

Control Statements Revisited

Announcements

- Assignment 2 (**Welcome to Java!**) is due on Monday, January 26 at 3:15PM.
- Quick reminder about the Honor Code:
 - Please cite any outside assistance you receive on the programming assignments.
 - Forgot to cite something? Please let us know within a week of submitting the assignment.

Outline for Today

- **Recap from Last Time**
 - Where are we, again?
- **Variable Assignment**
 - Changing variables over time.
- **Loops Revisited**
 - Aggregating information across loop iterations.
- **Random Numbers (ITA)**
 - Randomness meets computing.

Recap from Last Time

Passing Parameters

- A method can accept ***parameters*** when it is called.
- Syntax:

```
private void name(parameters) {  
    /* ... method body ... */  
}
```

- We use parameters to make methods *customizable* in their behavior.
- There are a lot of nuances that we'll talk about later on, but for now, treat parameters as “the way” to communicate this sort of information.

Assignment Statements

- A statement of the form

variable = newValue;

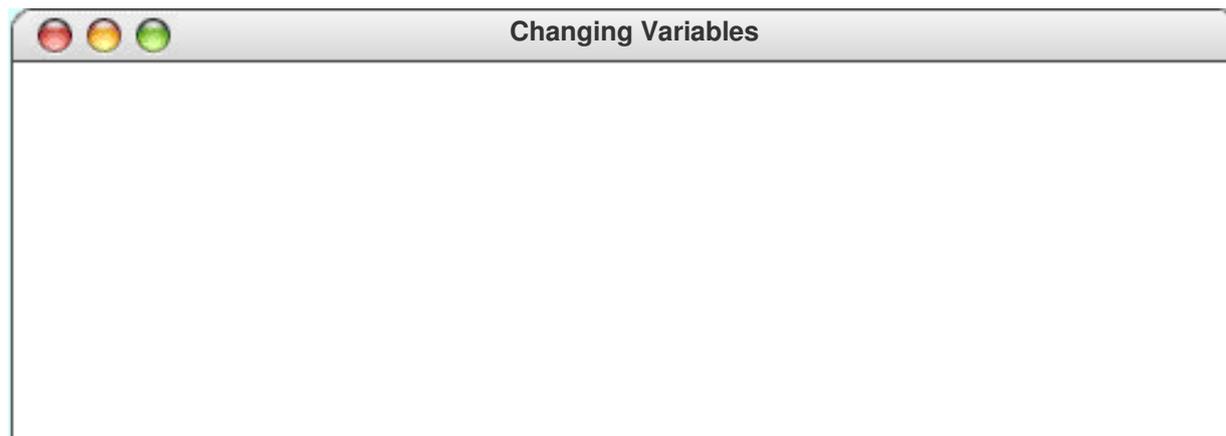
changes ***variable*** so that it now stores ***newValue*** instead of its old value.

- This statement is called an ***assignment statement***.

An Example

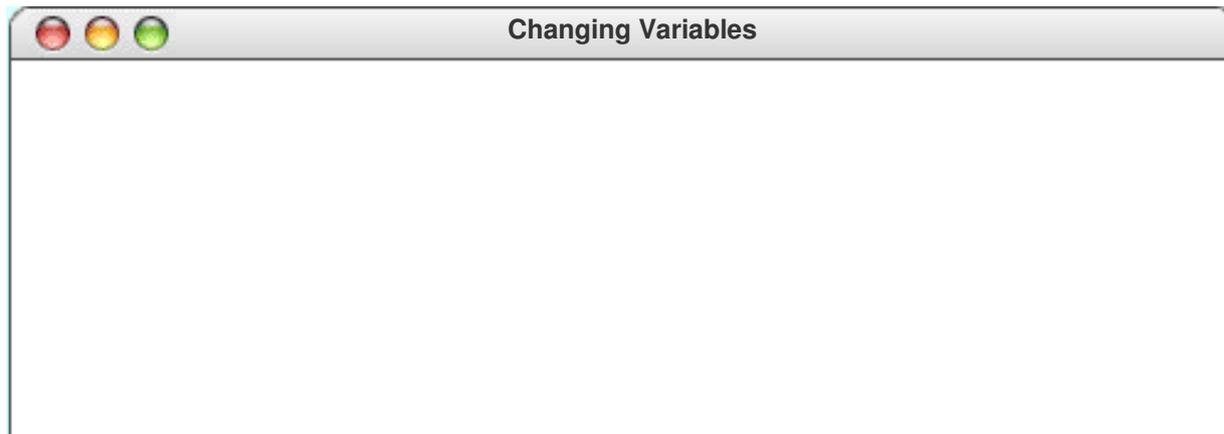
```
public void run() {
```

```
}
```



An Example

```
public void run() {  
    int favoriteNumber = 4;  
  
}
```



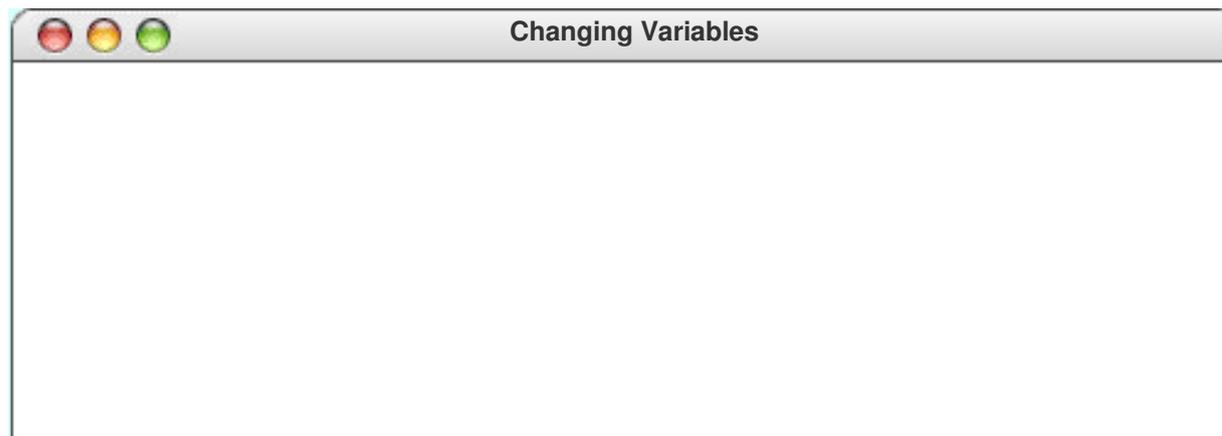
An Example

```
public void run() {  
    int favoriteNumber = 4;  
}
```

favoriteNumber



}



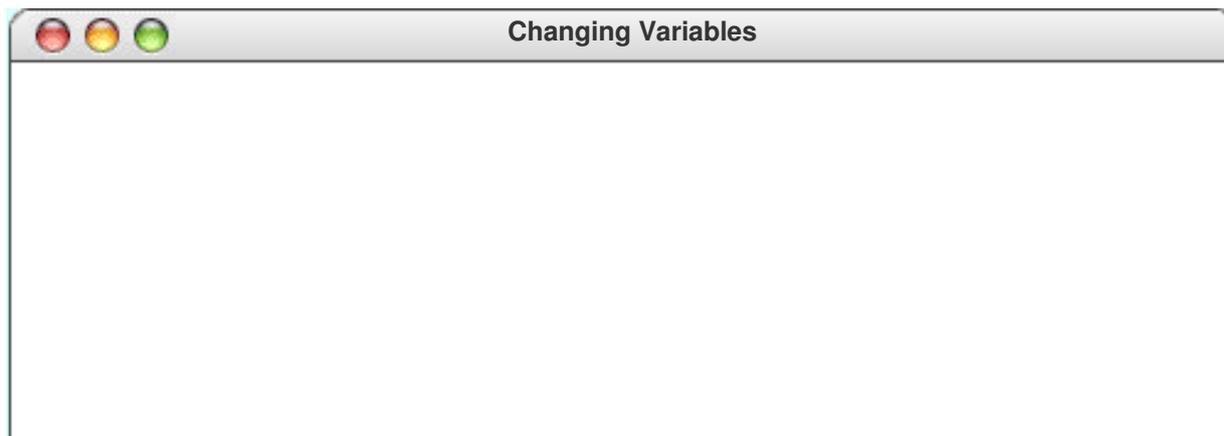
An Example

```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);  
}
```

favoriteNumber



}



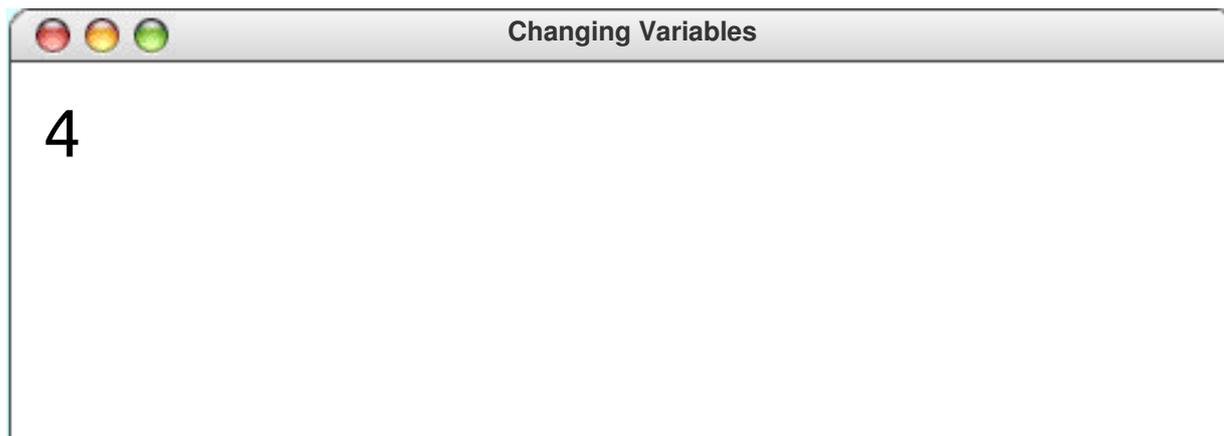
An Example

```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);  
}
```

favoriteNumber



}



An Example

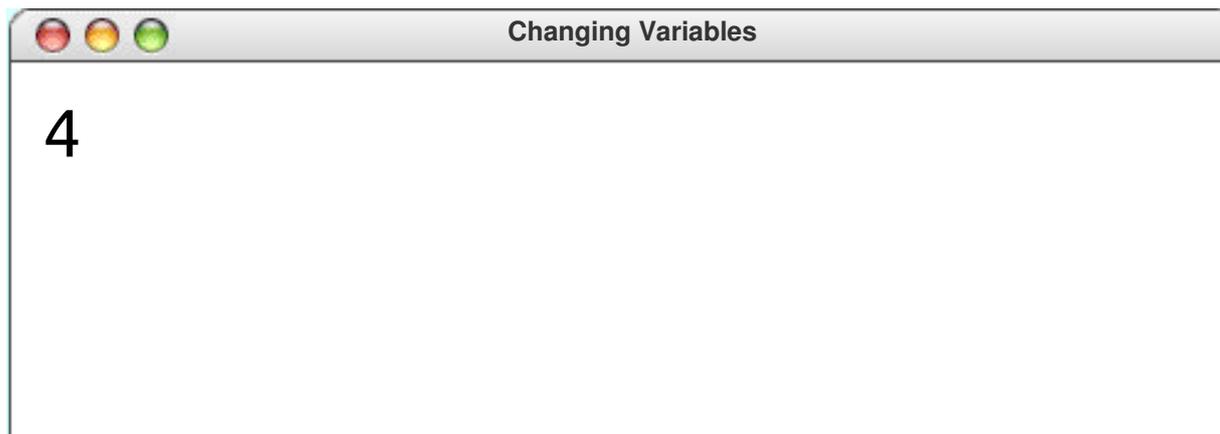
```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);
```

favoriteNumber

4

```
favoriteNumber = 137;
```

```
}
```



An Example

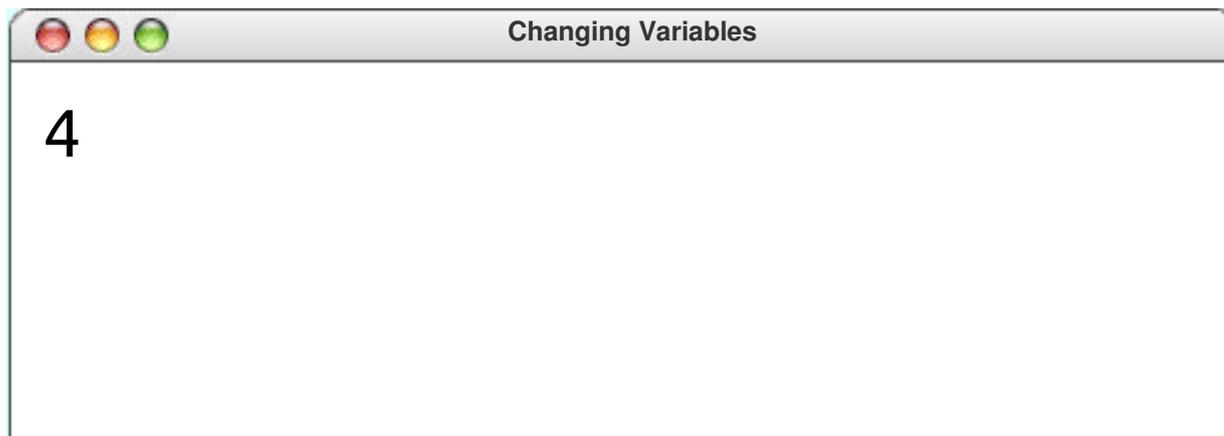
```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);
```

favoriteNumber

137

```
favoriteNumber = 137;
```

```
}
```



An Example

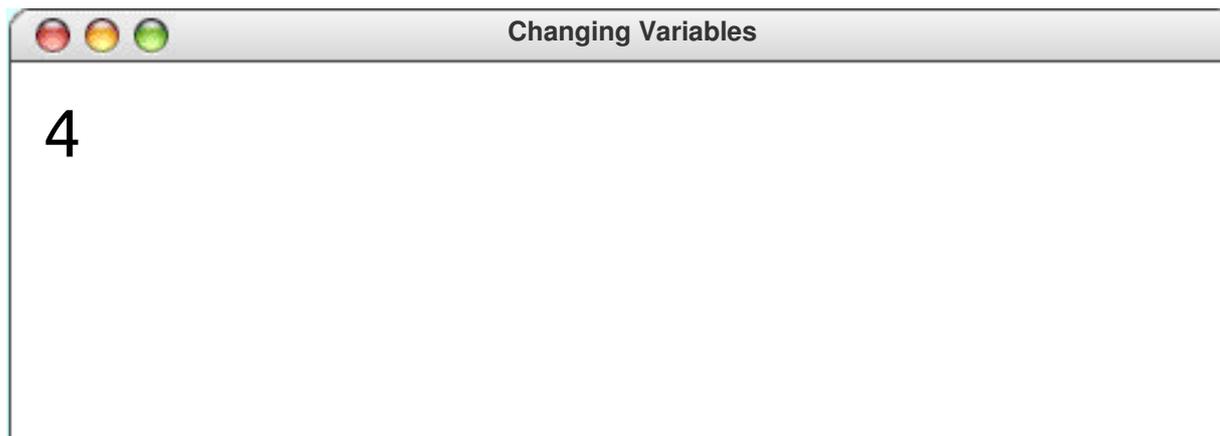
```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);
```

