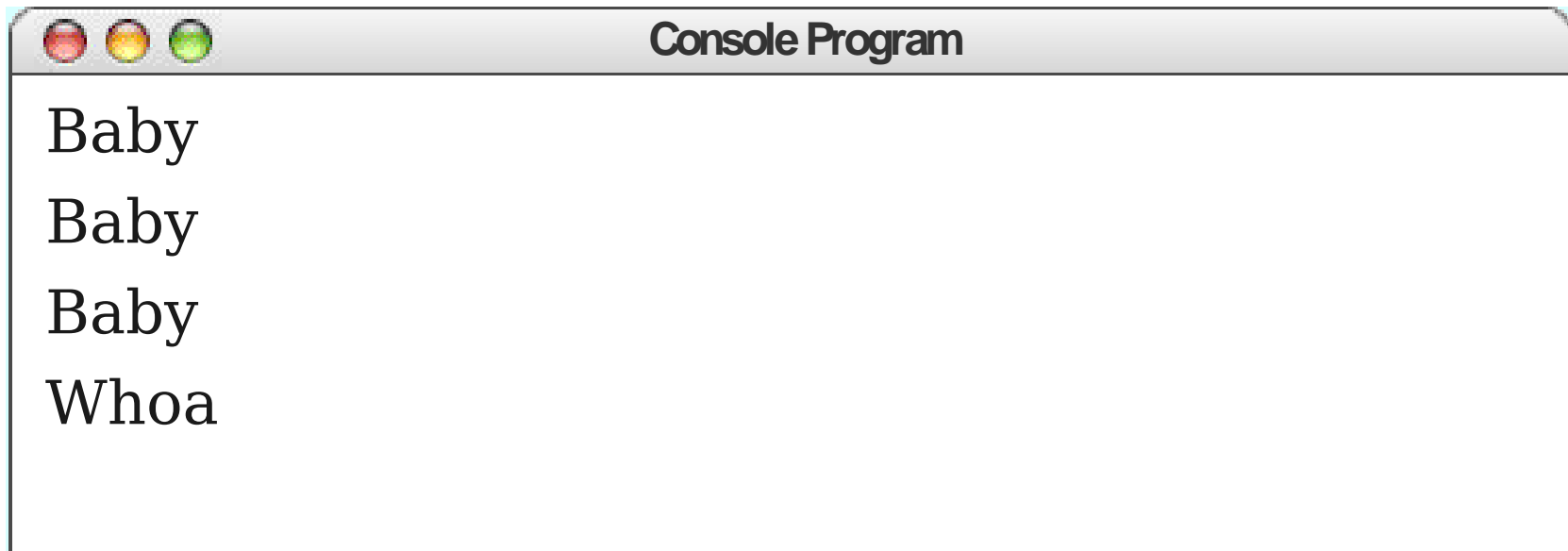


```
Console Program  
Baby  
Baby  
Baby  
Whoa
```

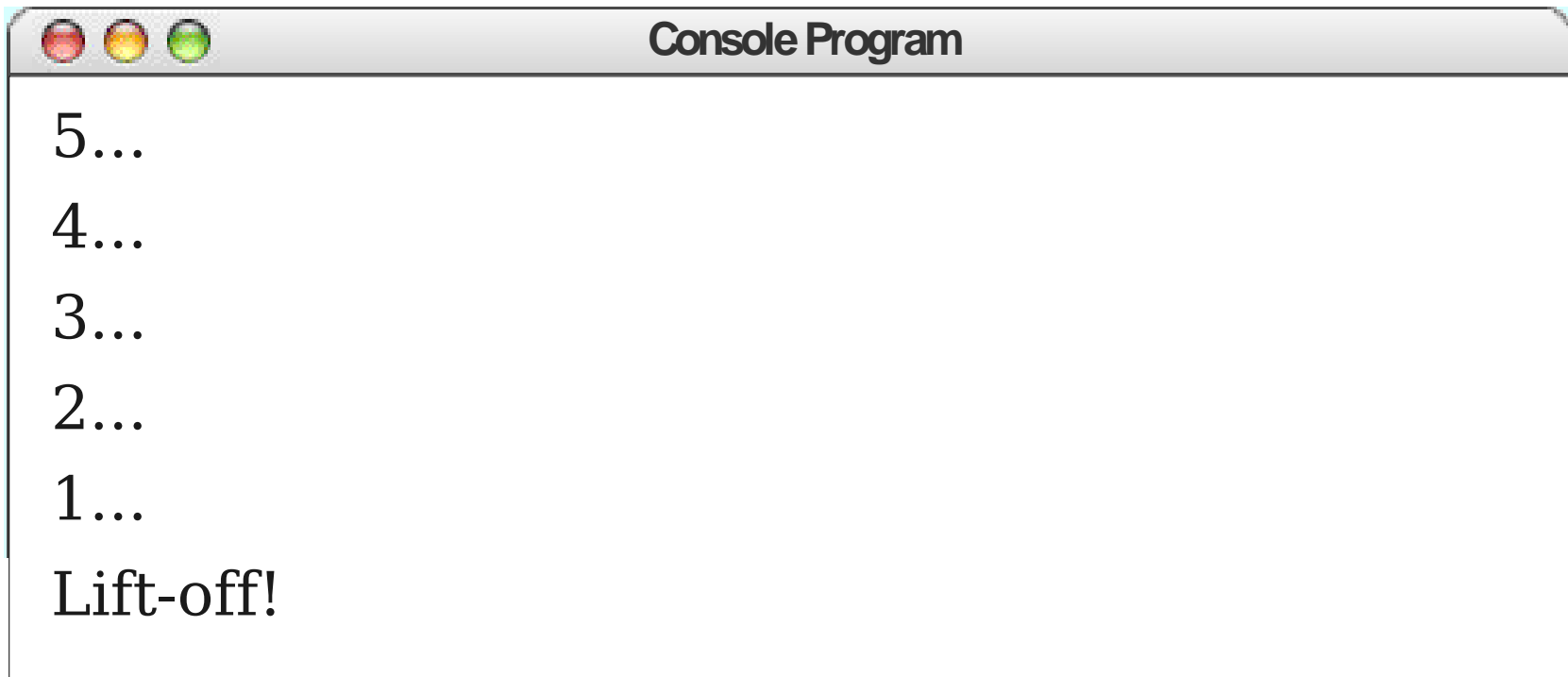
```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Whoa");
```

int i

3



```
for (int i = 5; i > 0; i--) {  
    println(i + "...");  
}  
println("Lift-off!");
```



Console Program

5...  
4...  
3...  
2...  
1...  
Lift-off!

# An Actual Rocket Launch

**<http://www.youtube.com/watch?v=ShRa2RG2KDI>**

(Start at 9:25)

T-31 Seconds: Handoff has occurred  
T-17 Seconds: Firing chain armed  
T-10 Seconds: Hydrogen burn  
T-0 Seconds: Liftoff!

```
for (int i = 40; i > 0; i--) {  
    println("T-" + i + "...");  
}  
println("Lift-off!");
```

```
public void run() {
    /* Do the launch countdown! */
    for (int i = 40; i > 0; i--) {
        println("T-" + i + " seconds.");

        /* Specific mission commands. */
        if (i == 31) {
            println("Handoff has occurred.");
        } else if (i == 17) {
            println("Firing chain is armed.");
        } else if (i == 10) {
            println("Hydrogen burn.");
        }
    }

    println("Lift-off!");
}
```

```
public void run() {
    /* Do the launch countdown! */
    for (int i = 40; i > 0; i--) {
        println("T-" + i + " seconds.");

        /* Specific mission commands. */
        if (i == 31) {
            println("Handoff has occurred.");
        } else if (i == 17) {
            println("Firing chain is armed.");
        } else if (i == 10) {
            println("Hydrogen burn.");
        }
    }

    println("Lift-off!");
}
```



```
public void run() {
    /* Do the launch countdown! */
    for (int i = 40; i > 0; i--) {
        println("T-" + i + " seconds.");

        /* Specific mission commands. */
        if (i == 31) {
            println("Handoff has occurred.");
        } else if (i == 17) {
            println("Firing chain is armed.");
        } else if (i == 10) {
            println("Hydrogen burn.");
        }
    }

    println("Lift-off!");
}
```

# Magic Numbers

- A **magic number** is a number written in a piece of code whose meaning cannot easily be deduced from context.

```
double weight = 9.8 * (mass - 14.3);
```

- Magic numbers are a Bad Thing; they make code harder to read.

# Constants

- Not all variables actually *vary*.
- A **constant** is a name for a value that never changes.
- Syntax (defined outside of any method):  
`private static final type name = value;`
- By convention, constants are named in **UPPER\_CASE\_WITH\_UNDERSCORES** to differentiate them from variables.

# General Rules for Constants

- You can usually use 0 and 1 in loops without needing constants.
- When computing averages, it's fine to just use the number 2.
- Anything more complex than this should probably be made into a constant.

**Time-Out for Announcements!**

Friday Four Square!  
Today at 4:15PM, Outside Gates

# Announcements

- Programming Assignment #1 due right now.
  - Due on next Wednesday if using a late day.
  - LaIR closed on Sunday but open on Monday.
- Email due on Sunday night.
- **Programming Assignment #2** out today, due Friday, January 31 at 3:15PM.
  - Play around with Java statements and control structures!
  - Make some pretty pictures!
  - Explore your creative potential!

# Casual Dinner for Women in CS

- Next **Wednesday, January 22** at 6:00PM on the fifth floor of the Gates building.
- Good food, great company, and everyone is invited!
- RSVP through email link (sent out Tuesday).