favoriteNumber

137

```
    favoriteNumber = 137;  
    println(favoriteNumber);
```

```
}
```



An Example

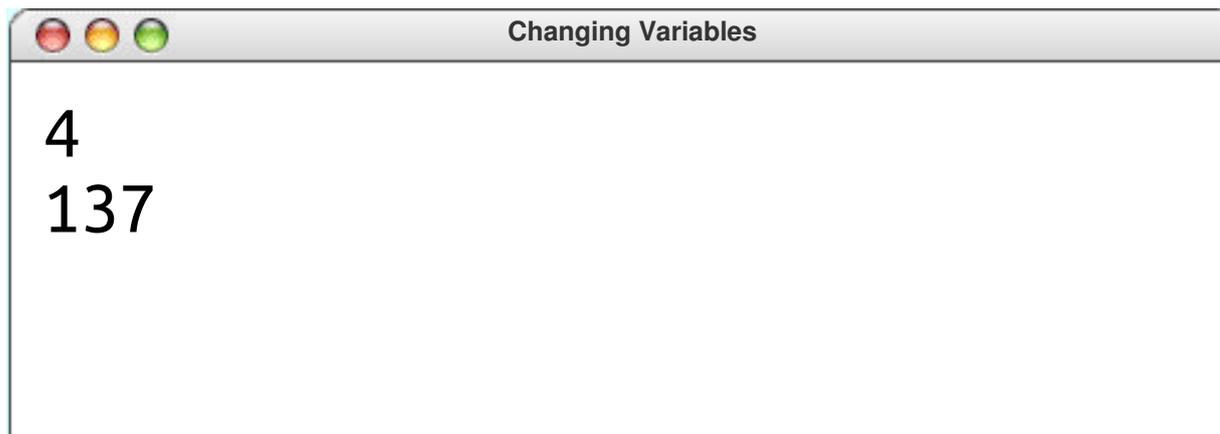
```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);
```

favoriteNumber

137

```
    favoriteNumber = 137;  
    println(favoriteNumber);
```

```
}
```



An Example

```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);
```

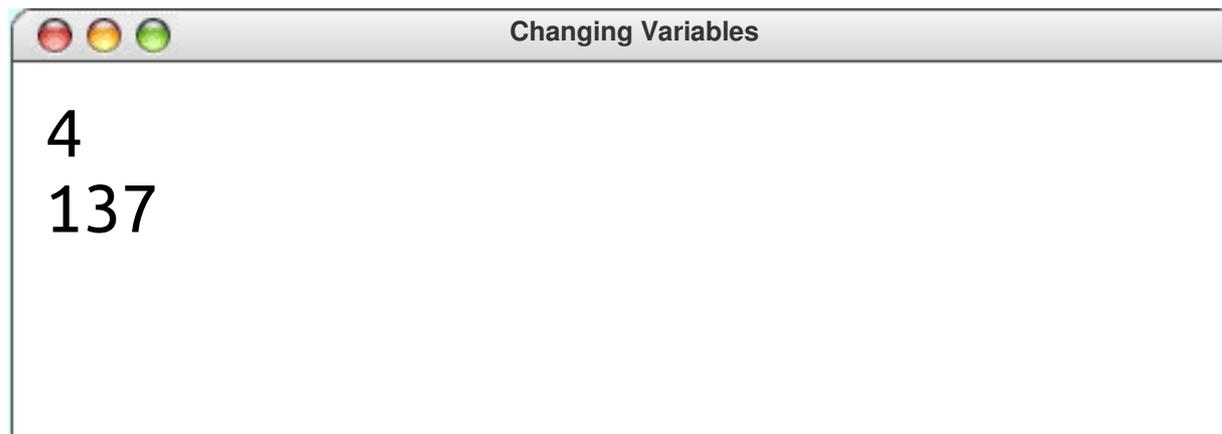
favoriteNumber

137

```
    favoriteNumber = 137;  
    println(favoriteNumber);
```

```
    favoriteNumber = 137 + 42;
```

```
}
```



An Example

```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);
```

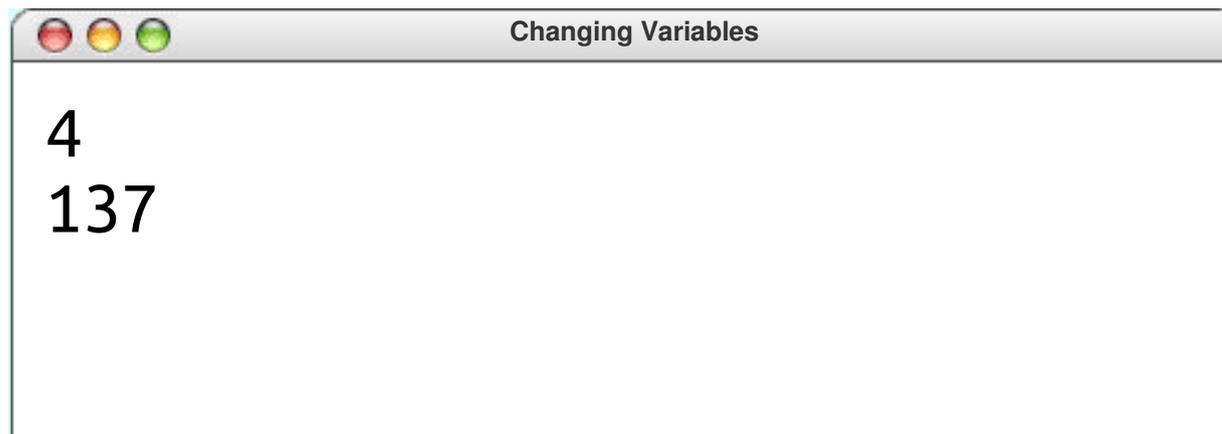
favoriteNumber

179

```
    favoriteNumber = 137;  
    println(favoriteNumber);
```

```
    favoriteNumber = 137 + 42;
```

```
}
```



An Example

```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);
```

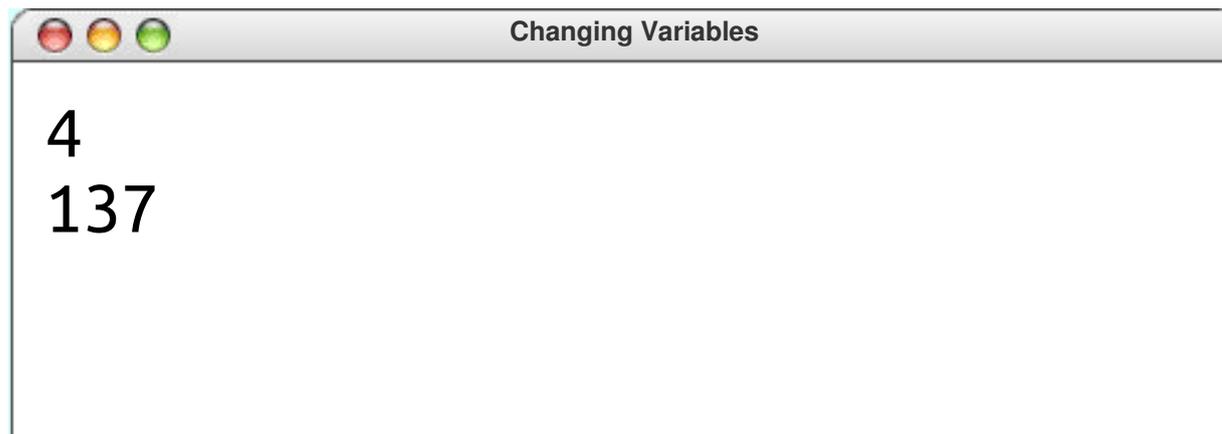
favoriteNumber

179

```
    favoriteNumber = 137;  
    println(favoriteNumber);
```

```
    favoriteNumber = 137 + 42;  
    println(favoriteNumber);
```

```
}
```



An Example

```
public void run() {  
    int favoriteNumber = 4;  
    println(favoriteNumber);
```

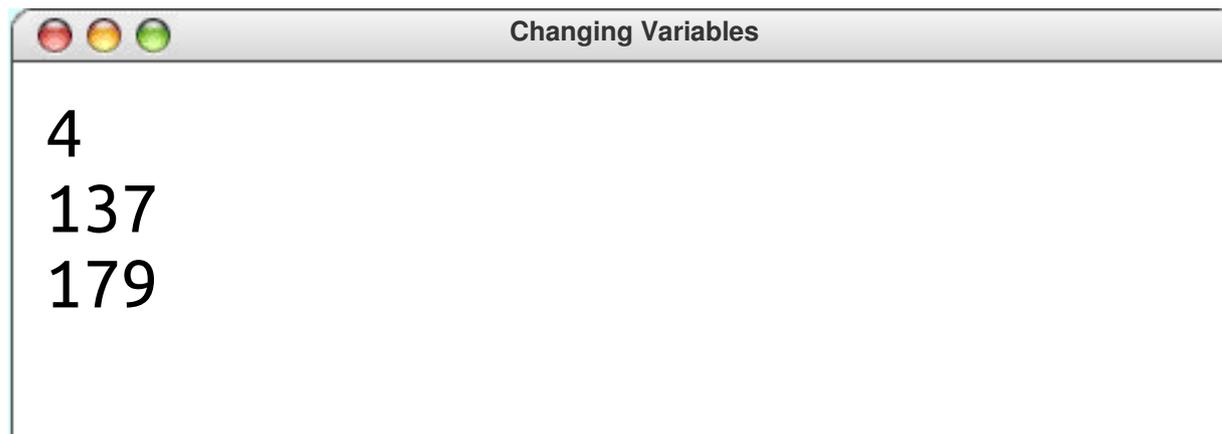
favoriteNumber

179

```
    favoriteNumber = 137;  
    println(favoriteNumber);
```

```
    favoriteNumber = 137 + 42;  
    println(favoriteNumber);
```

```
}
```



Another Example

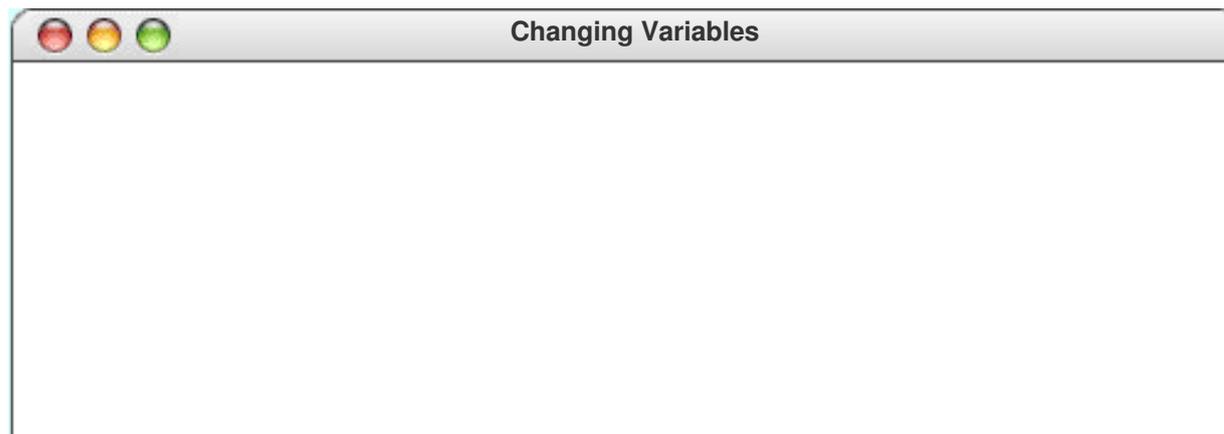
```
public void run() {
```

```
}
```

Another Example

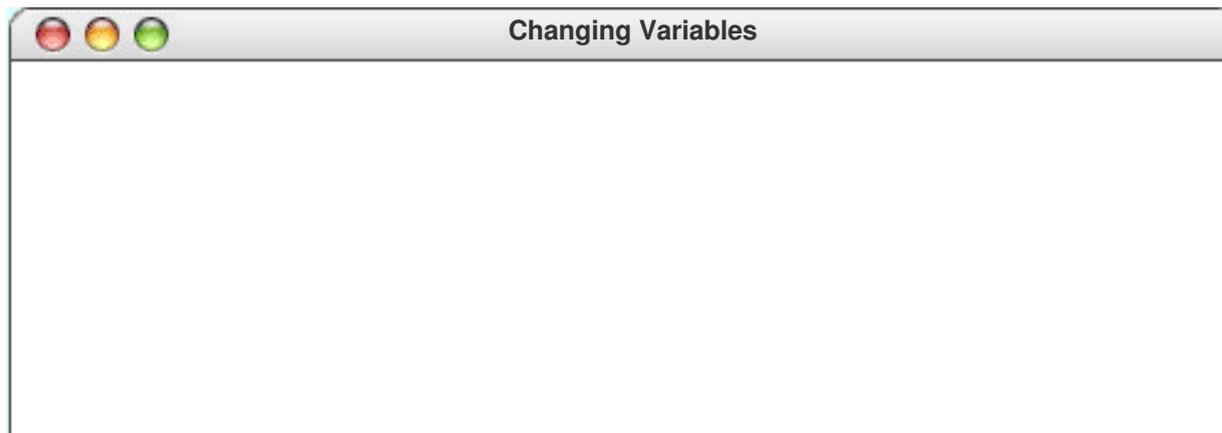
```
public void run() {
```

```
}
```



Another Example

```
public void run() {  
    int a = 5;  
    int b = 7;  
  
}
```



Another Example

```
public void run() {  
    int a = 5;  
    int b = 7;
```

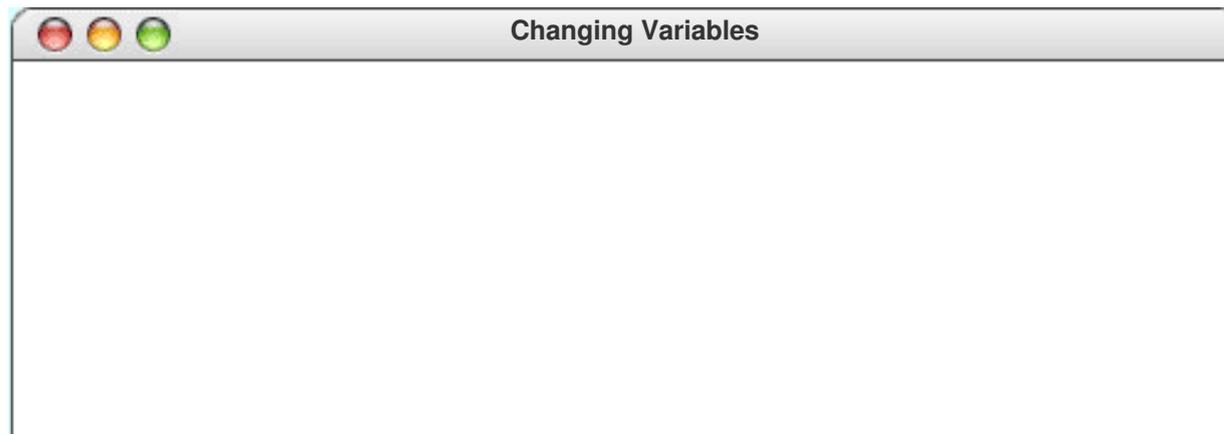
a

5

b

7

```
}
```



Another Example

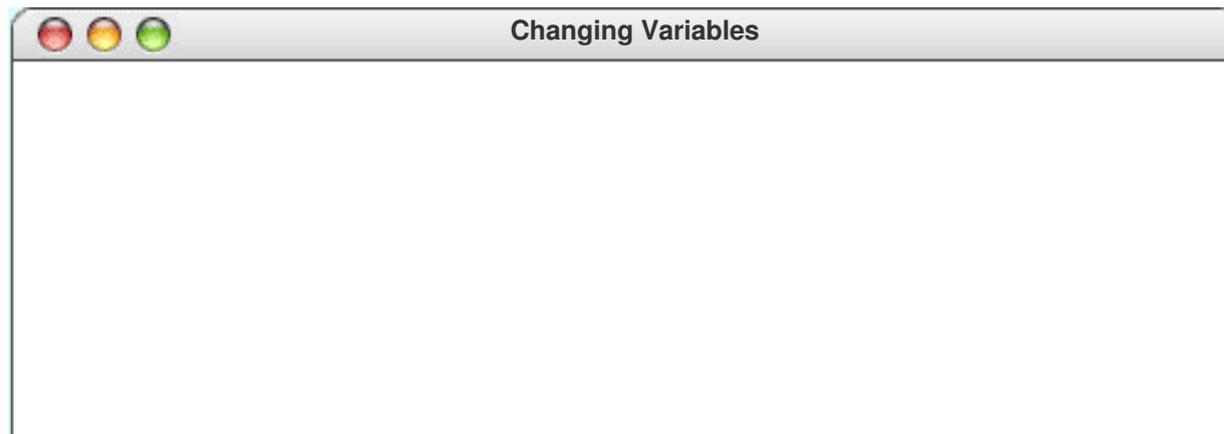
```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
  
}
```

a

5

b

7



Another Example

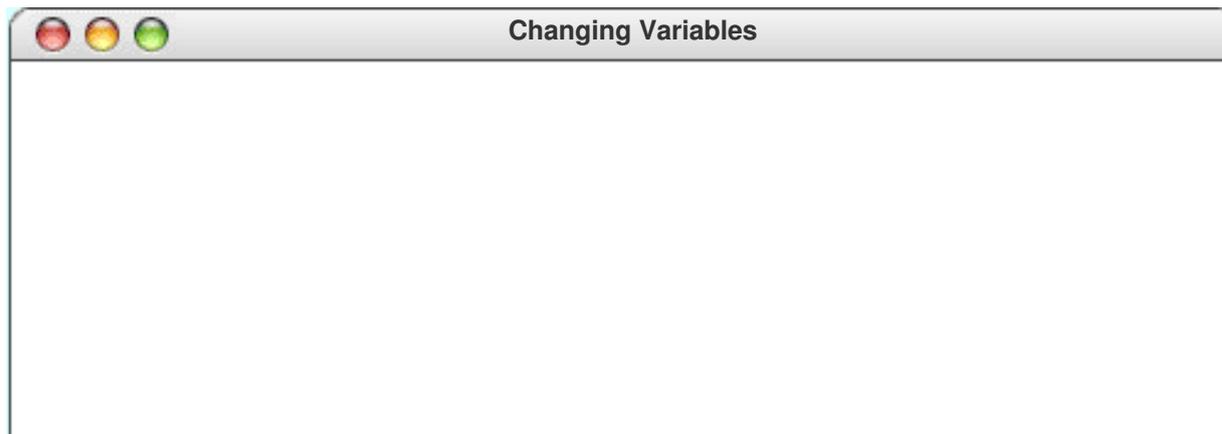
```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
  
}
```

a

10

b

7



Another Example

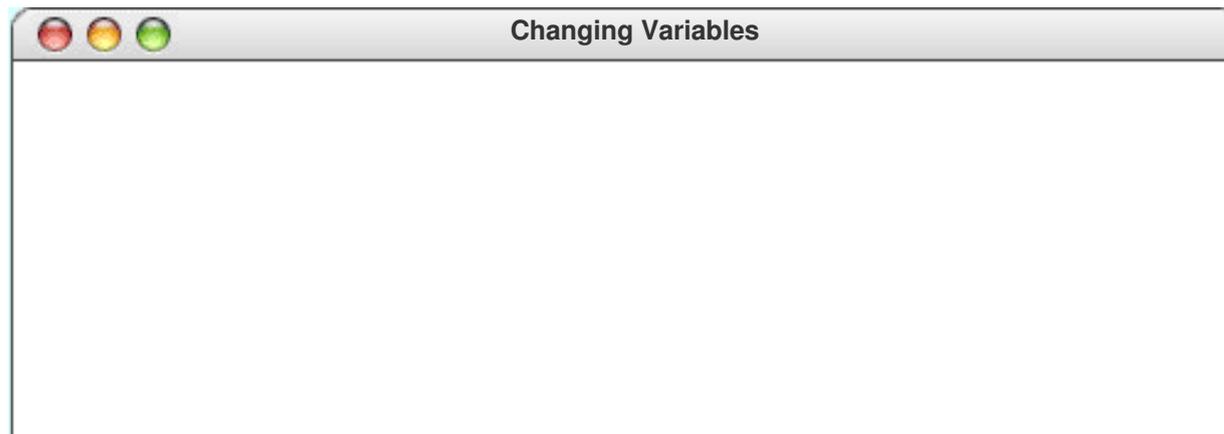
```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
    println(a);  
  
}
```

a

10

b

7



Another Example

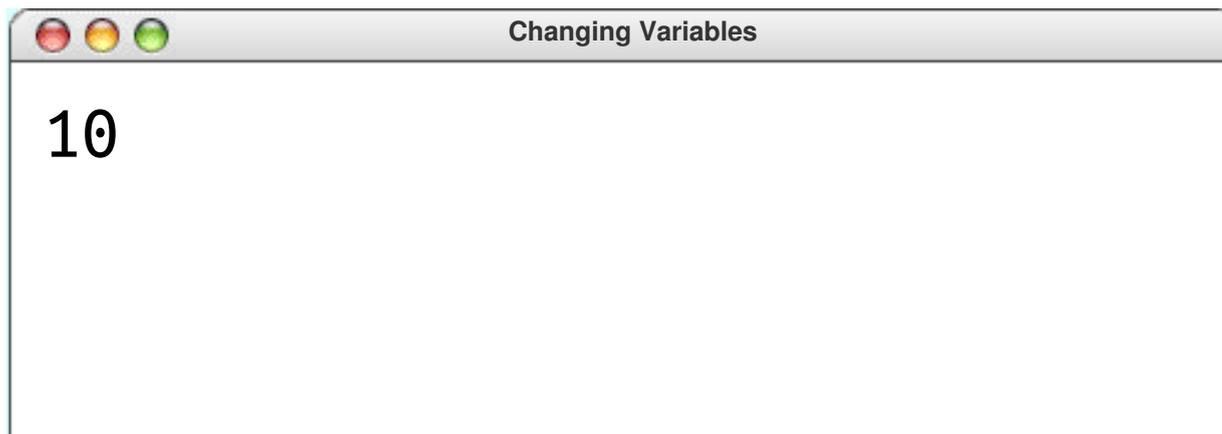
```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
    println(a);  
  
}
```

a

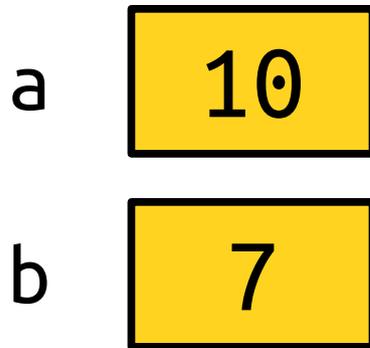
10

b

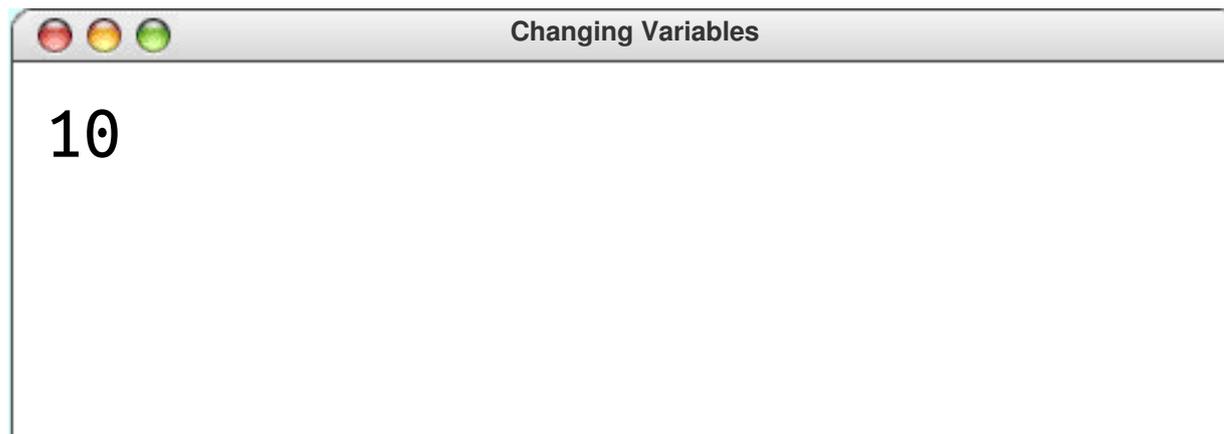
7



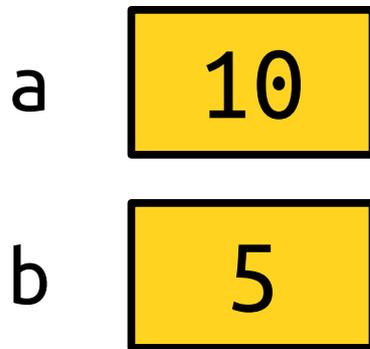
Another Example



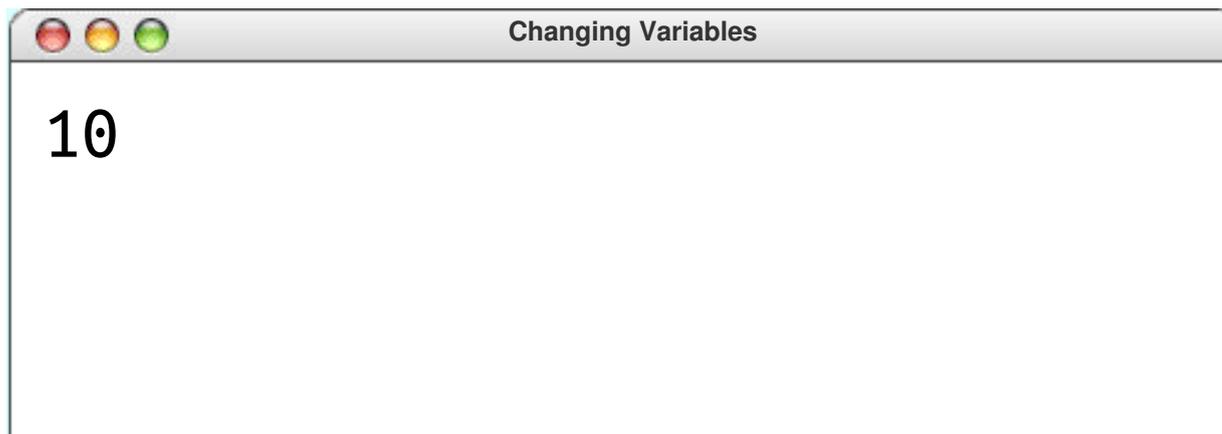
```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
    println(a);  
  
    b = 5;  
  
}
```



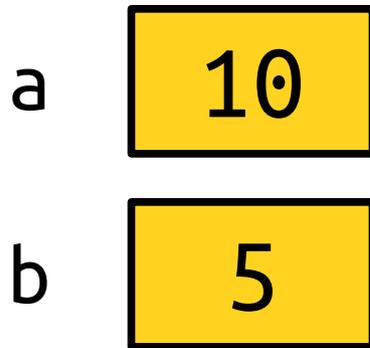
Another Example



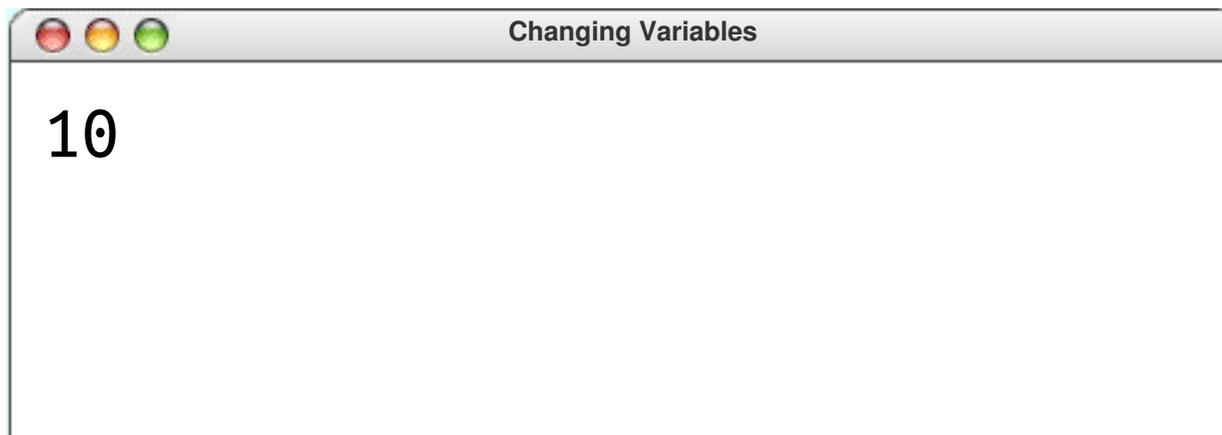
```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
    println(a);  
  
    b = 5;  
  
}
```



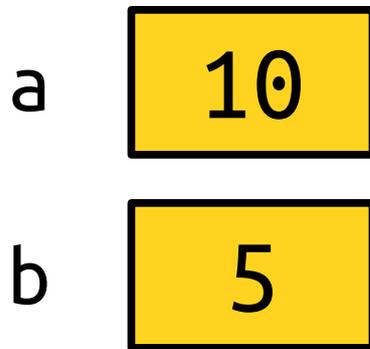
Another Example



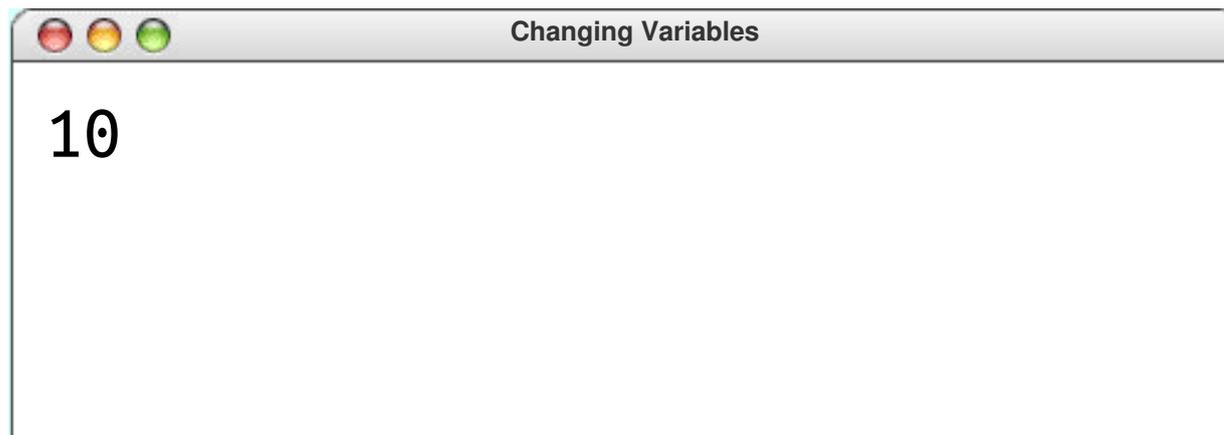
```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
    println(a);  
  
    b = 5;  
  
}
```