# Control Statements

`for`

`if`

`while`

# Control Statements

for

if

**while**

# The `while` Loop

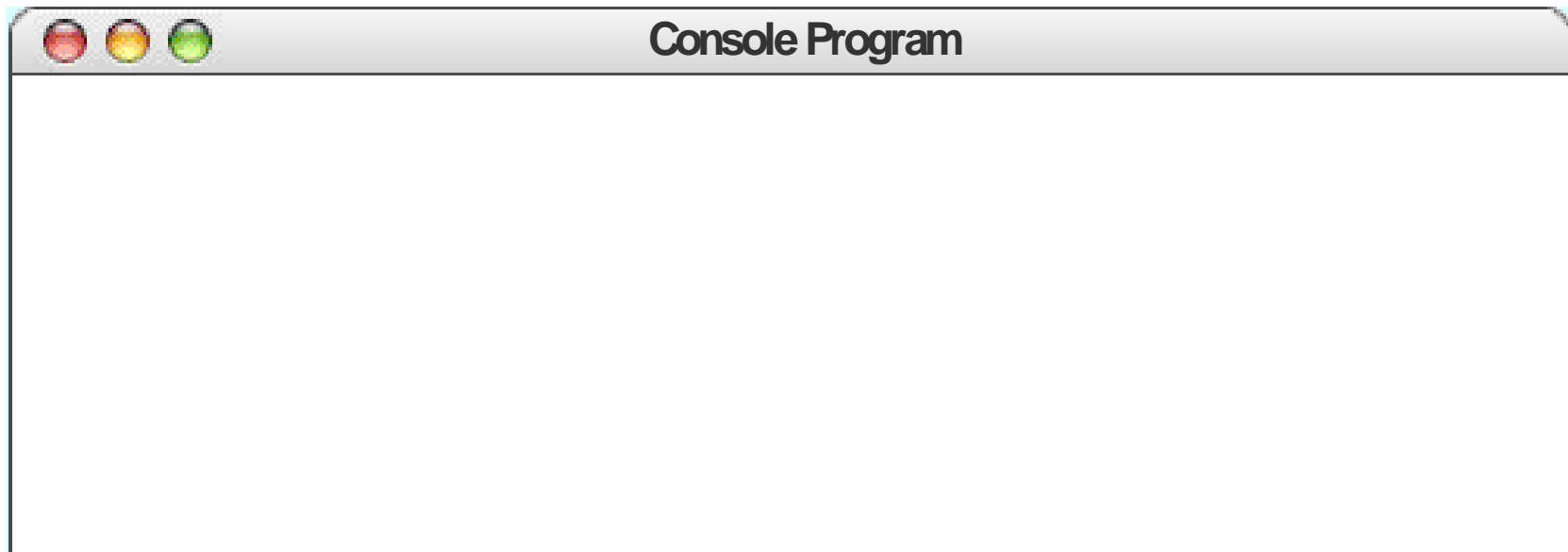
```
while (condition) {  
    ... statements ...  
}
```

- Checks *condition* before each iteration and executes *statements* if true.
- Does **not** check *condition* in the middle of the loop.