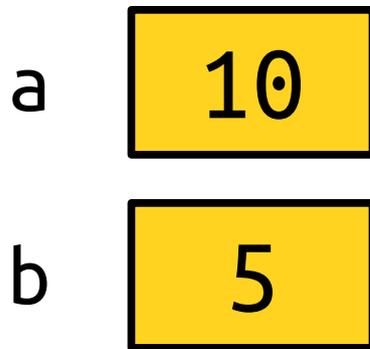
Another Example



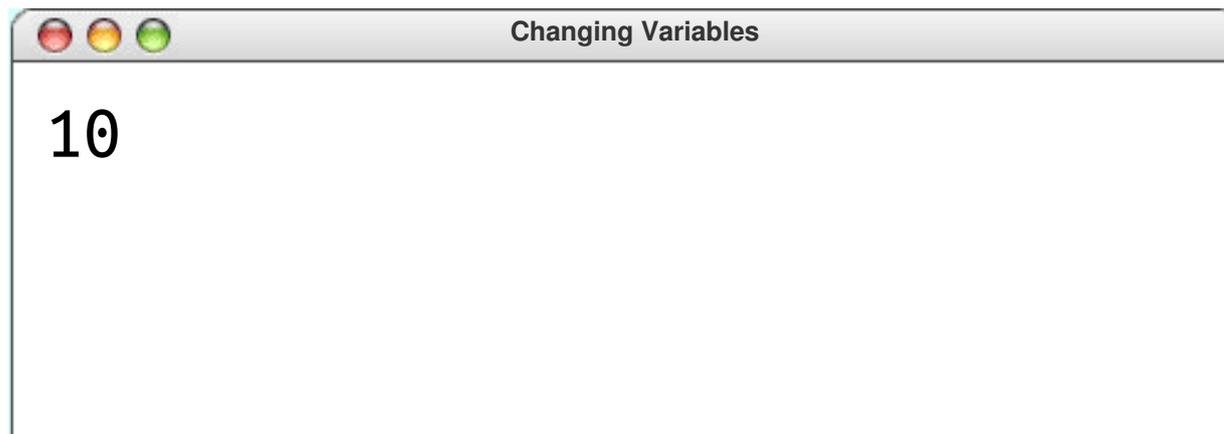
```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
    println(a);  
  
    b = 5;  
  
}
```



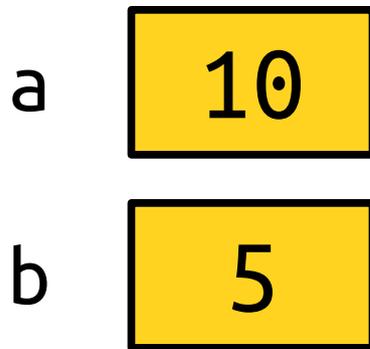
Another Example



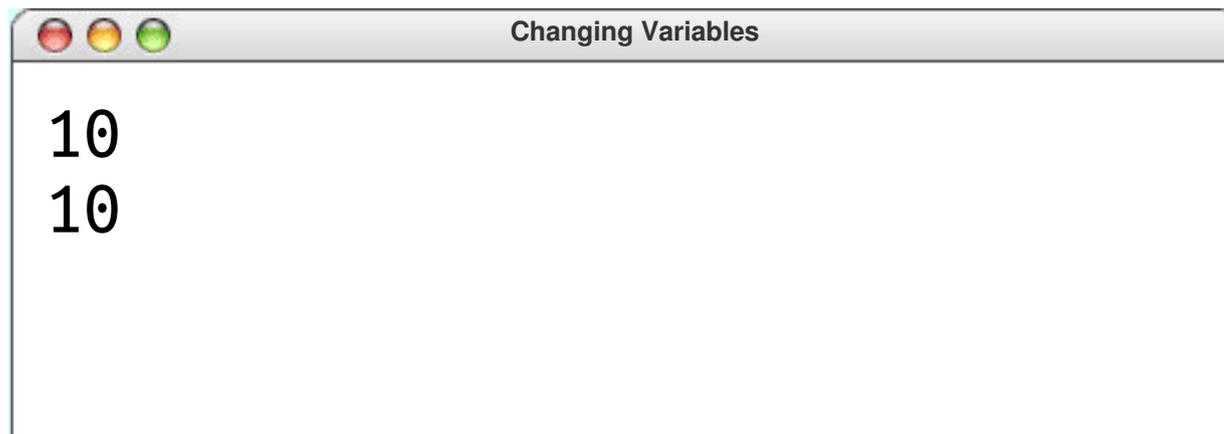
```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
    println(a);  
  
    b = 5;  
    println(a);  
}
```



Another Example



```
public void run() {  
    int a = 5;  
    int b = 7;  
  
    a = b + 3;  
    println(a);  
  
    b = 5;  
    println(a);  
}
```



Another Example

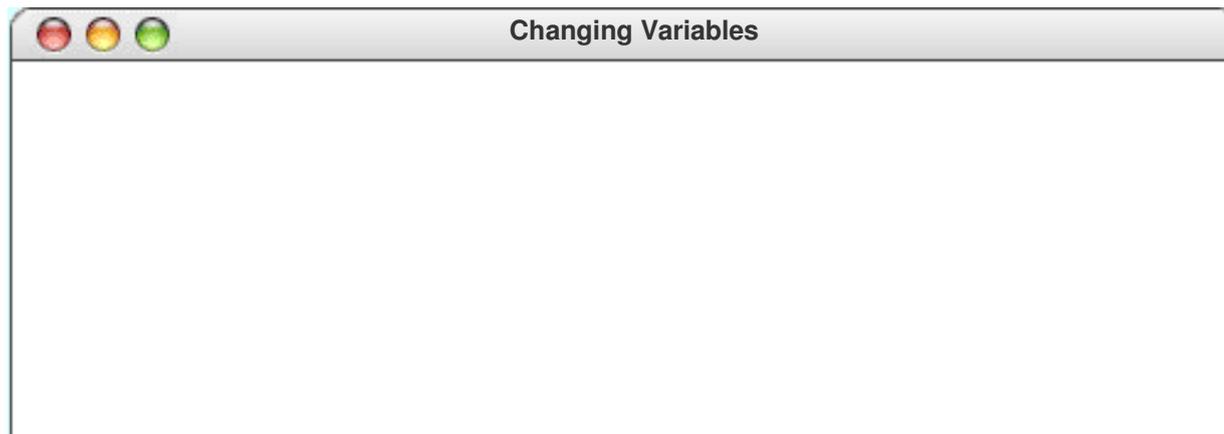
```
public void run() {
```

```
}
```

Another Example

```
public void run() {
```

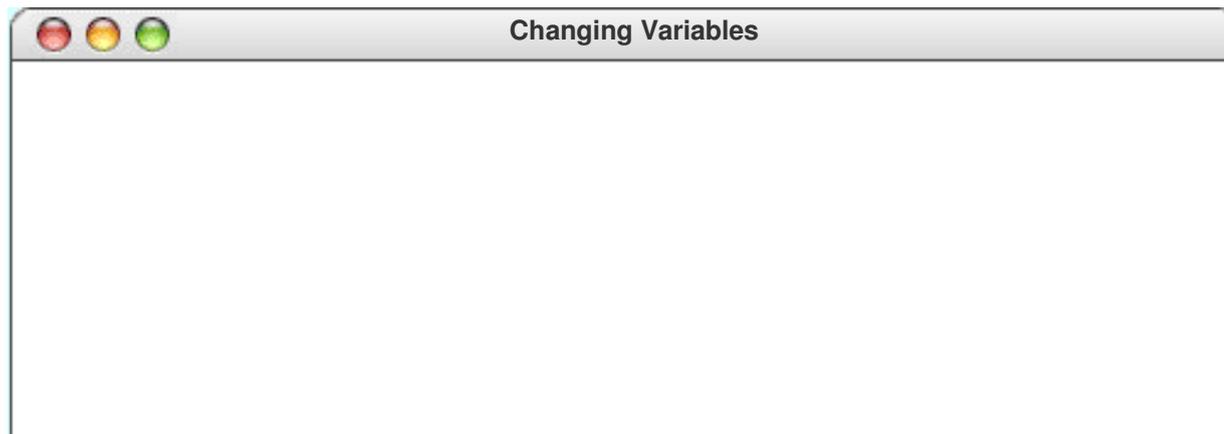
```
}
```



Another Example

```
public void run() {  
    int a = 5;
```

```
}
```

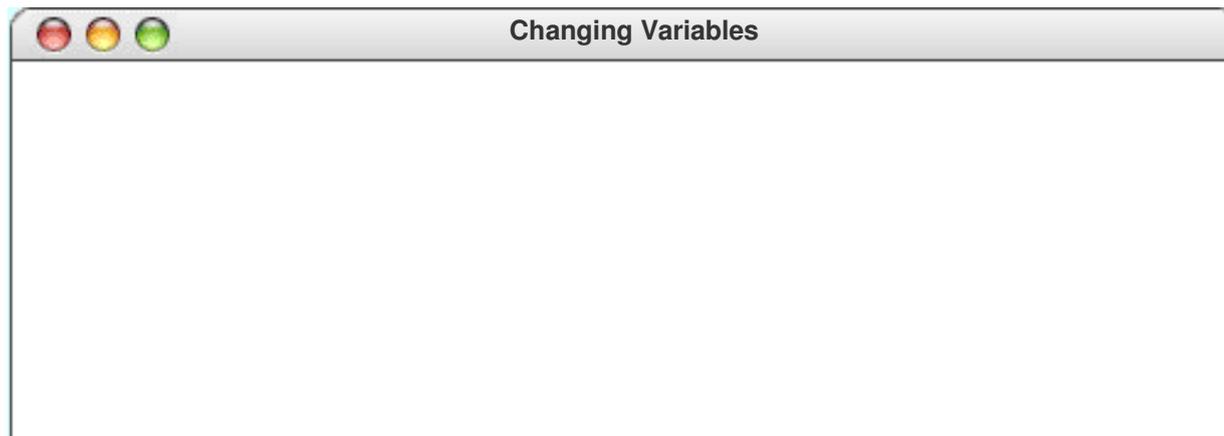


Another Example

```
public void run() {  
    int a = 5;
```

a 

```
}
```

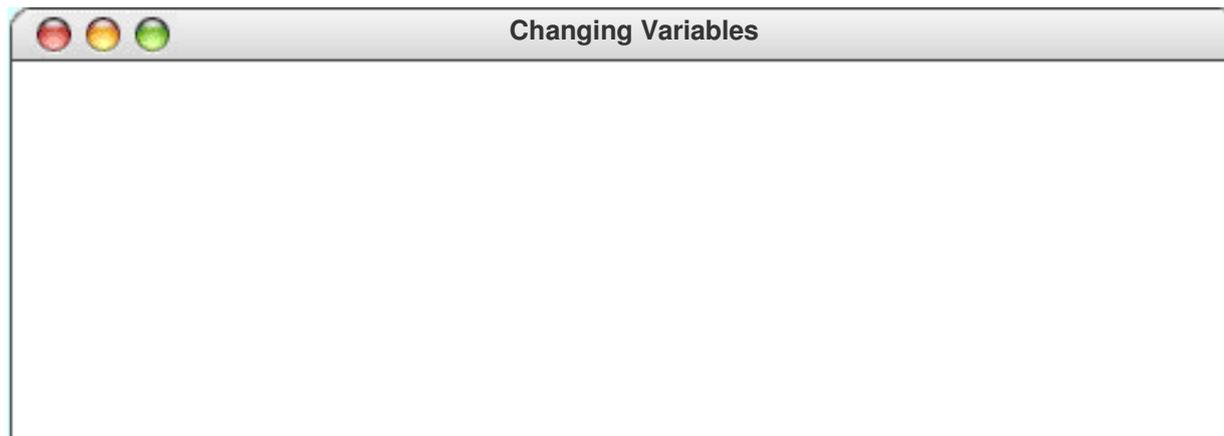


Another Example

```
public void run() {  
    int a = 5;  
    println(a);  
}
```

a 

}

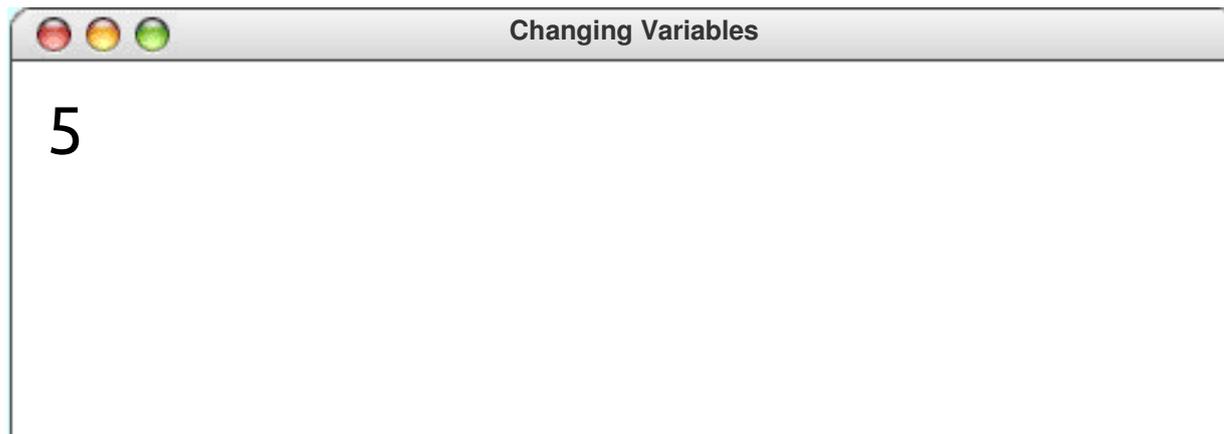


Another Example

```
public void run() {  
    int a = 5;  
    println(a);  
}
```

a 

}



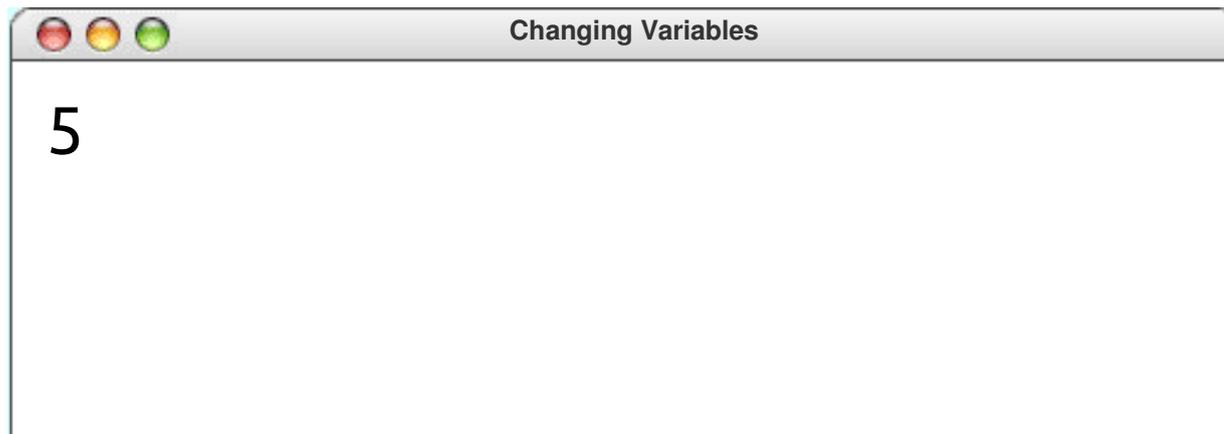
Another Example

```
public void run() {  
    int a = 5;  
    println(a);
```

a 

```
a = a + 1; // <--- Um, what?
```

```
}
```



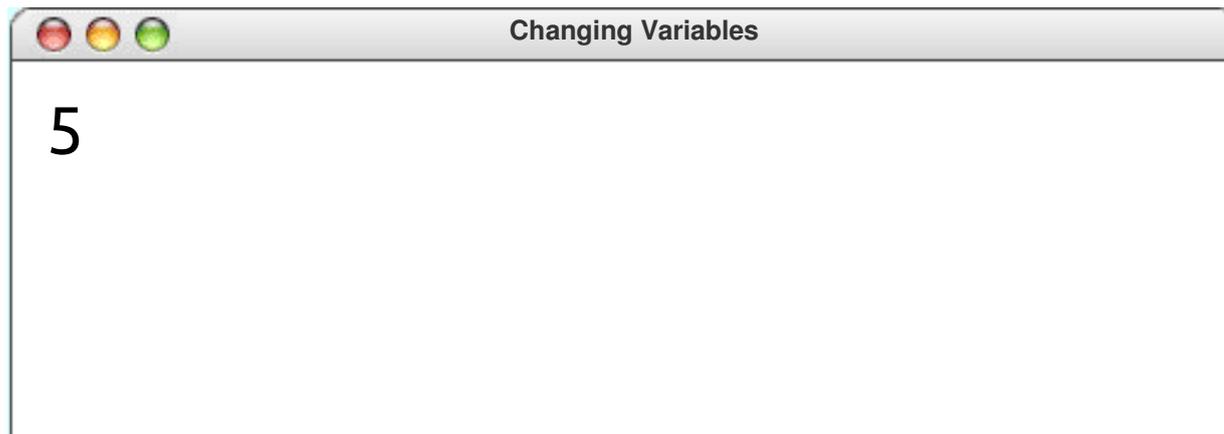
Another Example

```
public void run() {  
    int a = 5;  
    println(a);
```

a 

```
a = a + 1; // <--- Um, what?
```

```
}
```



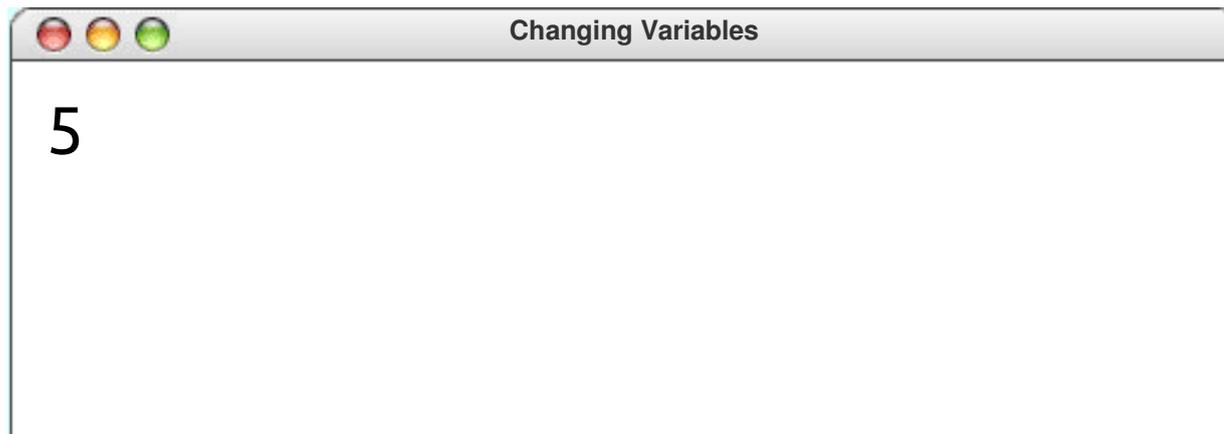
Another Example

```
public void run() {  
    int a = 5;  
    println(a);
```

a 

```
    a = a + 1; // <--- Um, what?  
    println(a);
```

```
}
```



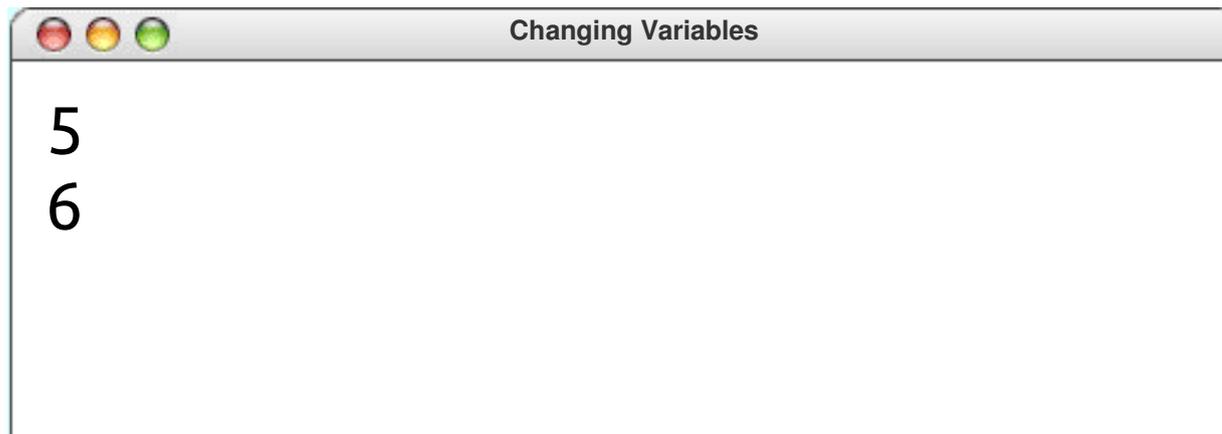
Another Example

```
public void run() {  
    int a = 5;  
    println(a);
```

a 

```
    a = a + 1; // <--- Um, what?  
    println(a);
```

```
}
```



Another Example

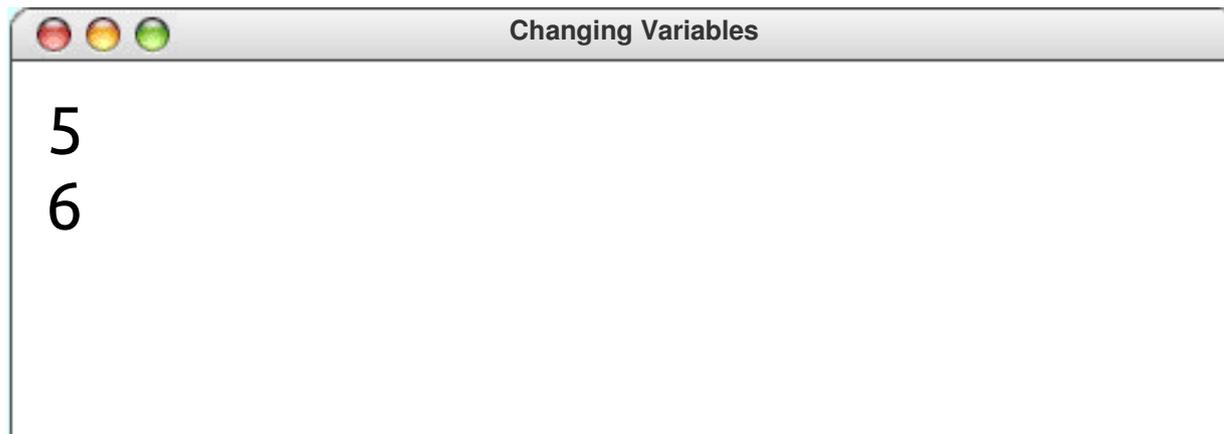
```
public void run() {  
    int a = 5;  
    println(a);
```

a 

```
    a = a + 1; // <--- Um, what?  
    println(a);
```

```
    a = a * 2; // <--- Seriously?
```

```
}
```



Another Example

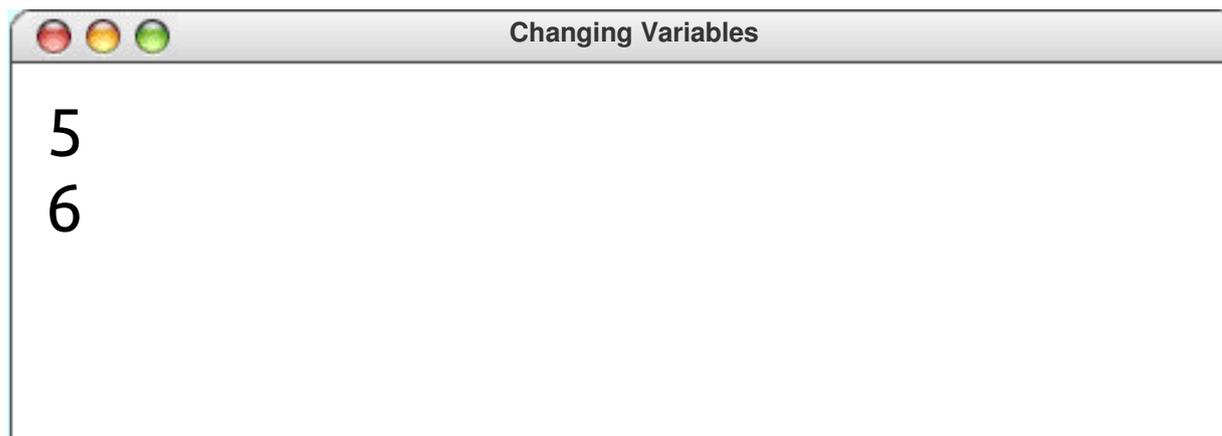
```
public void run() {  
    int a = 5;  
    println(a);
```

a 12

```
    a = a + 1; // <--- Um, what?  
    println(a);
```

```
    a = a * 2; // <--- Seriously?
```

```
}
```



Another Example

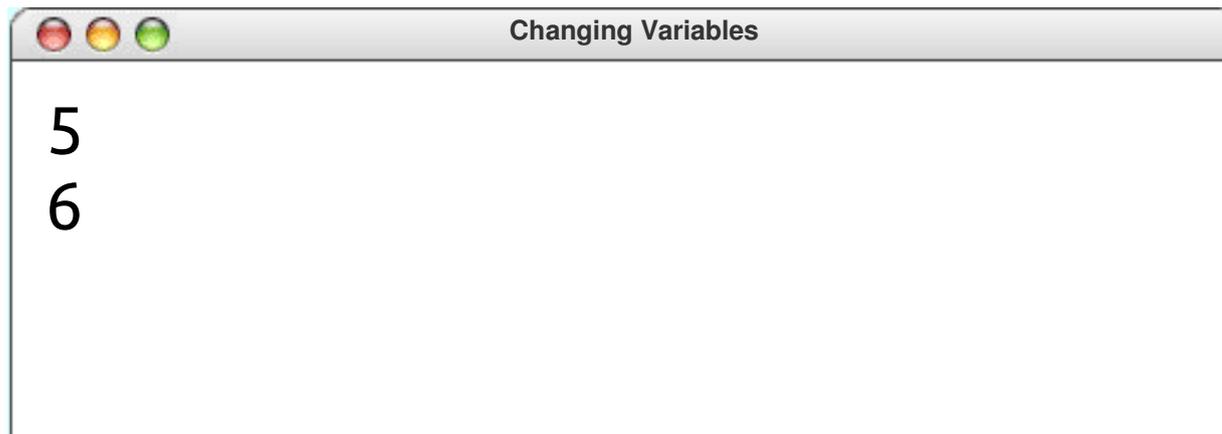
```
public void run() {  
    int a = 5;  
    println(a);
```

a 12

```
    a = a + 1; // <--- Um, what?  
    println(a);
```

```
    a = a * 2; // <--- Seriously?  
    println(a);
```

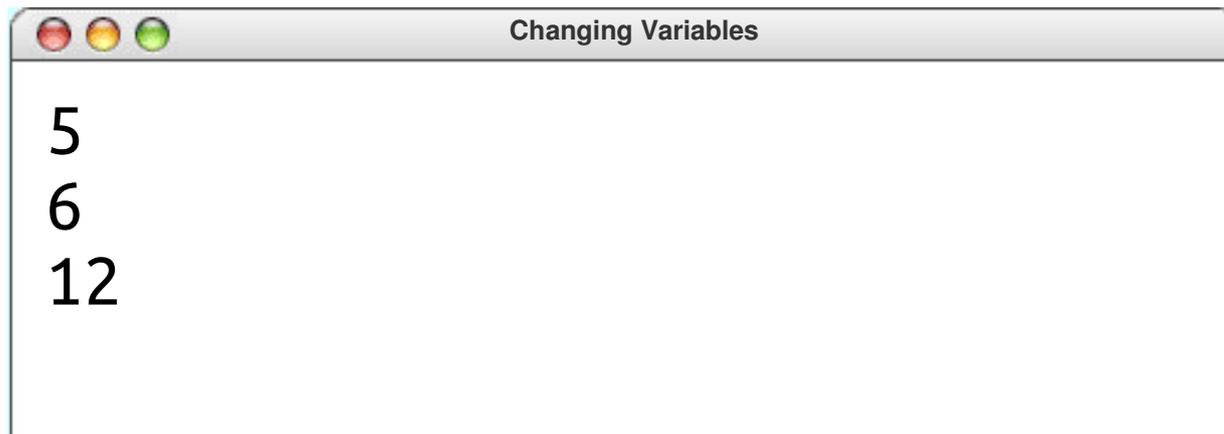
```
}
```



Another Example

```
public void run() {  
    int a = 5;  
    println(a);  
  
    a = a + 1; // <--- Um, what?  
    println(a);  
  
    a = a * 2; // <--- Seriously?  
    println(a);  
}
```

a 12



```
Changing Variables  
5  
6  
12
```

Nudging Values

- In Java (and many other languages), it's normal to see statements like these:

```
x = x + 1;
```

```
y = y / 137;
```

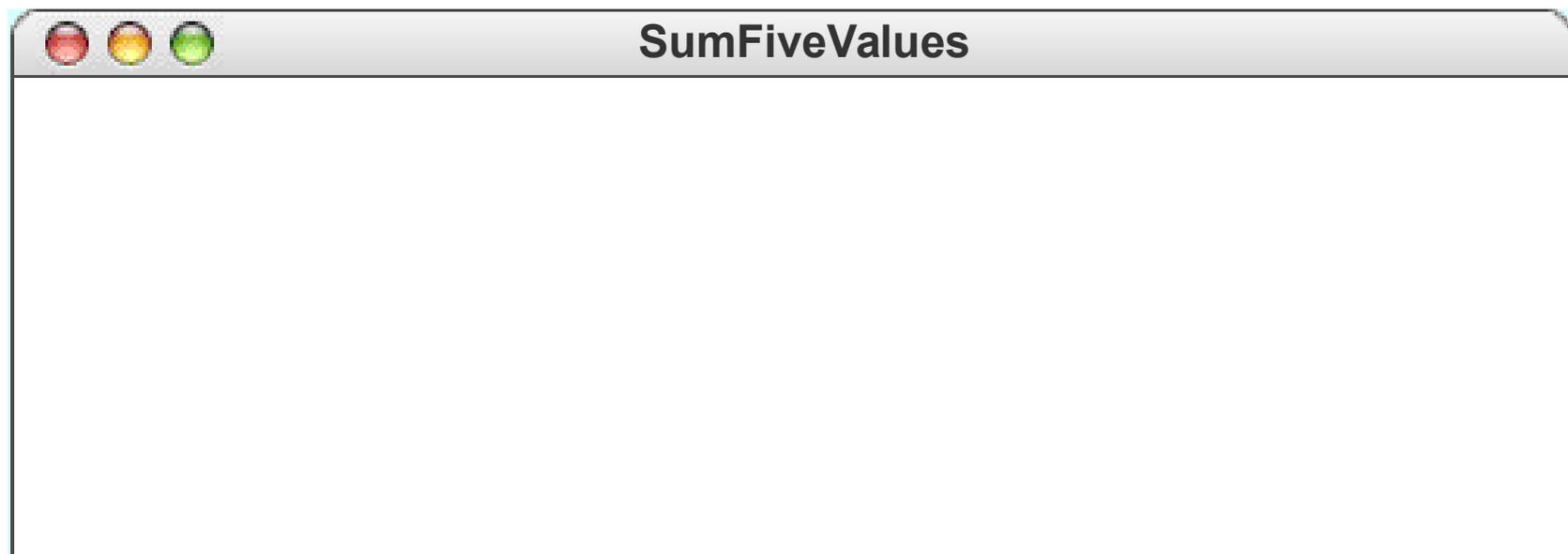
- Don't read these as mathematical statements – you'll just get confused.
- Two intuitions:
 - Read these statements as “add one to x ” or “divide y by 137.”
 - Read these statements as *commands* – we are ordering x and y to update the values they are storing.