# while loop

## Example:

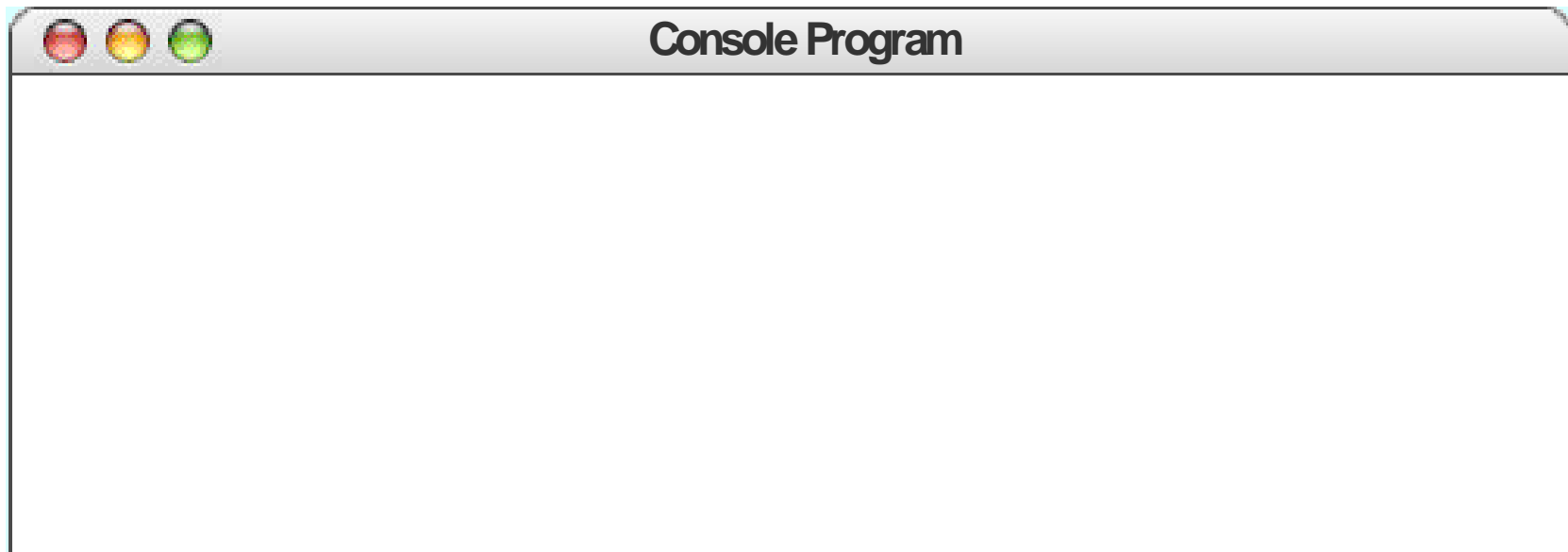
```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```



# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

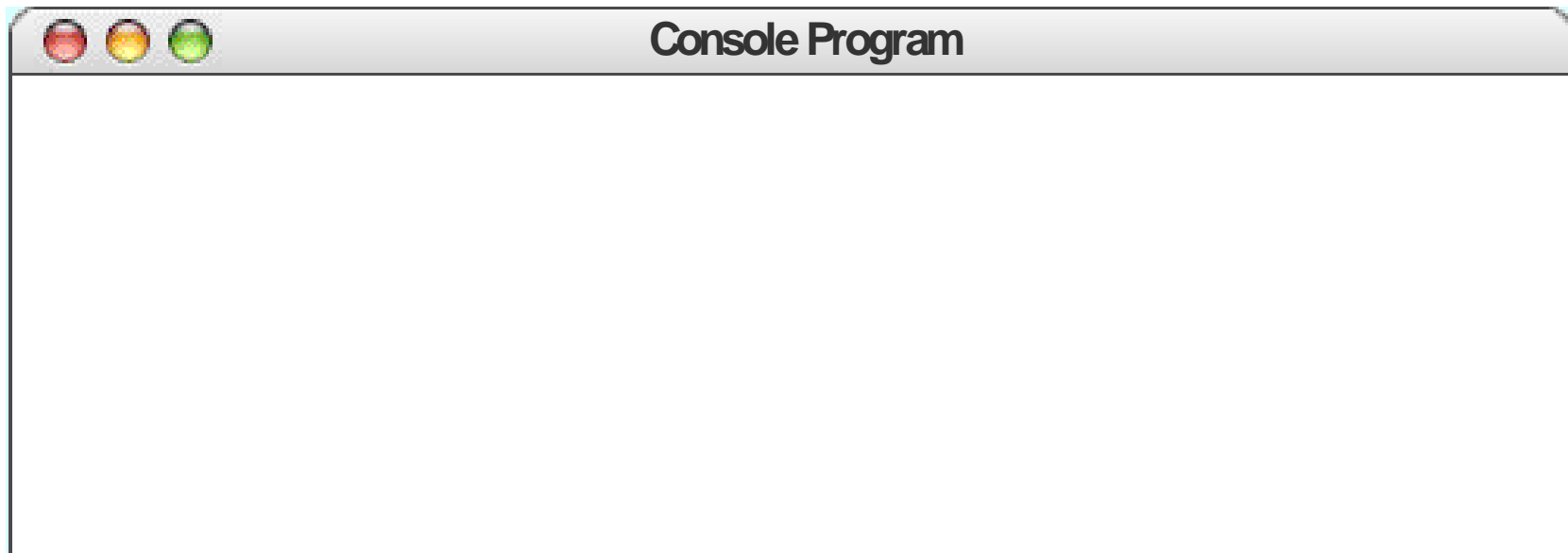


# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

15 int x

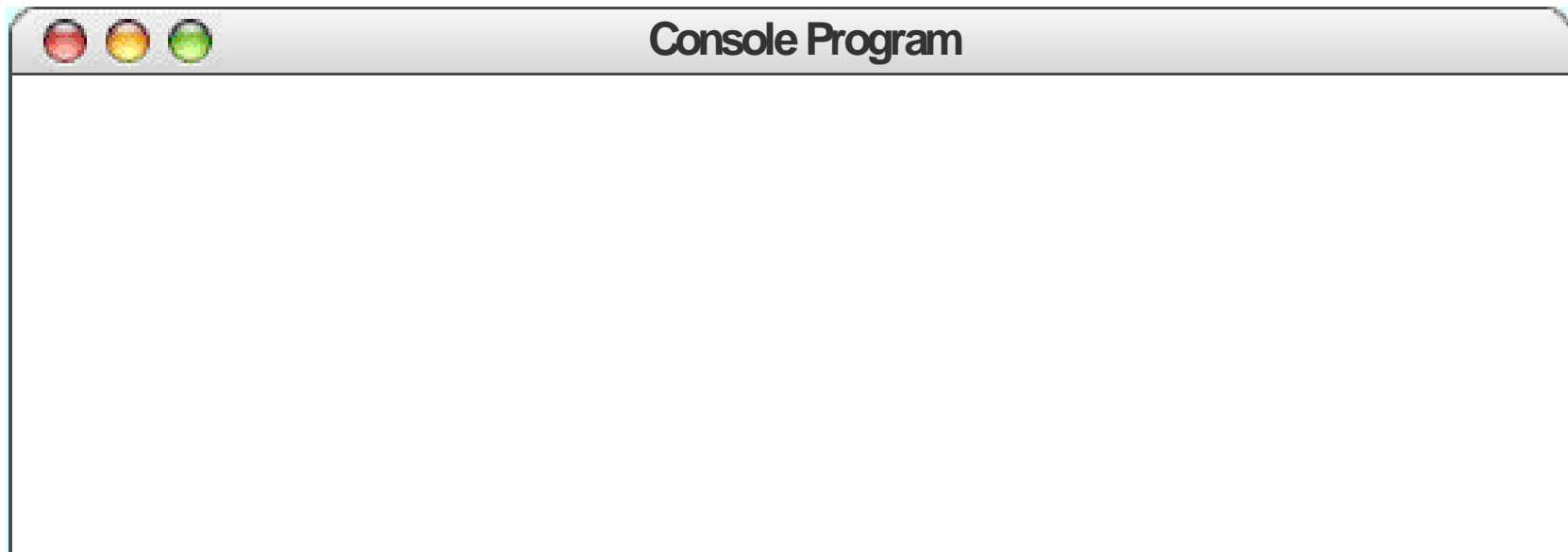


# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

15 int x



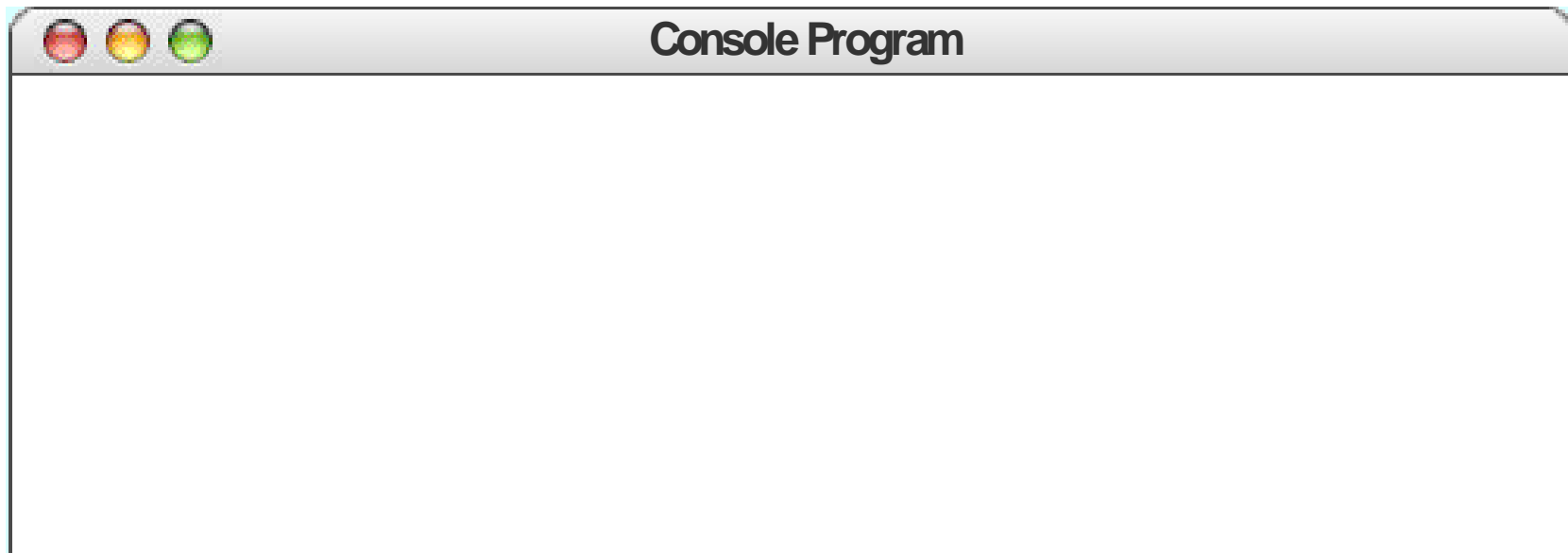
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

15

int x





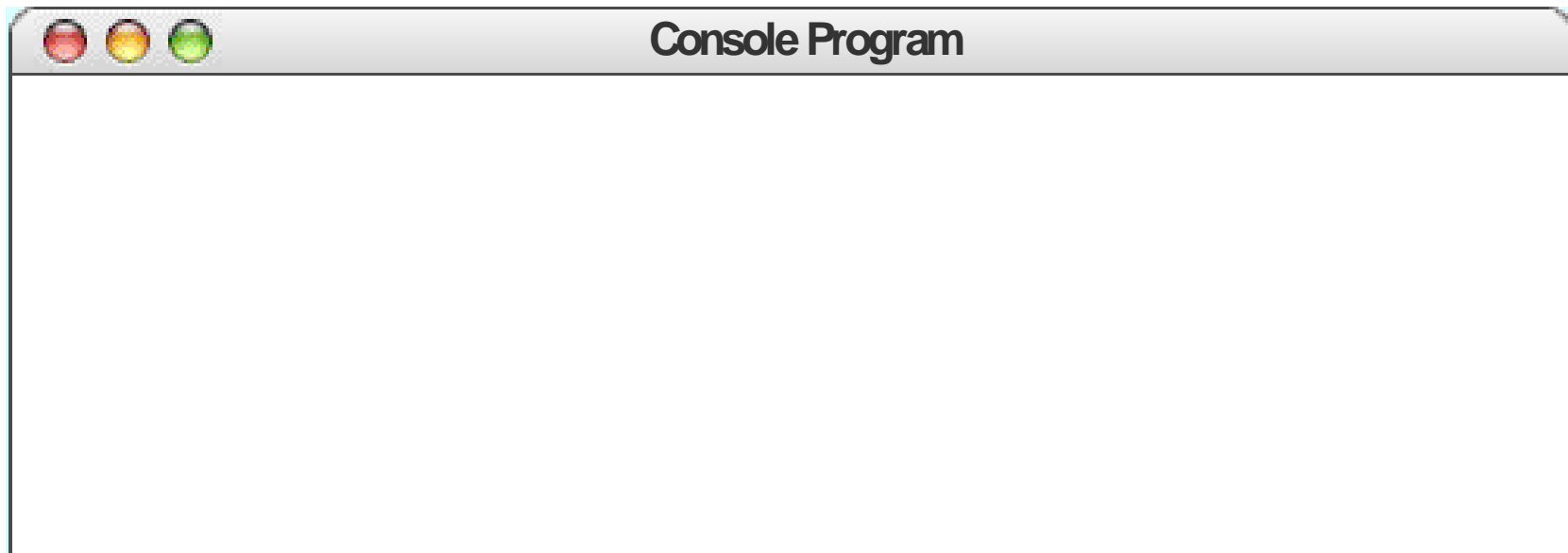
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

7

int x



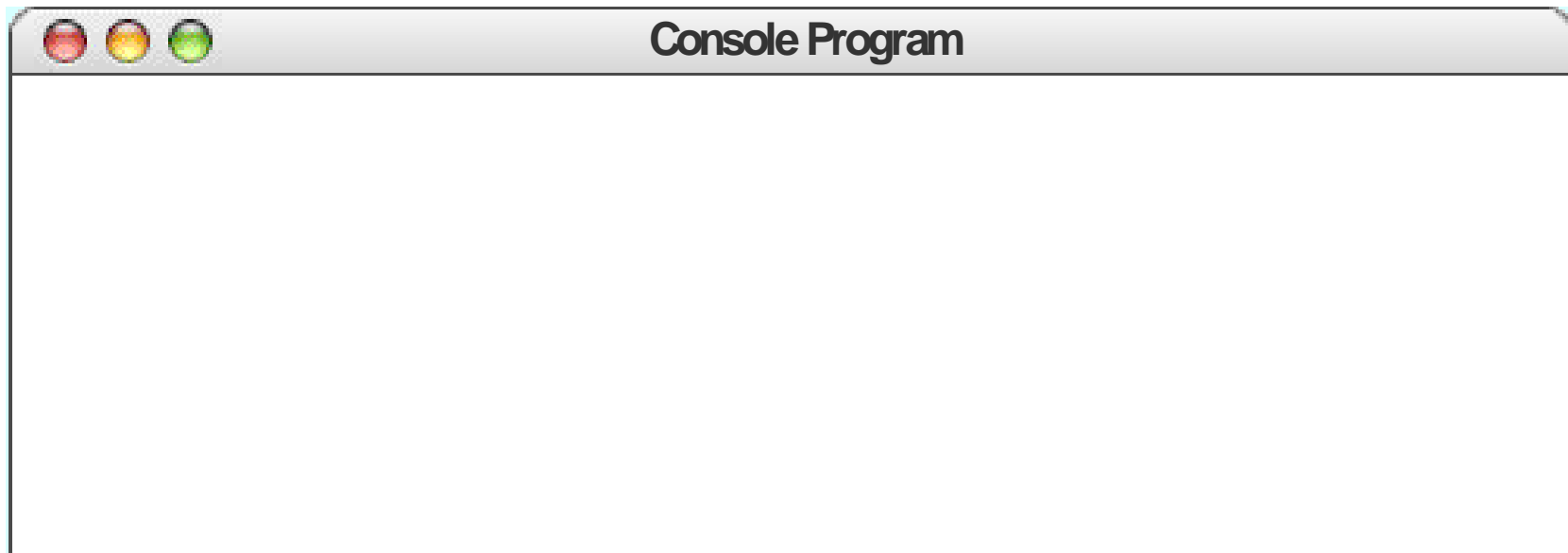
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

7

int x