Why would you do this?

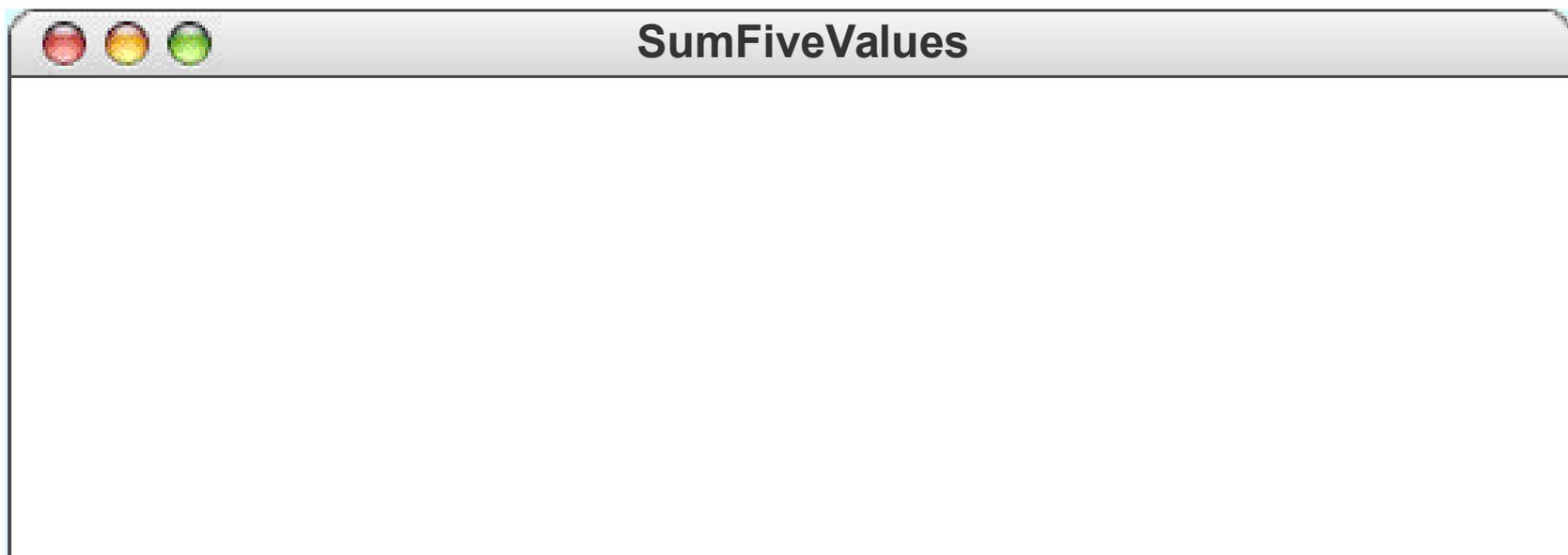
Write a program that reads in a list of five values, then outputs their sum.

```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

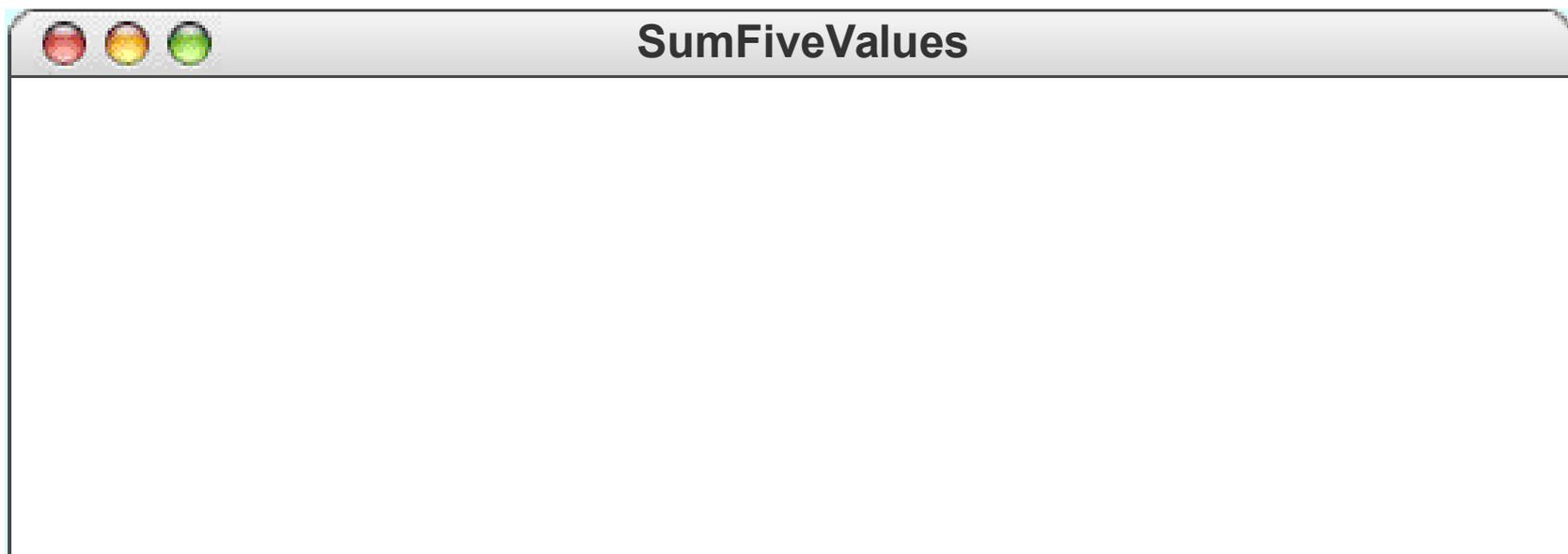


```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```



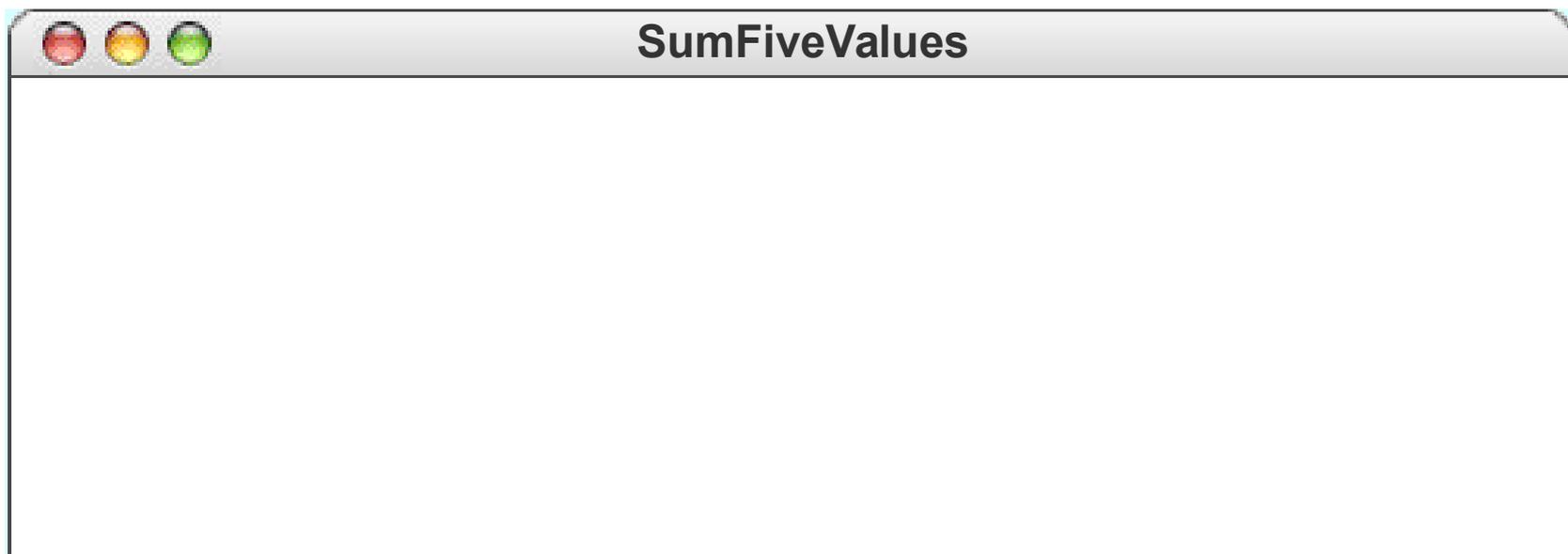
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 0



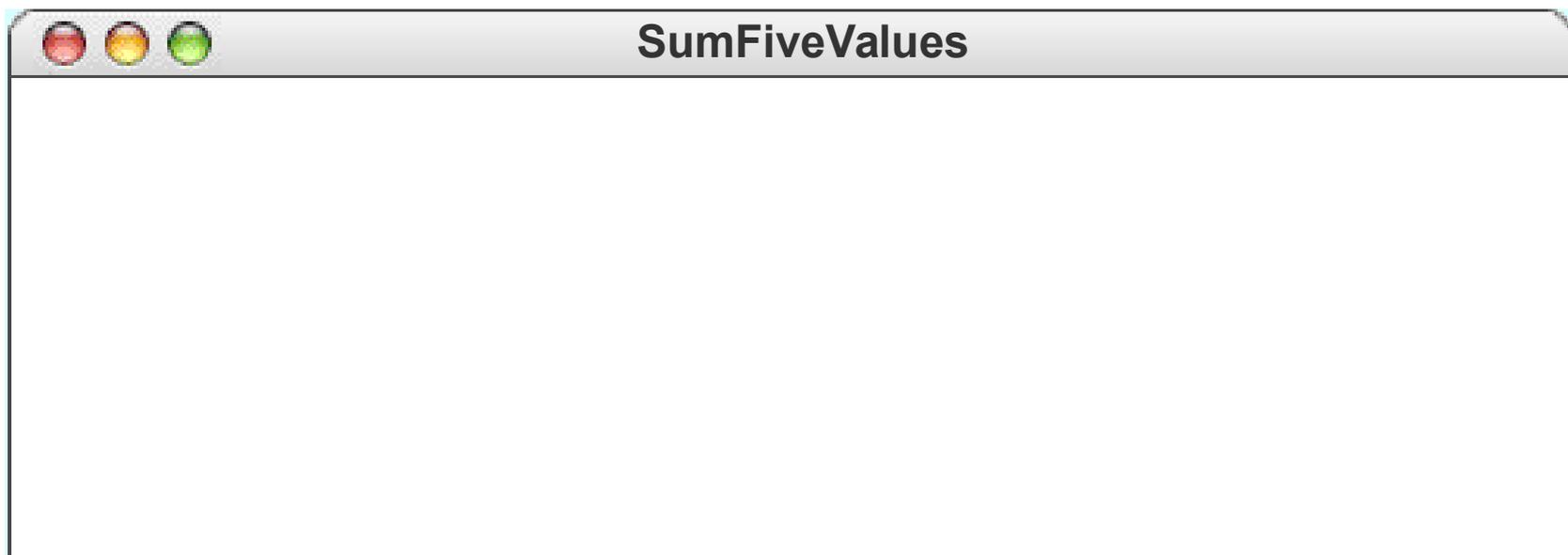
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 0



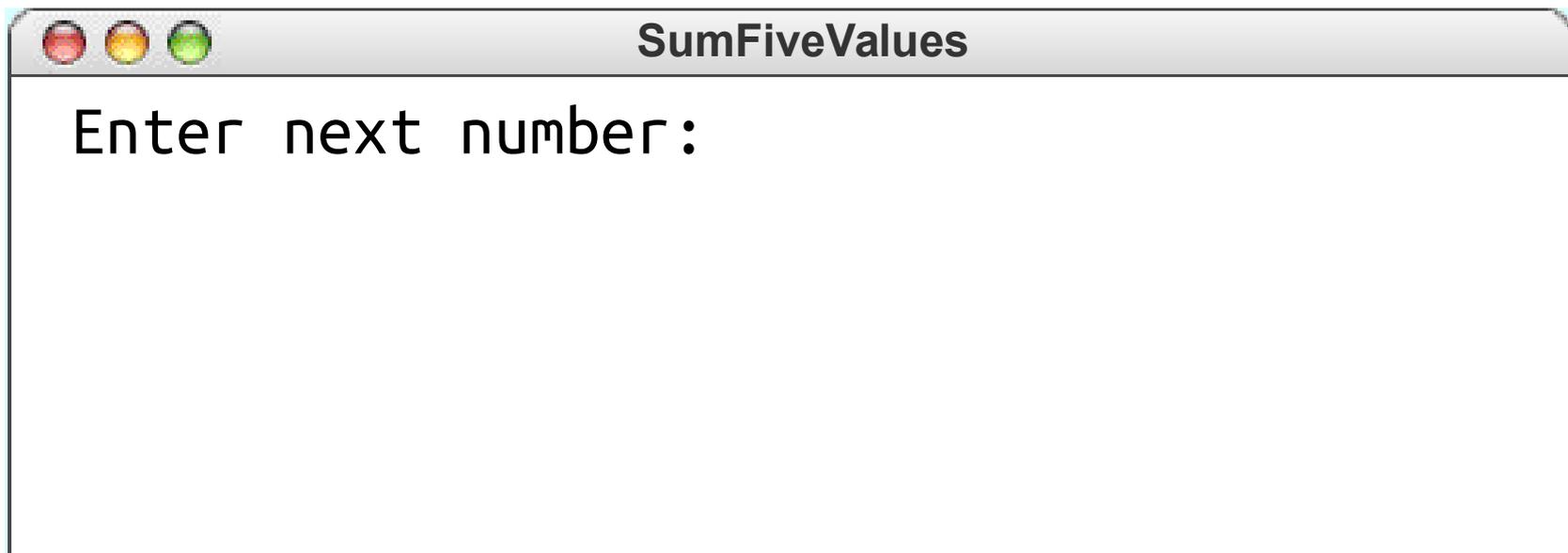
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 0



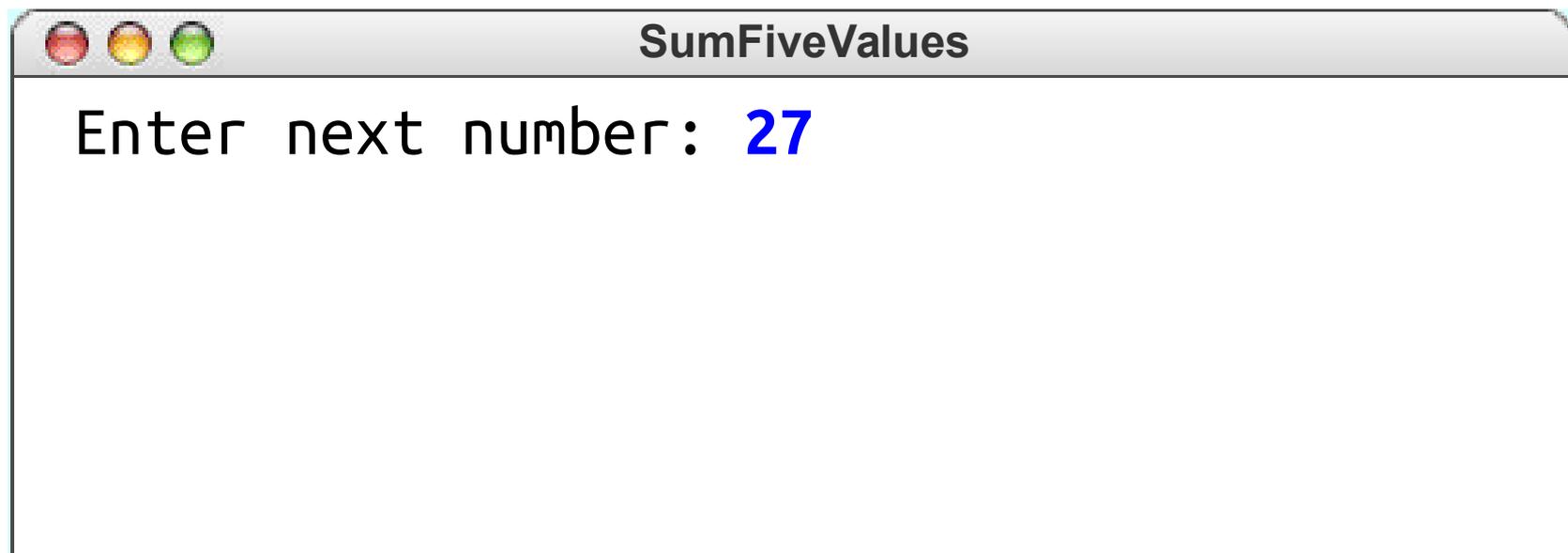
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 0