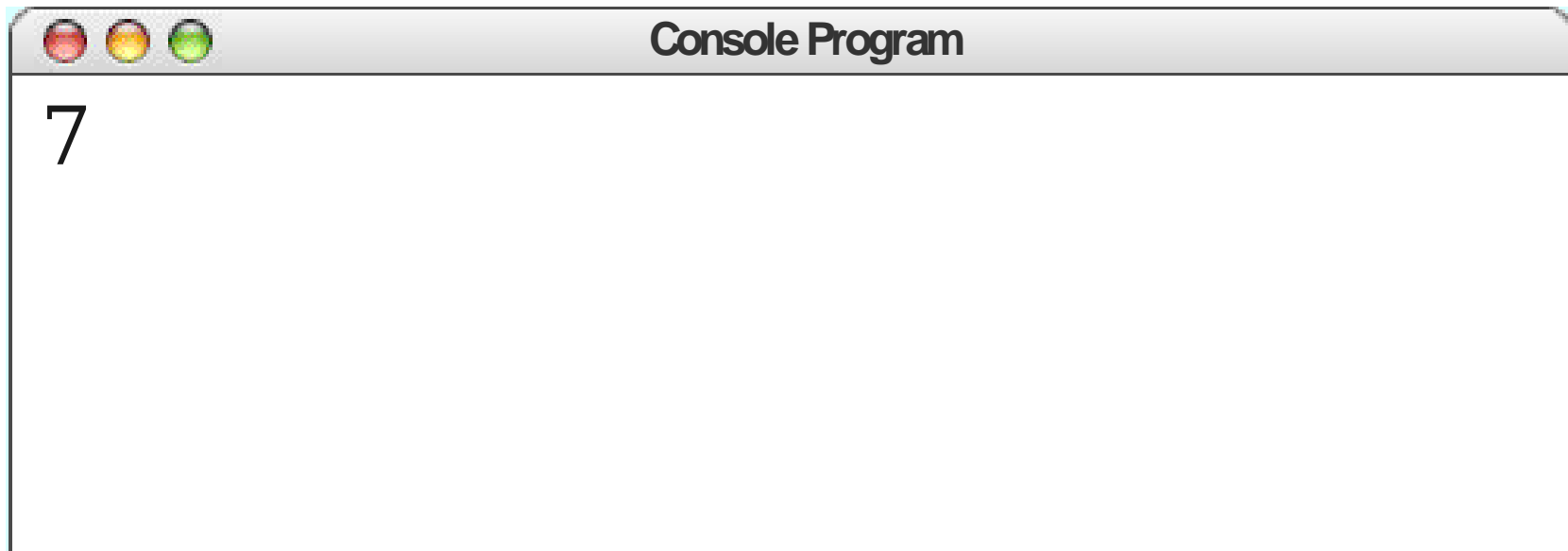
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

7

int x



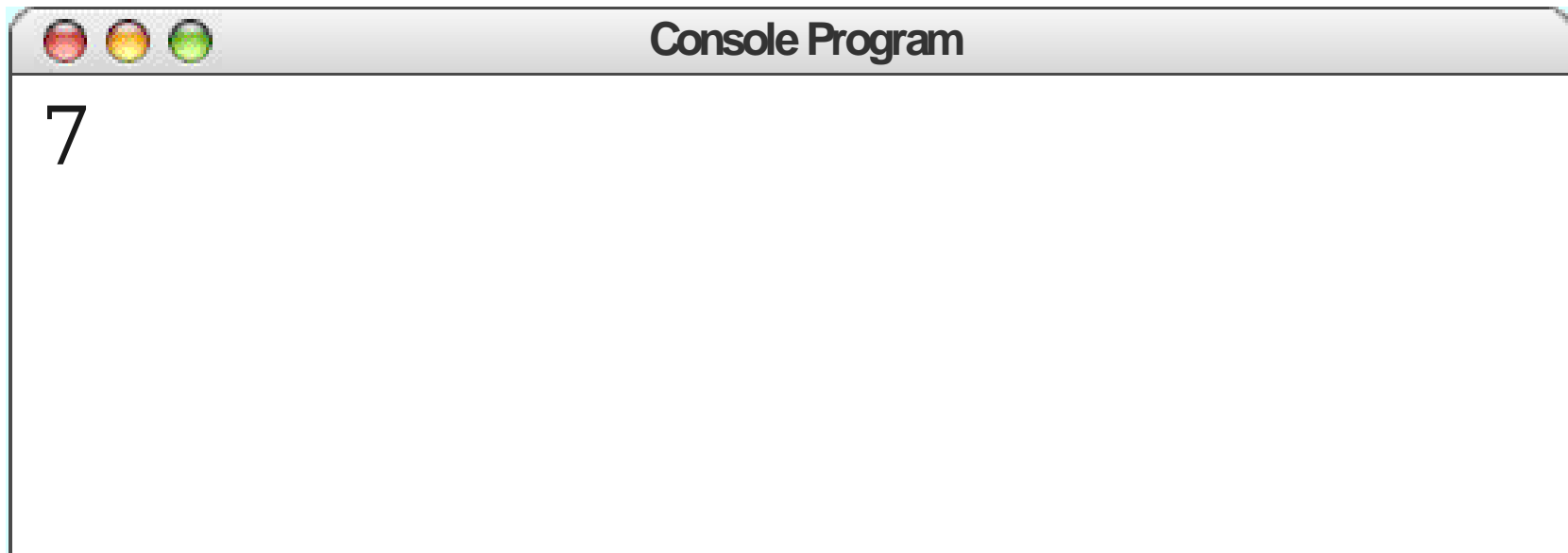
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

7

int x



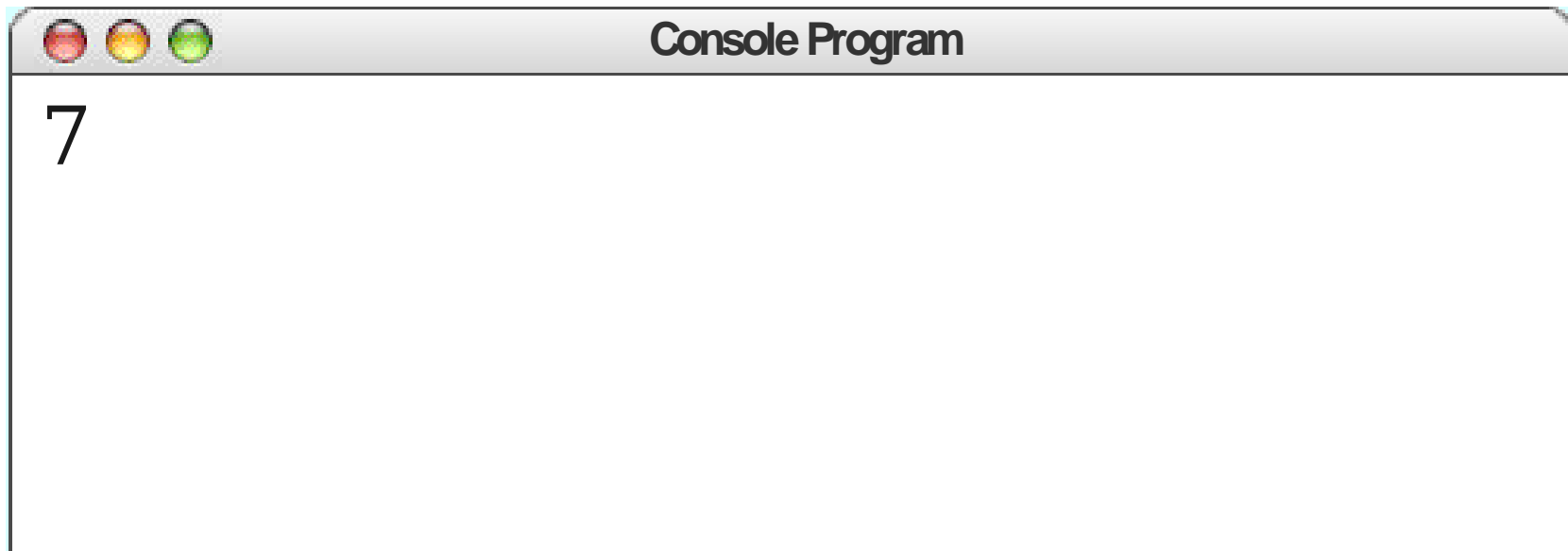
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

7

int x



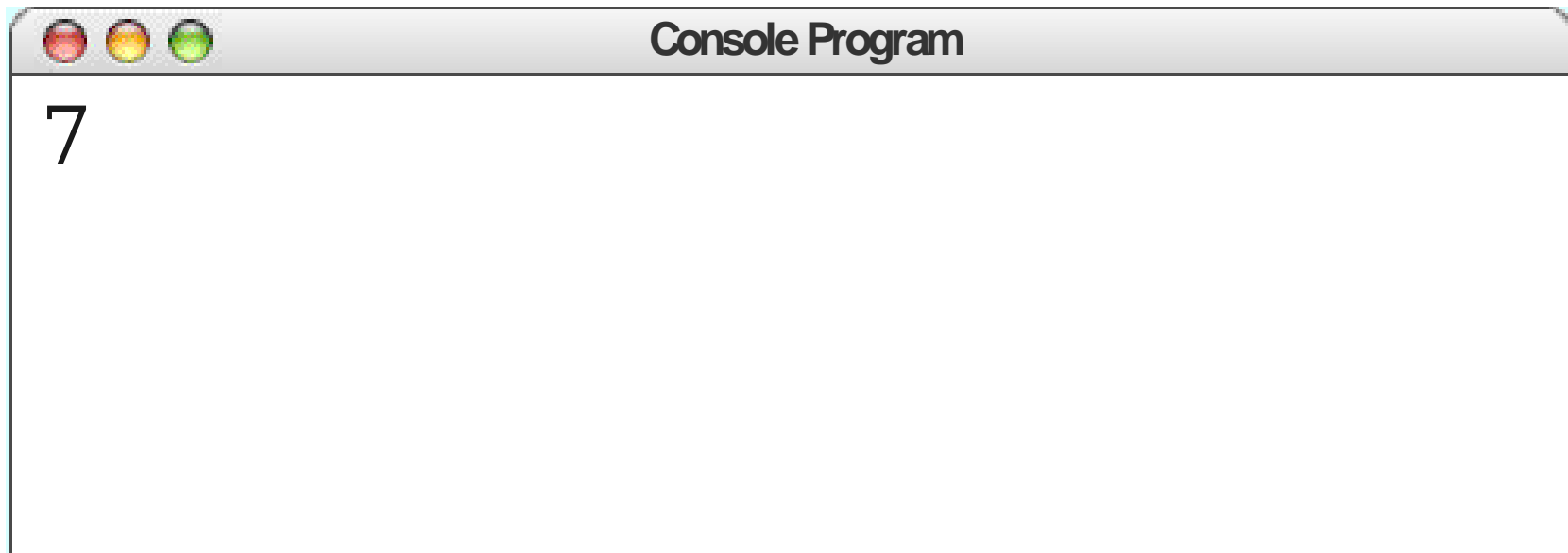
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

3

int x



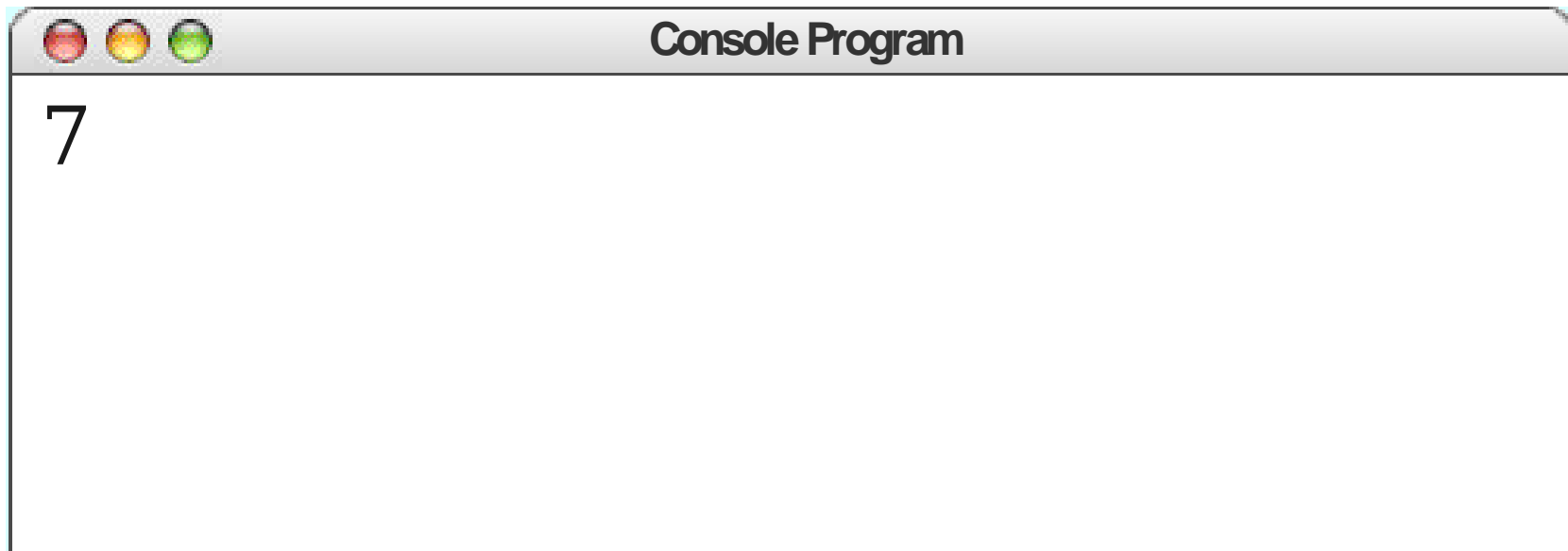
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

3

int x



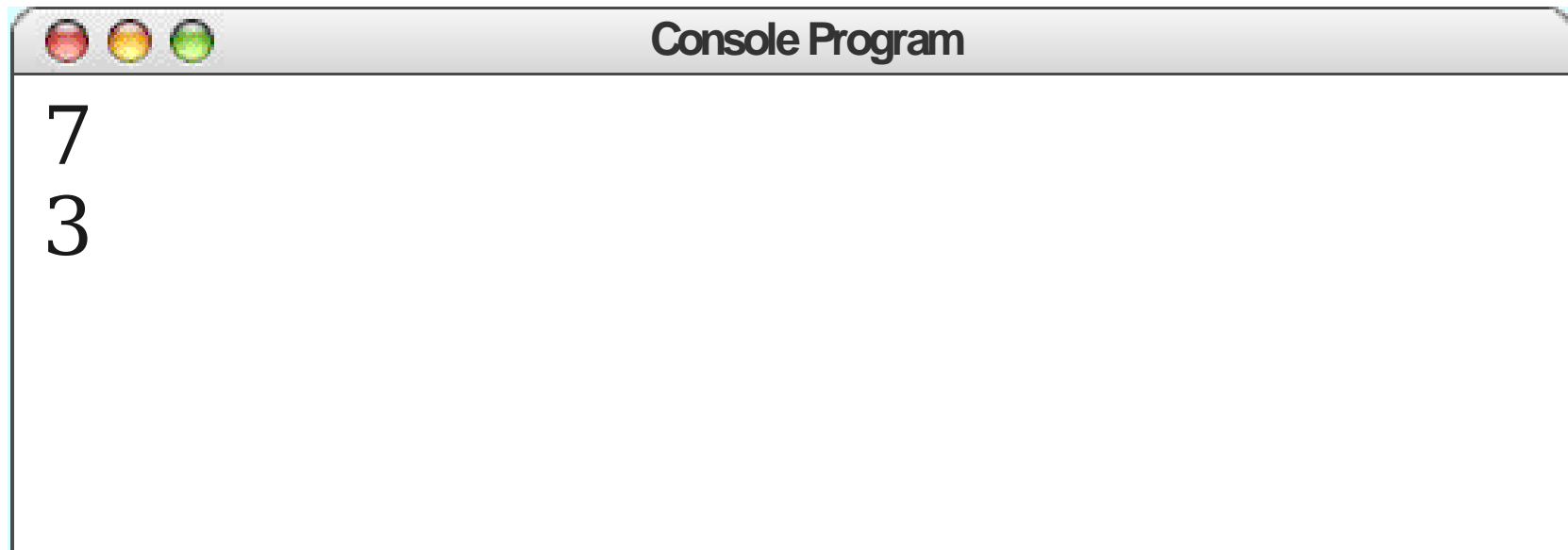
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

3

int x





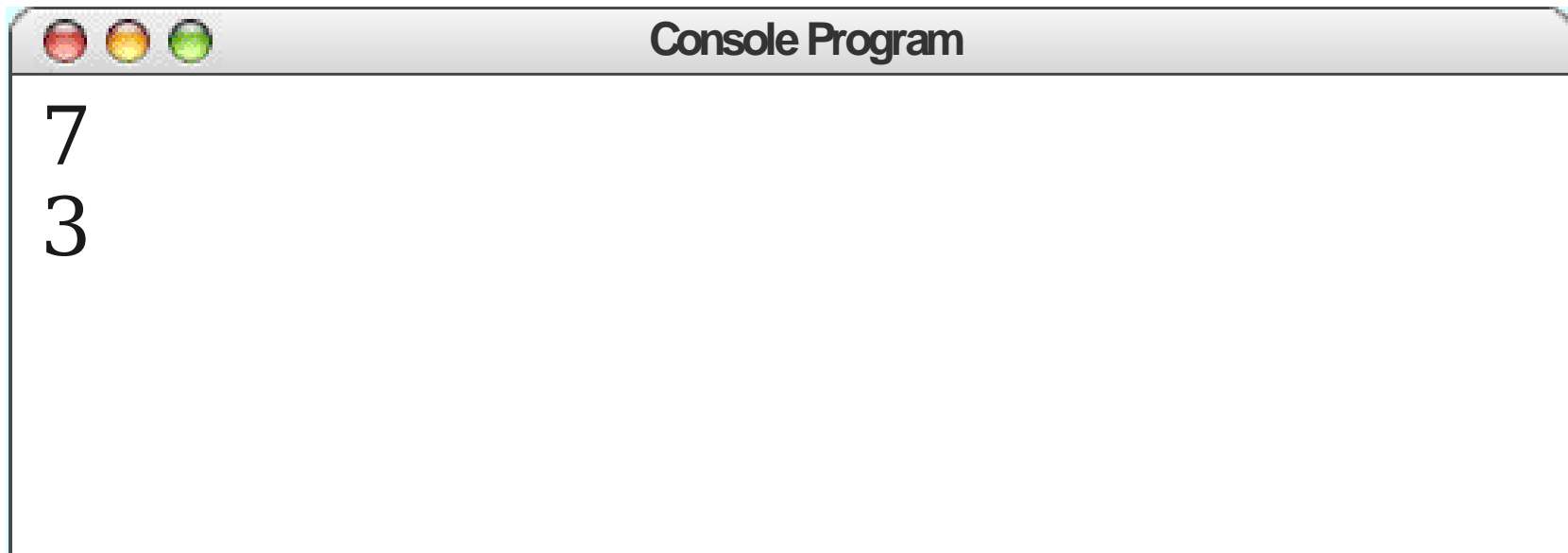
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

3

int x



Console Program

```
7  
3
```

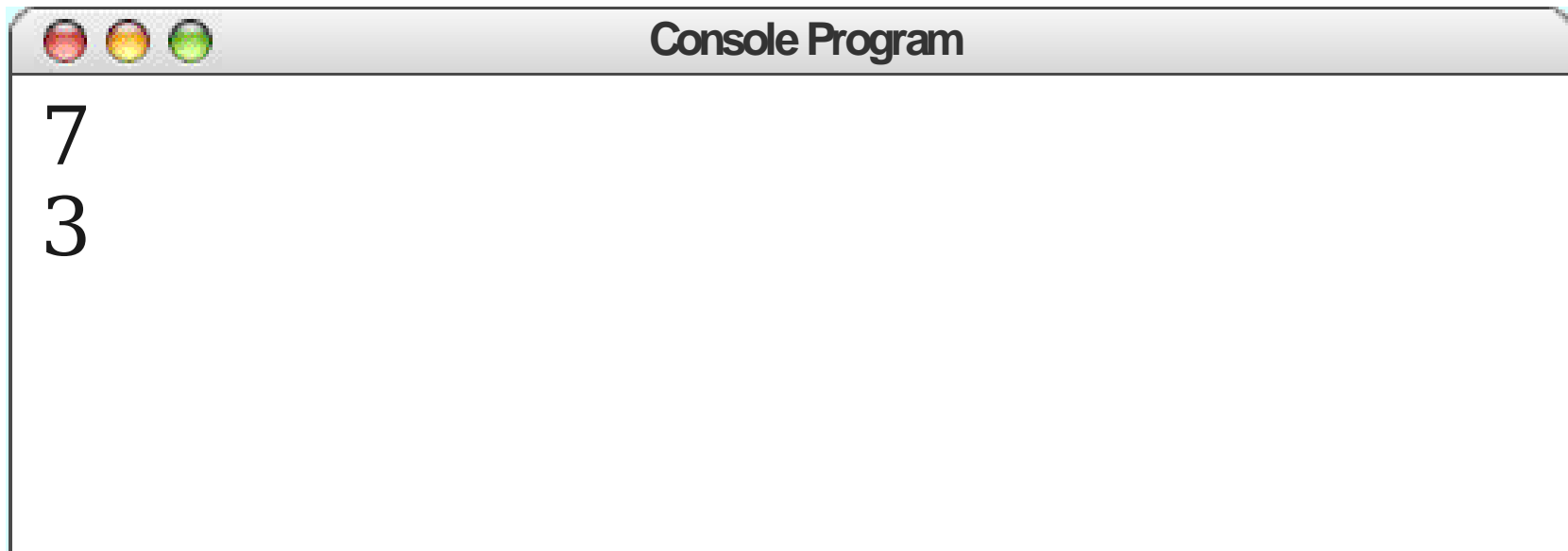
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

3

int x



Console Program