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

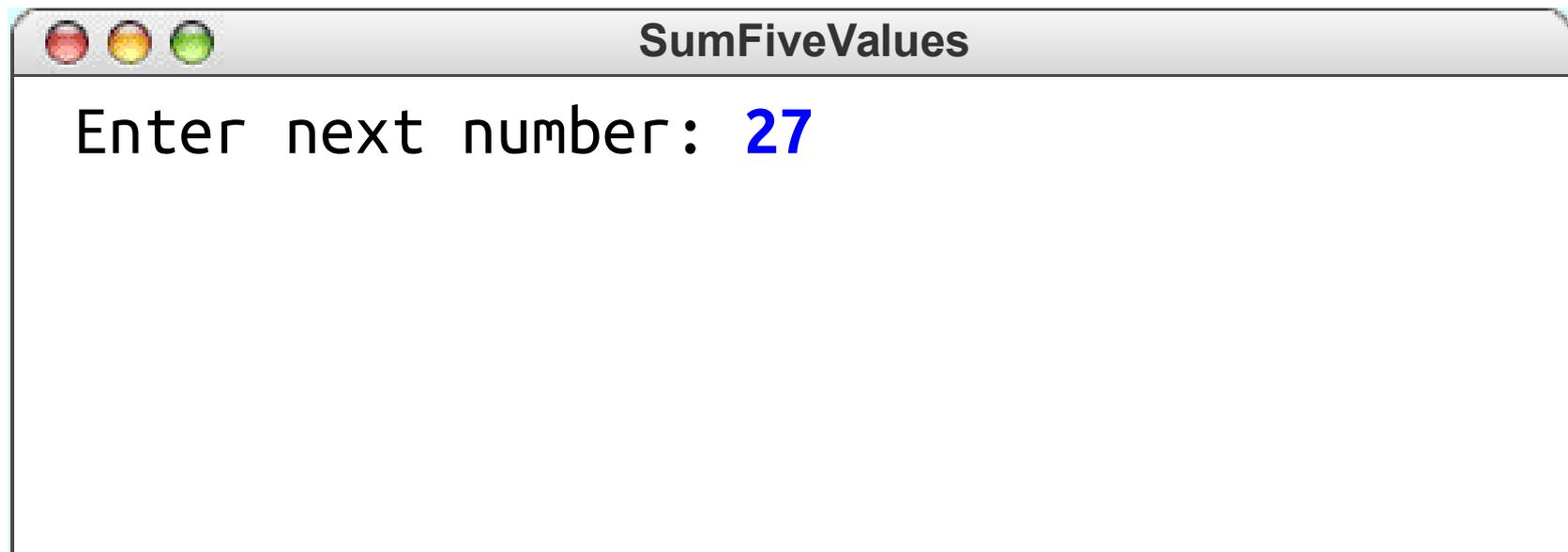
total 0



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 0

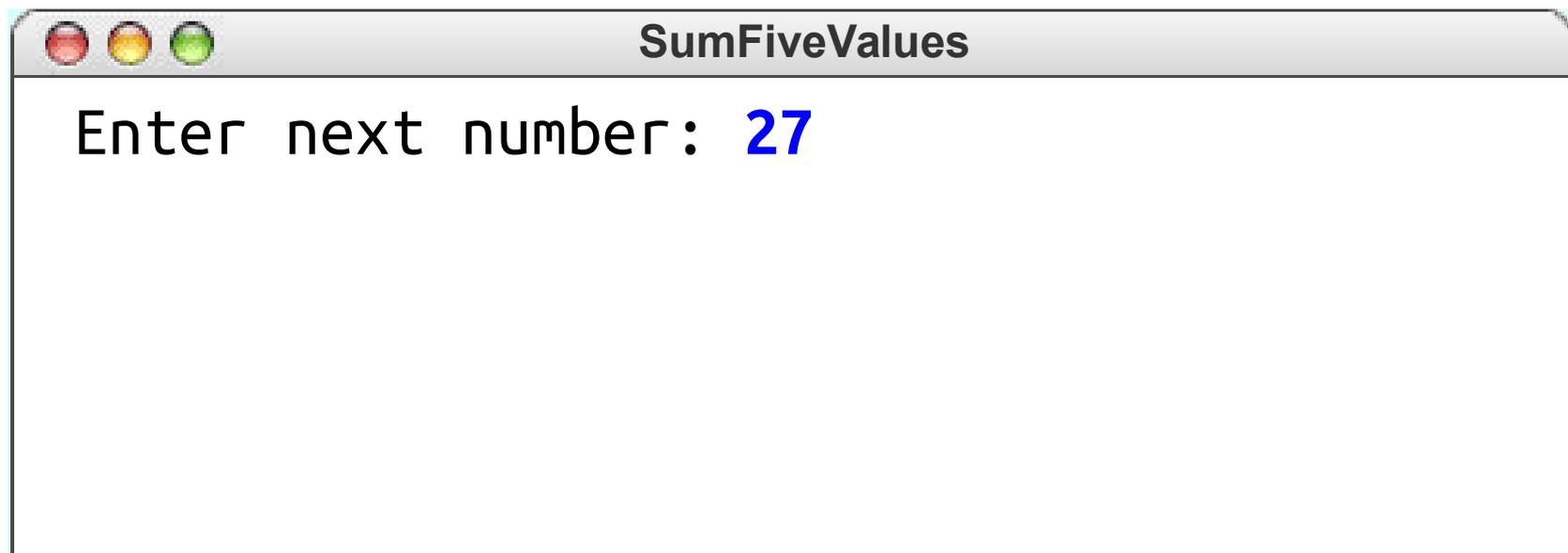
nextValue 27



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 0

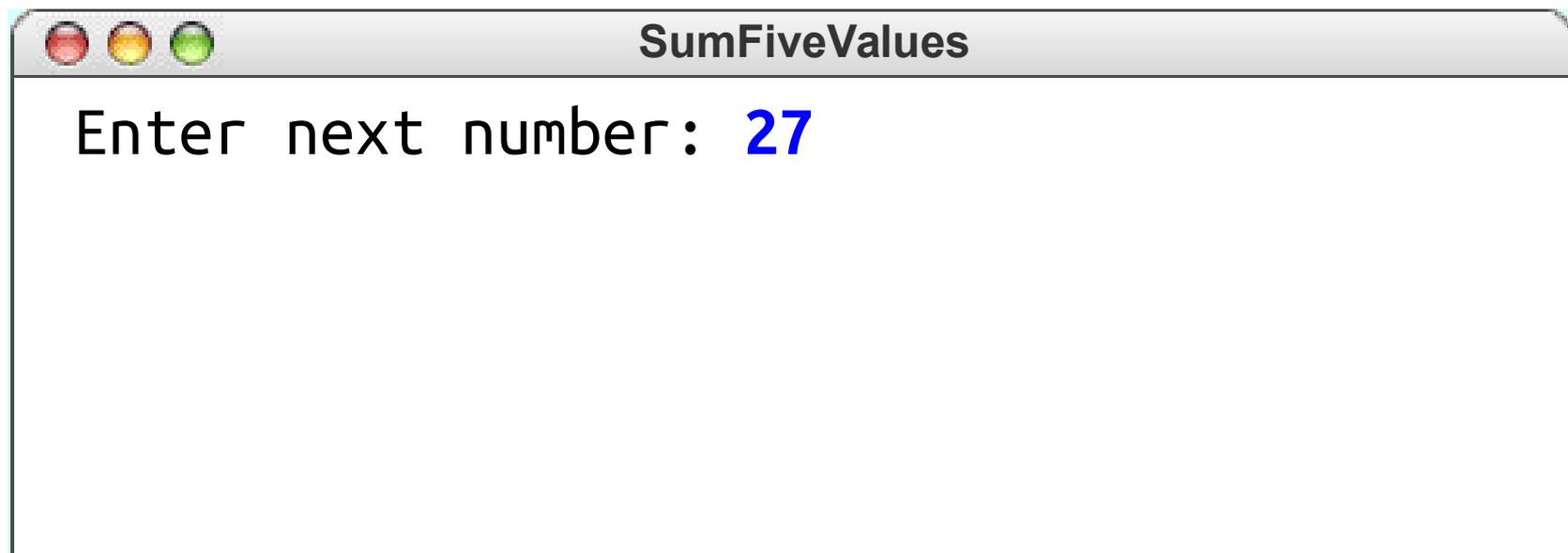
nextValue 27



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

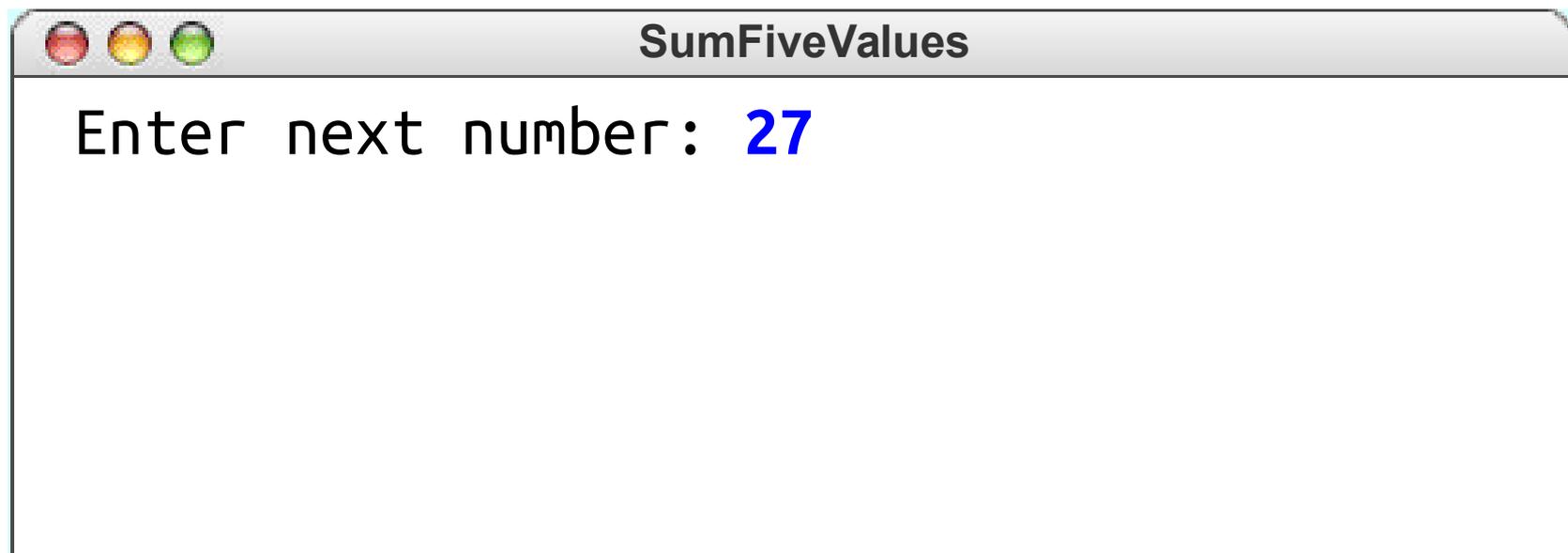
total 27

nextValue 27



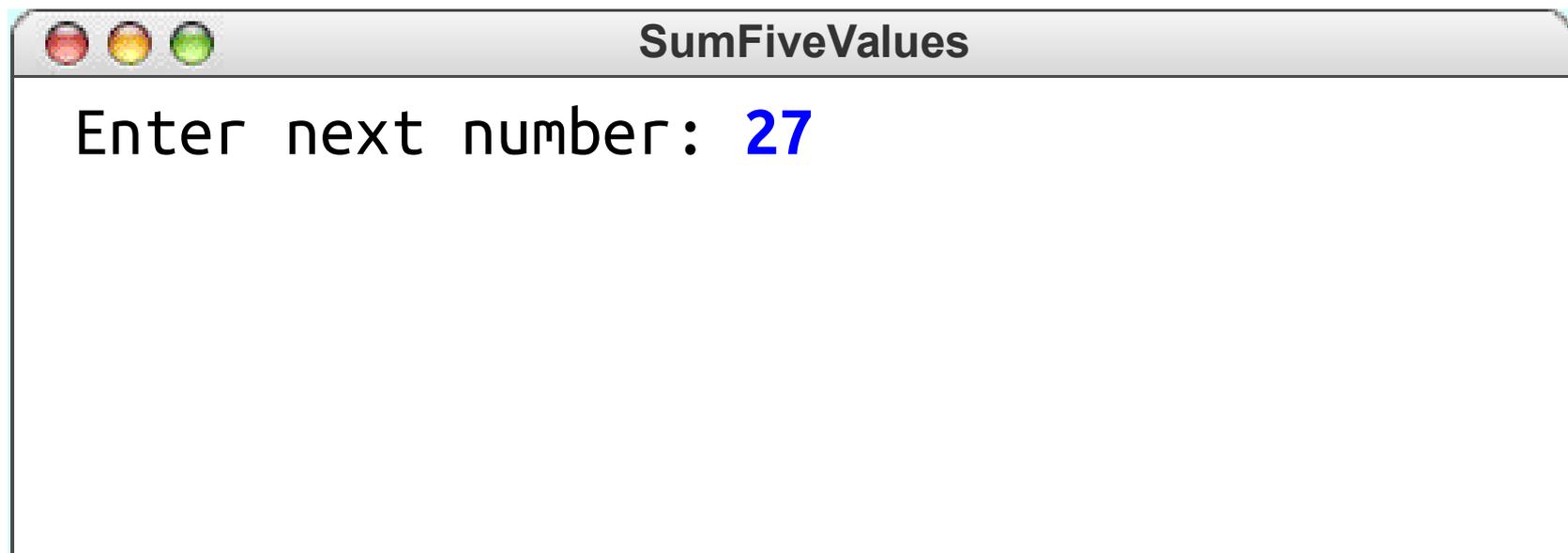
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 27