```
7  
3
```

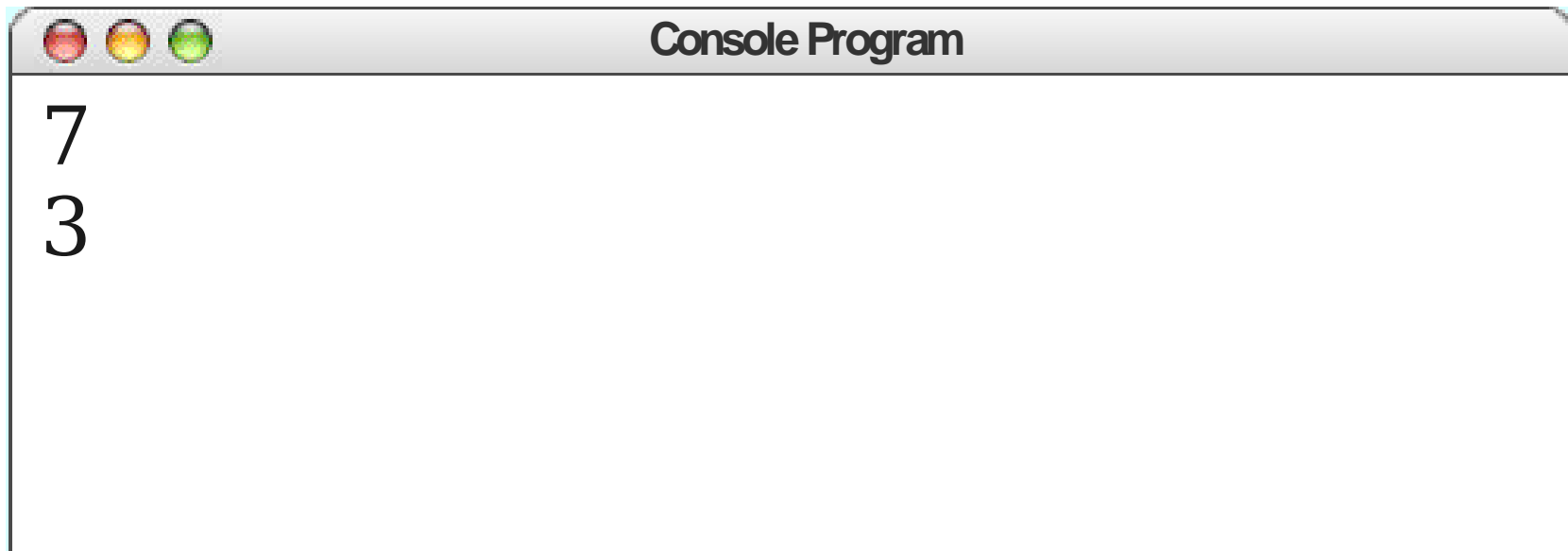
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

1

int x



Console Program

```
7  
3
```

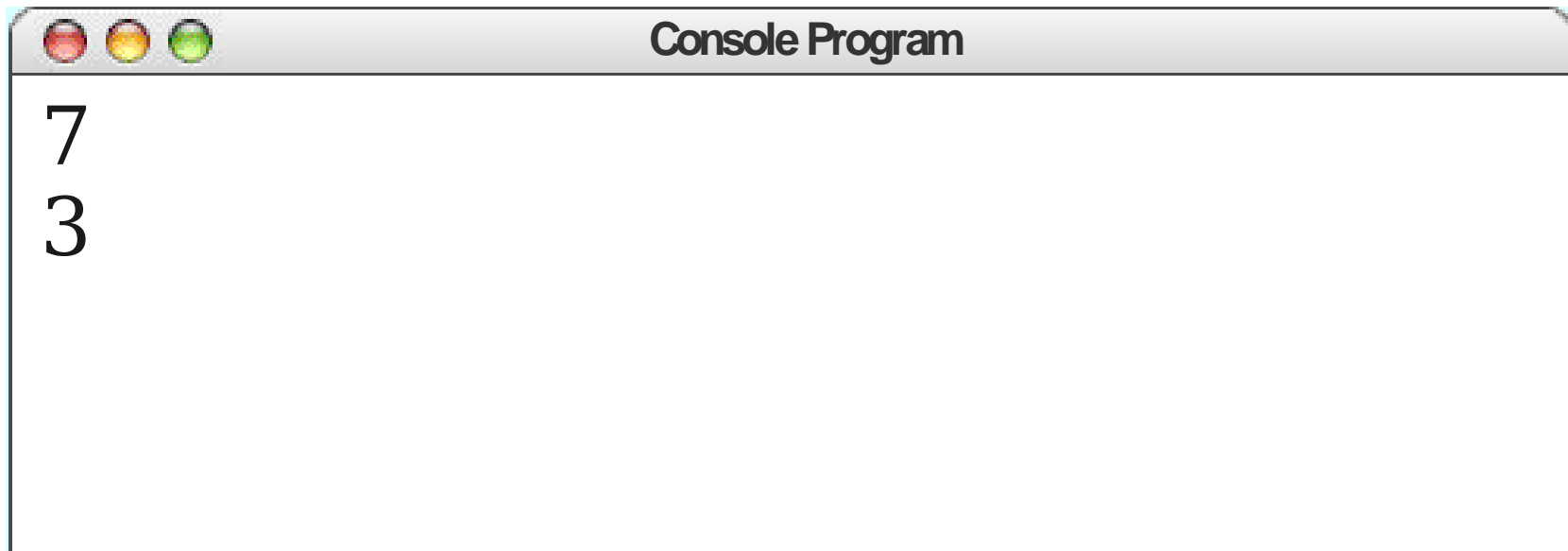
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

1

int x



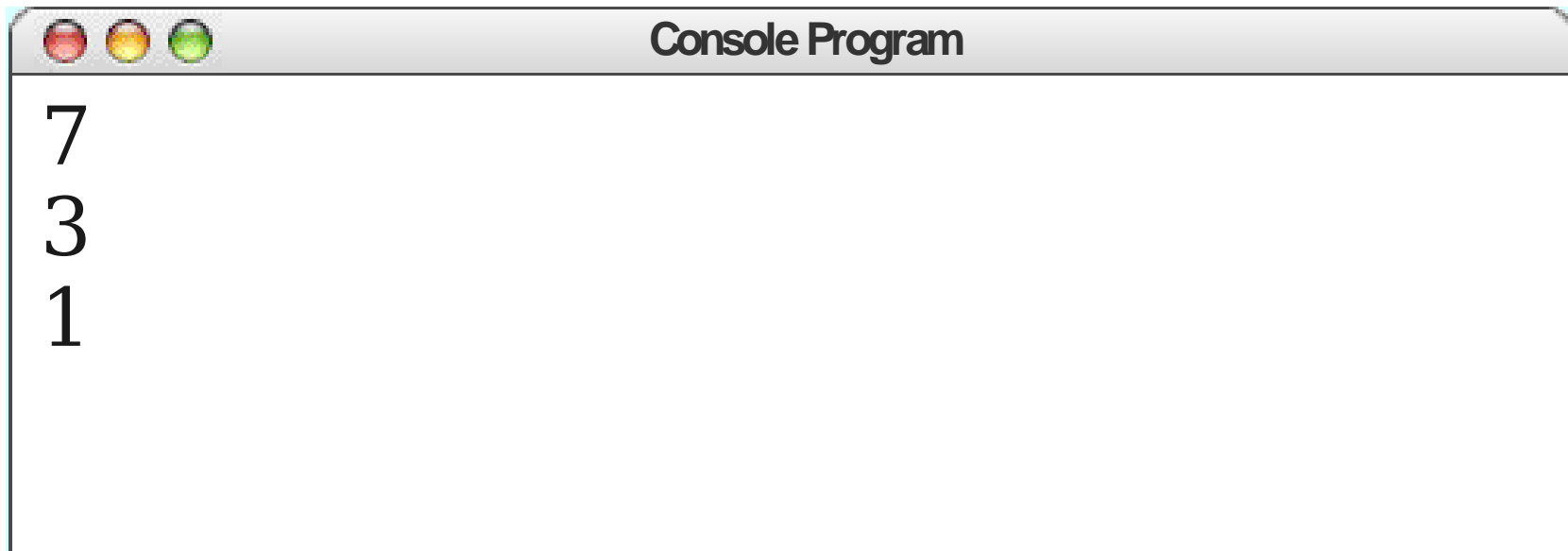
# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

1

int x



Console Program

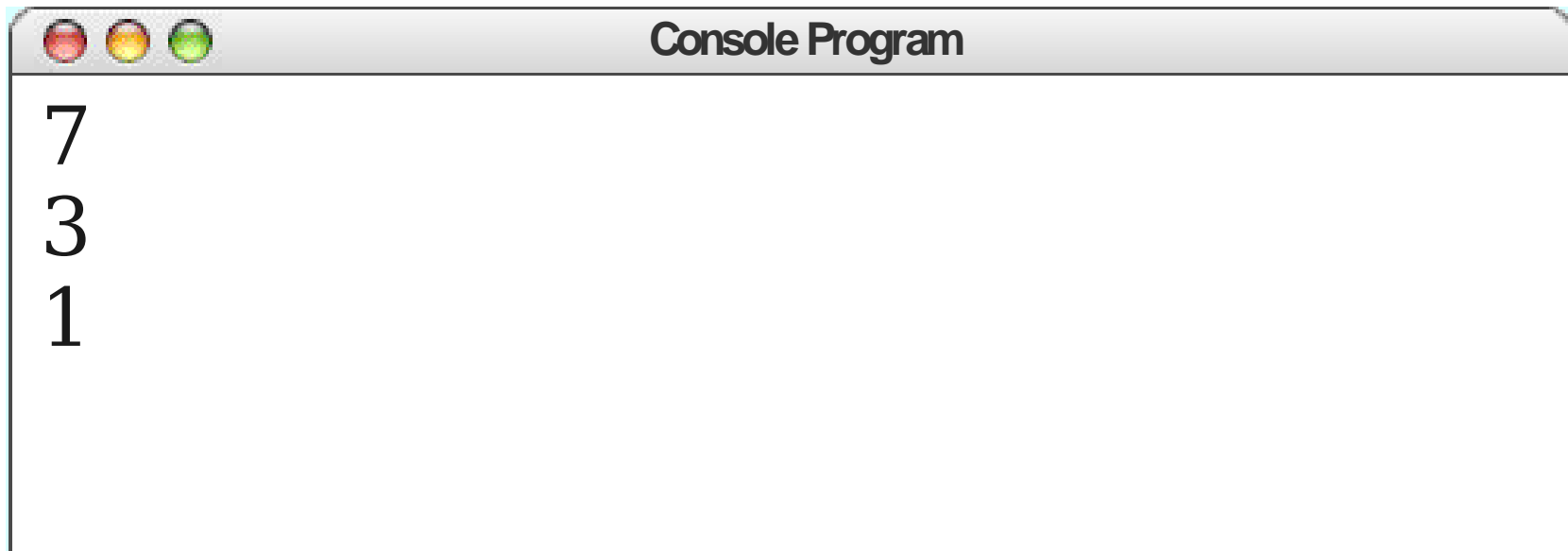
```
7  
3  
1
```

# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

1 int x



Console Program

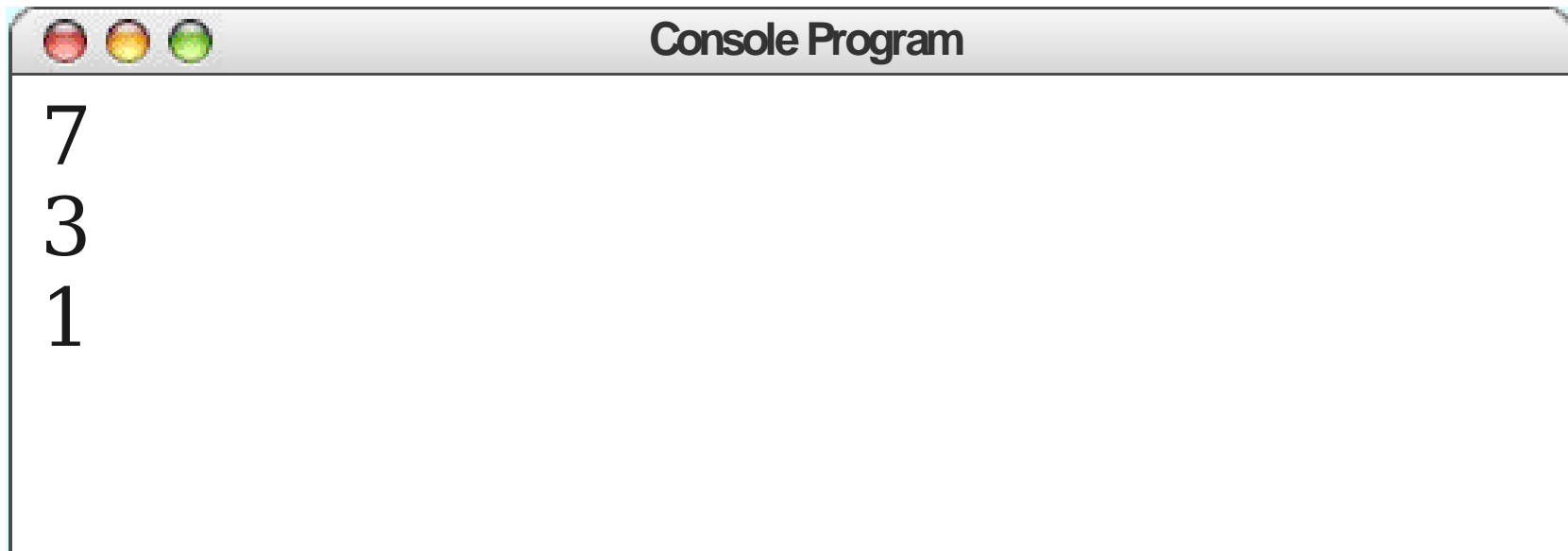
```
7  
3  
1
```

# while loop

## Example:

```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```

1 int x



Console Program

```
7  
3  
1
```

# Greatest Common Divisors

- Given two integers  $a$  and  $b$ , the **greatest common divisor** (or  $gcd$ ) of  $a$  and  $b$  is the largest number that divides  $a$  and  $b$ .
- Examples:
  - The  $gcd$  of 12 and 8 is 4.
  - The  $gcd$  of 100 and 10 is 10.
  - The  $gcd$  of 137 and 42 is 1.



# Euclid's Algorithm

45

35



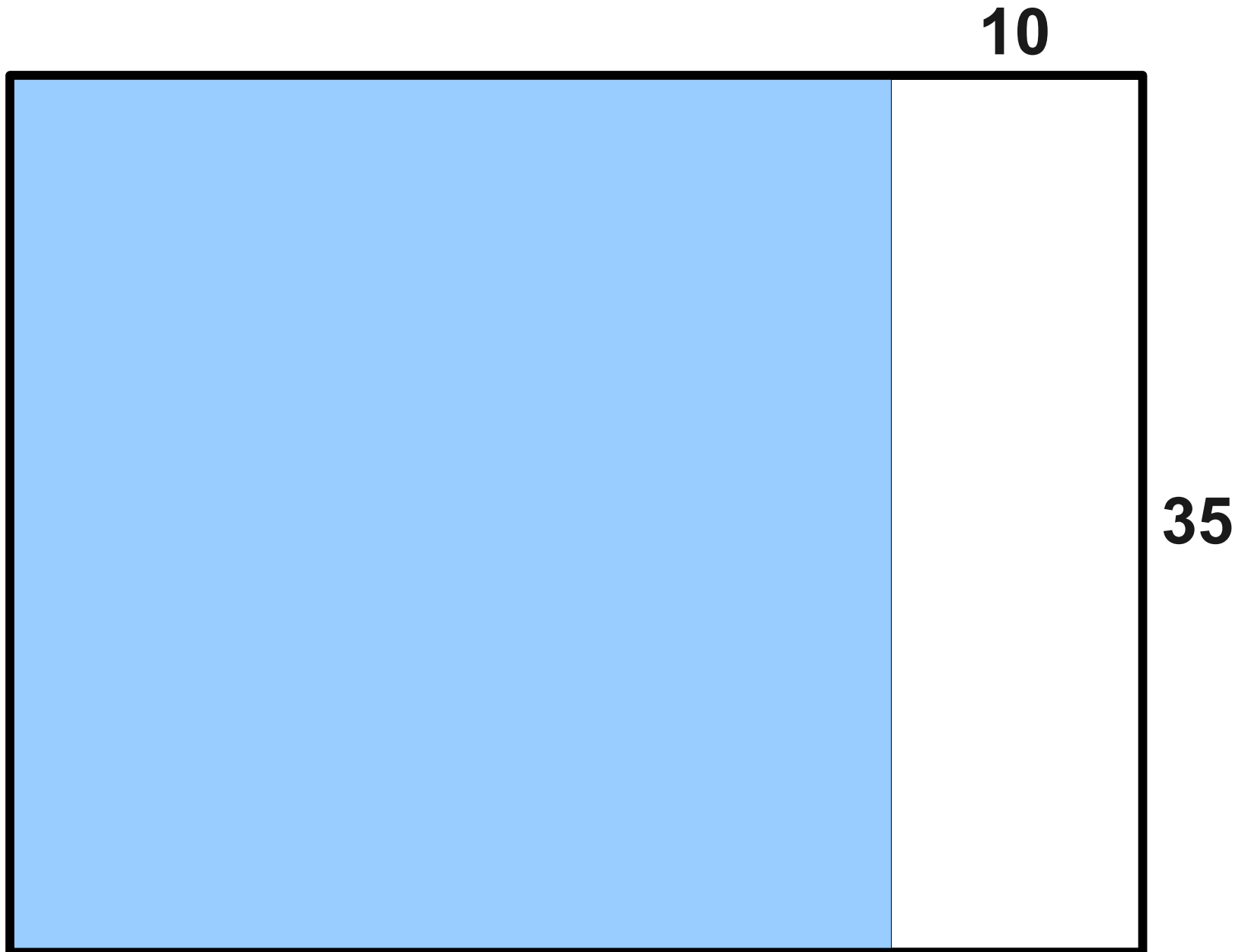
# Euclid's Algorithm

45

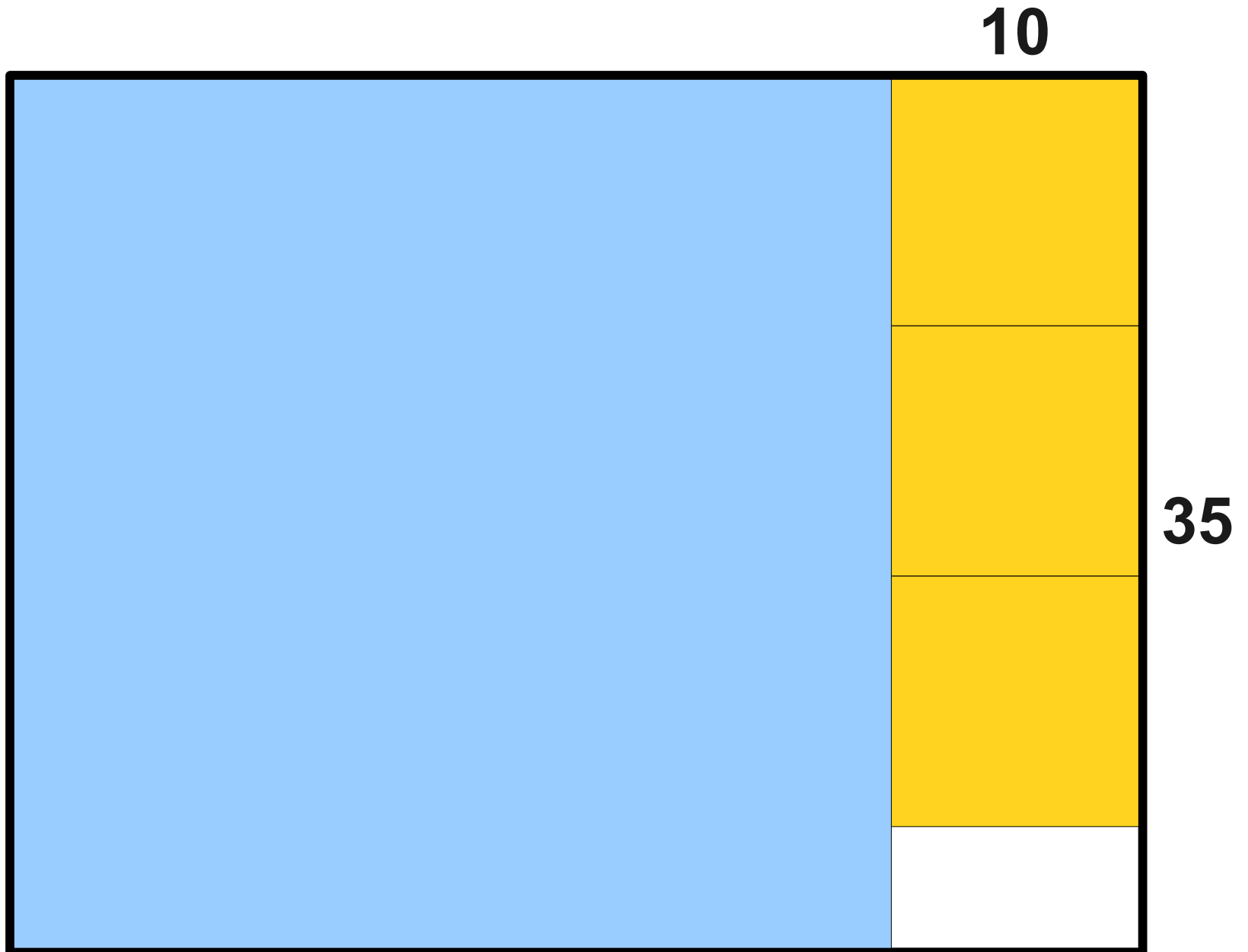
35



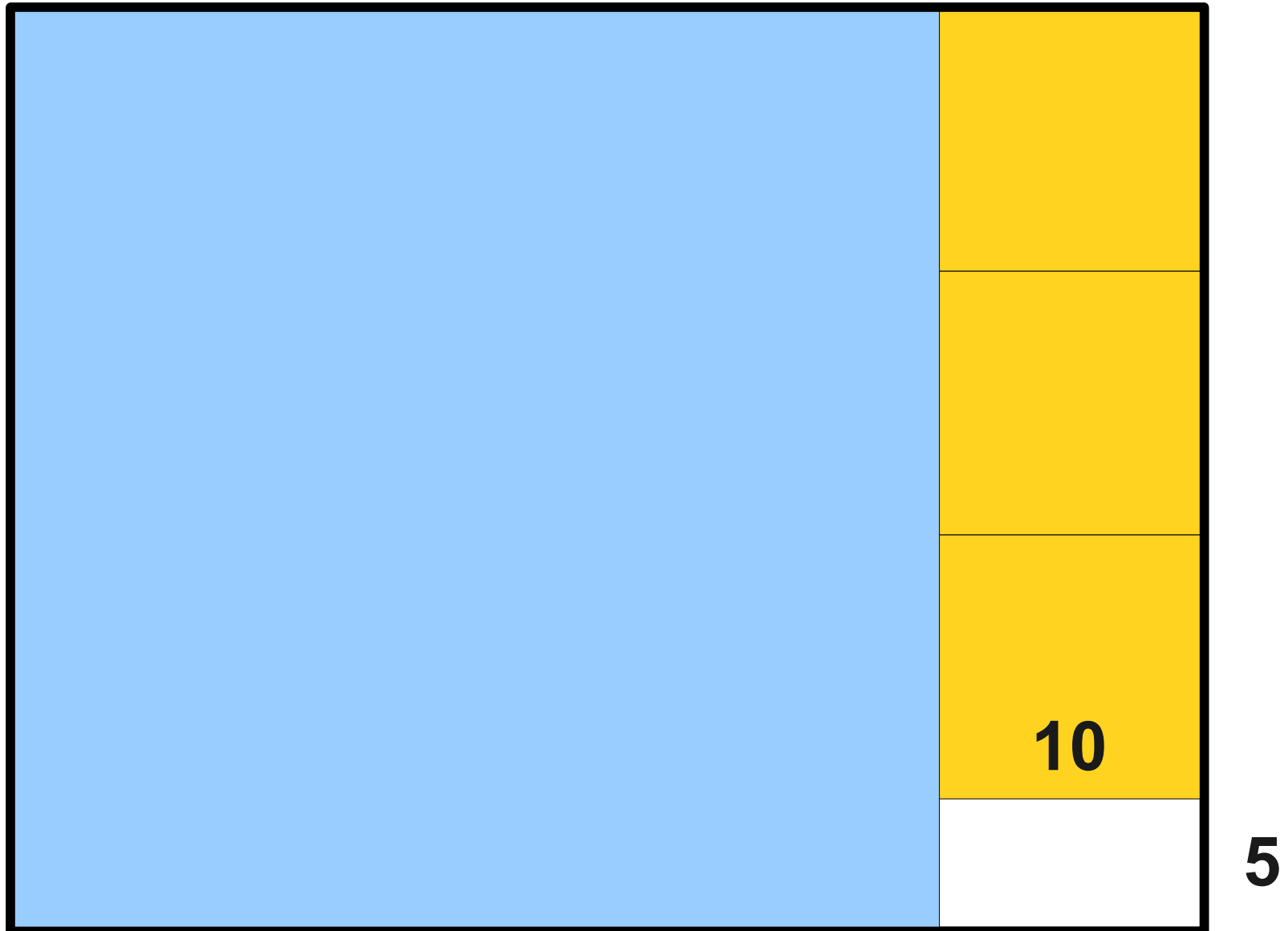
# Euclid's Algorithm



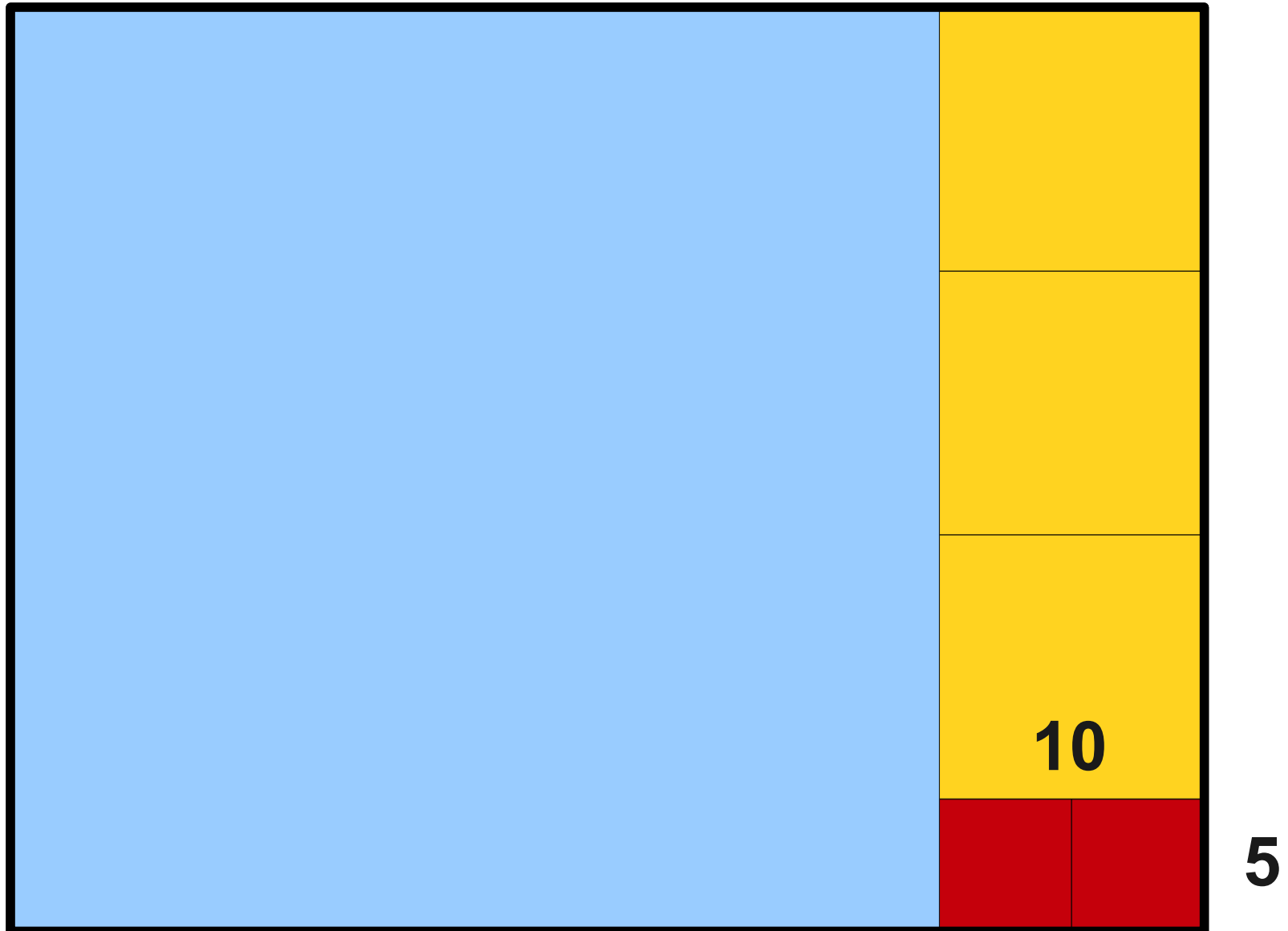
# Euclid's Algorithm



# Euclid's Algorithm



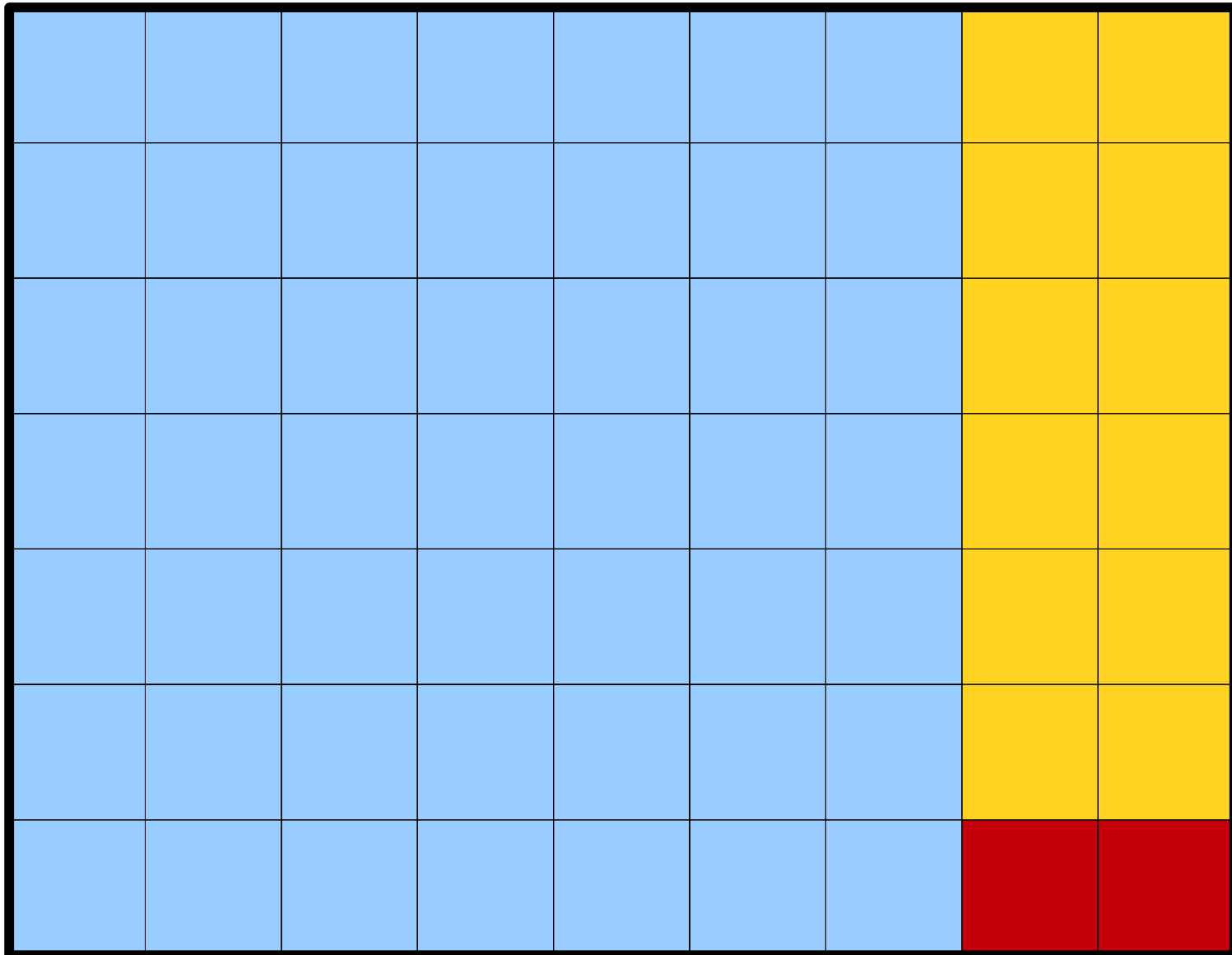
# Euclid's Algorithm



# Euclid's Algorithm

45

35



# Euclid's Algorithm

- To compute the *gcd* of  $a$  and  $b$ :
  - If  $b = 0$ , the *gcd* is  $a$ .
  - Otherwise:
    - Divide  $a$  by  $b$  and obtain the remainder  $r$ .
    - Set  $a$  equal to  $b$  and  $b$  equal to  $r$ .
    - Repeat.
- This procedure was known to the Greeks as **anthyphairesis**; it's almost always referred to now as **Euclid's algorithm**.
- It is one of the oldest algorithms still in use today.