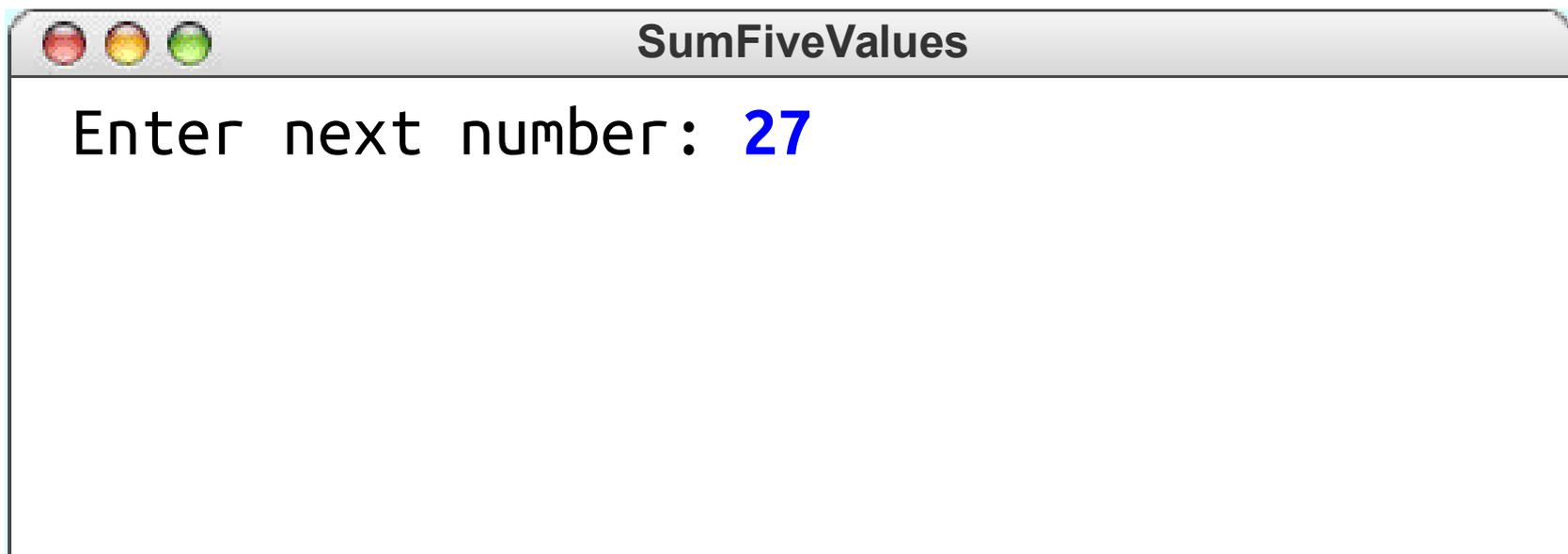
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 27



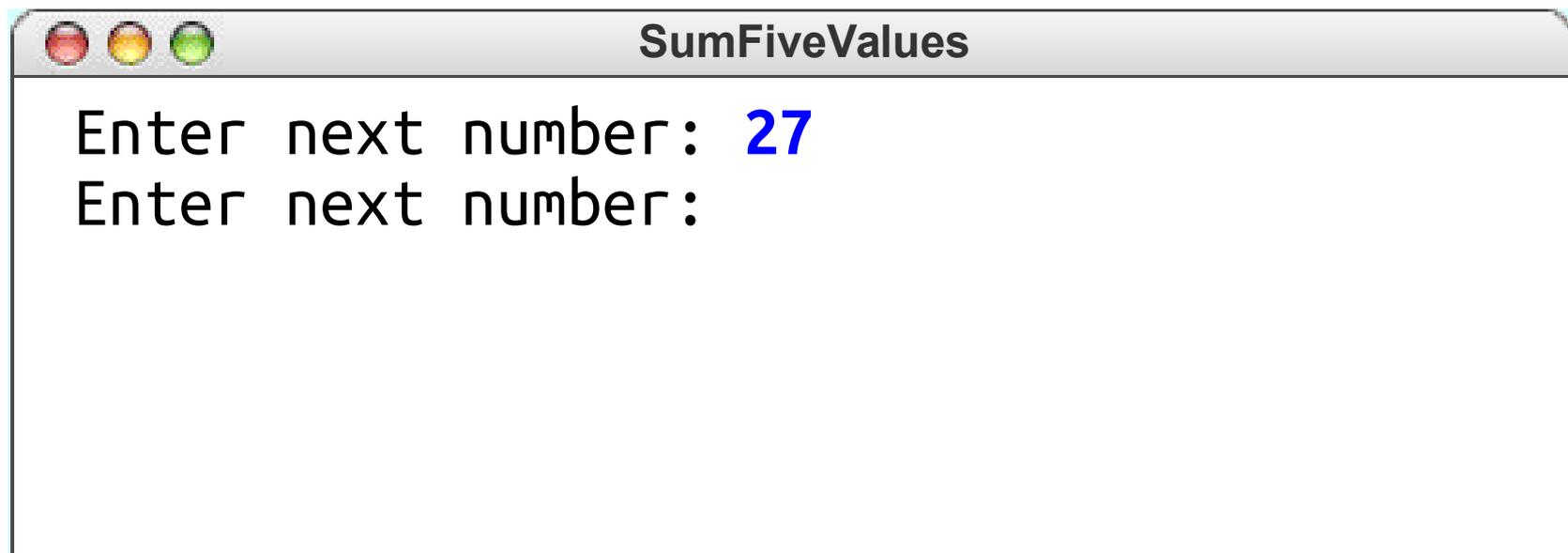
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 27



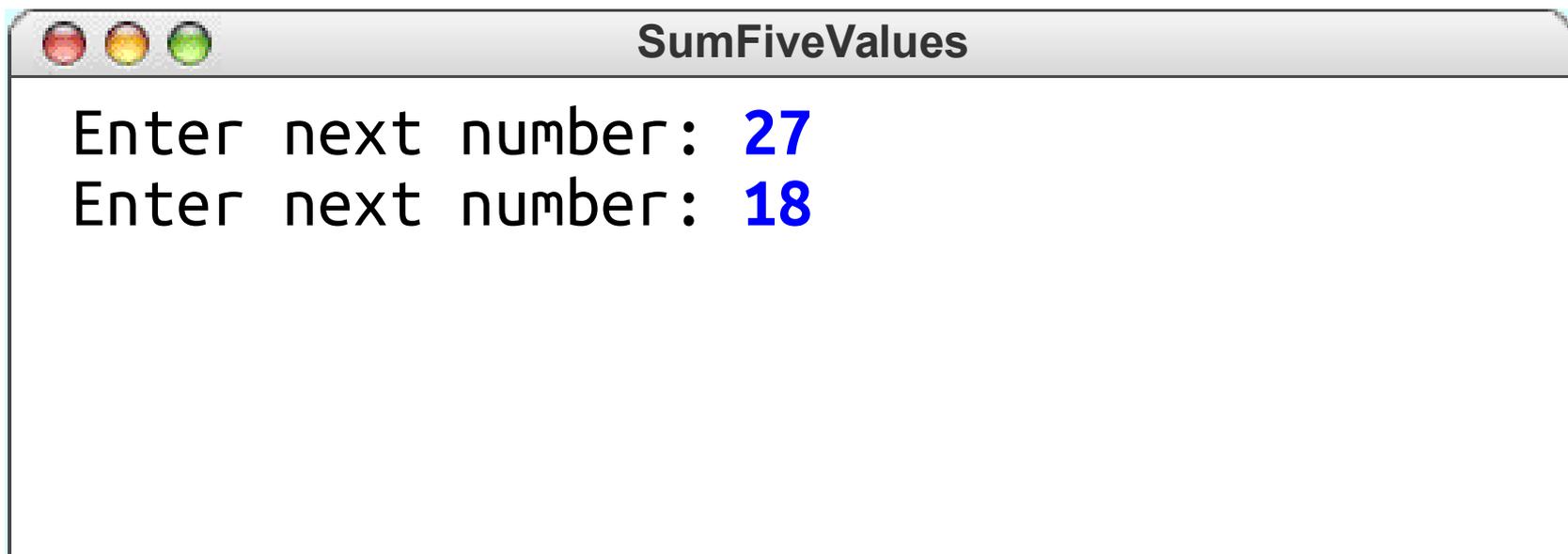
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 27



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

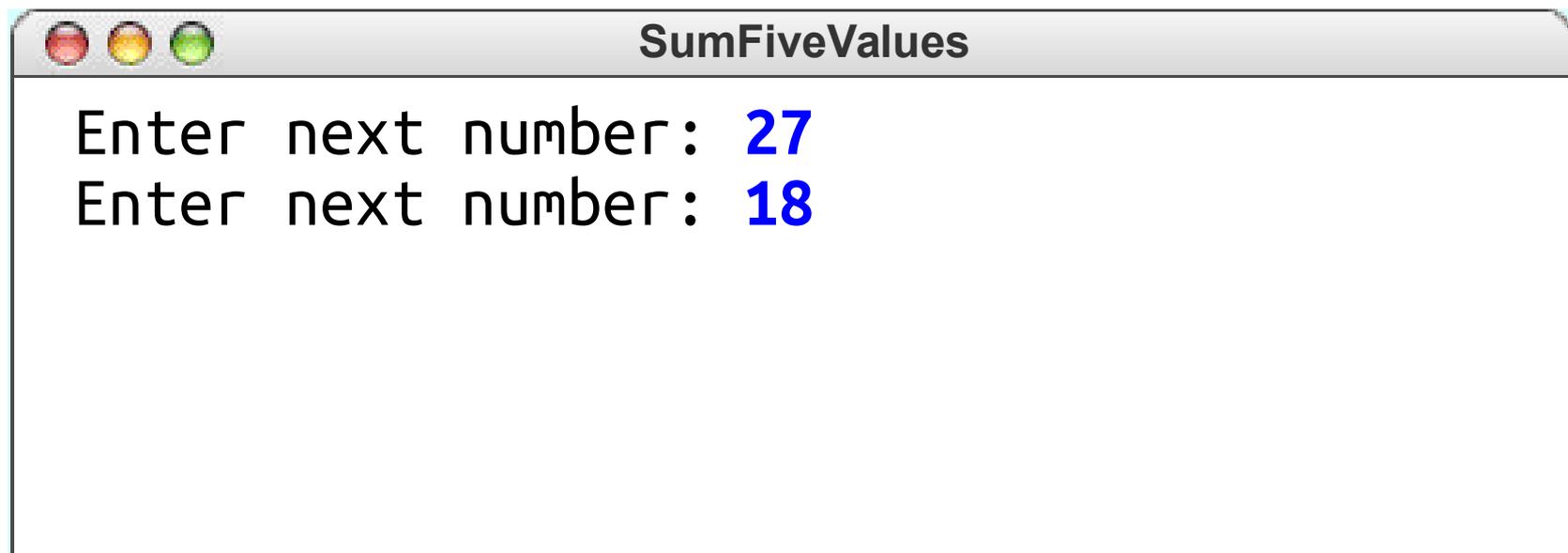
total 27



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 27

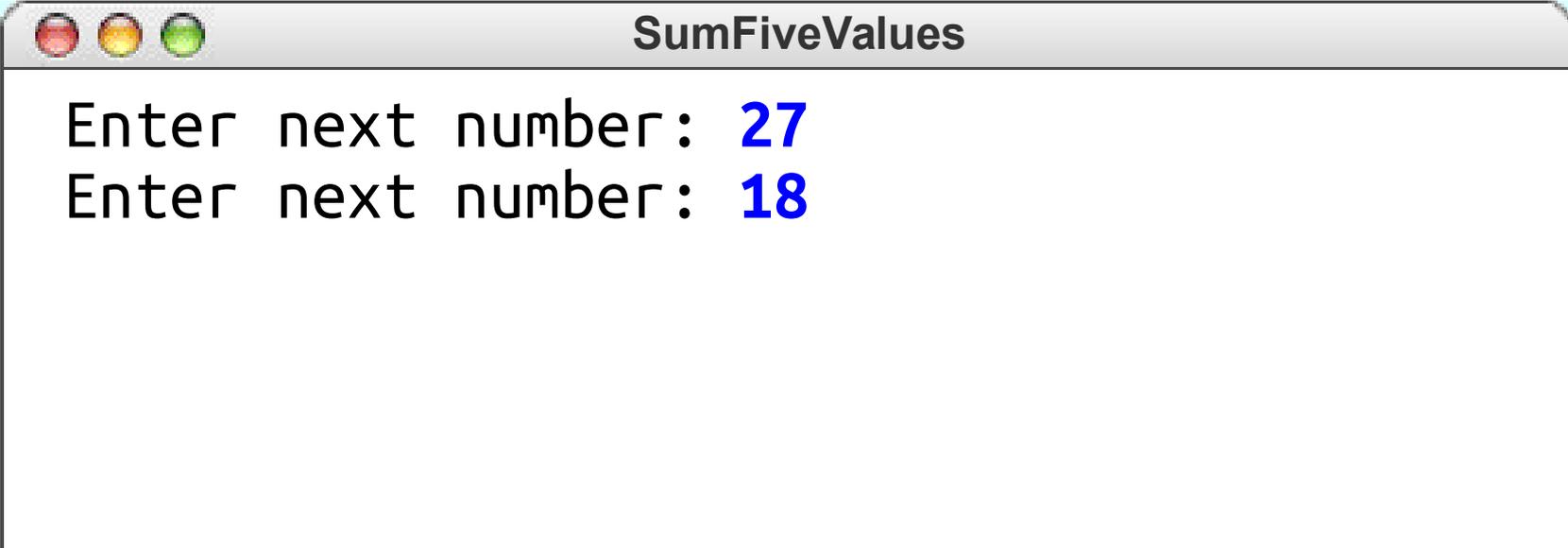
nextValue 18



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 27

nextValue 18

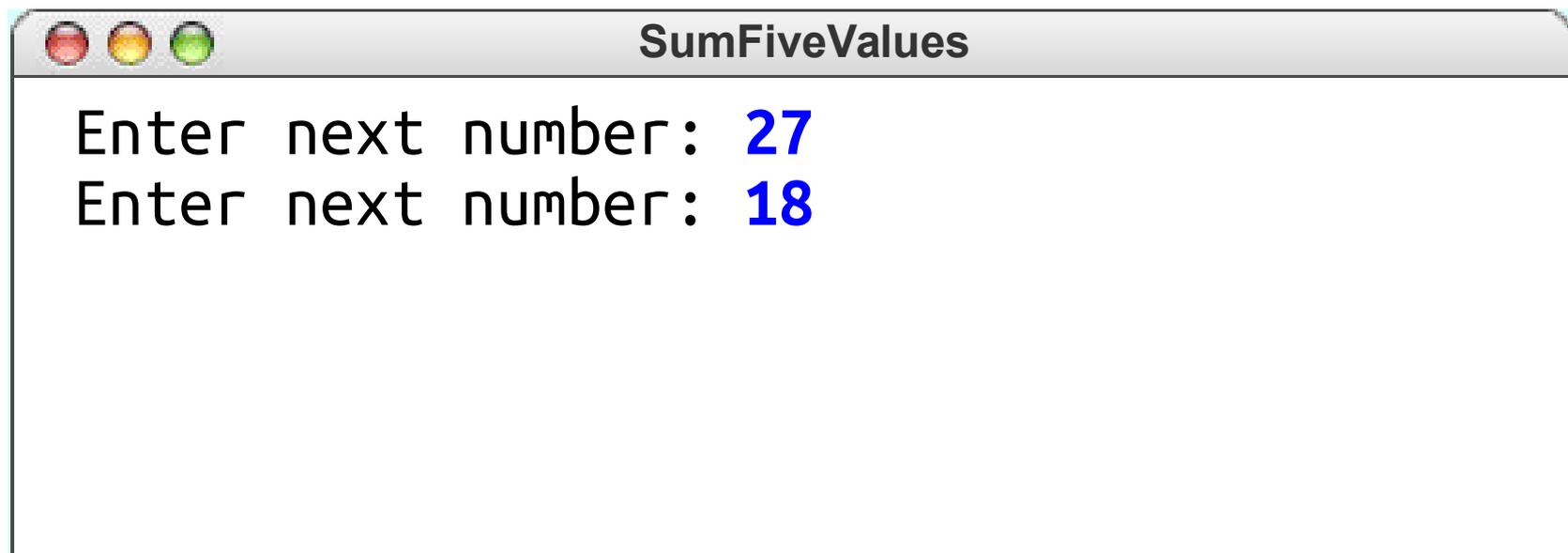


The screenshot shows a window titled "SumFiveValues" with a standard macOS-style title bar (red, yellow, and green buttons). The window contains two lines of text: "Enter next number: 27" and "Enter next number: 18". The numbers 27 and 18 are displayed in a blue font, indicating they are the values entered by the user.

```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

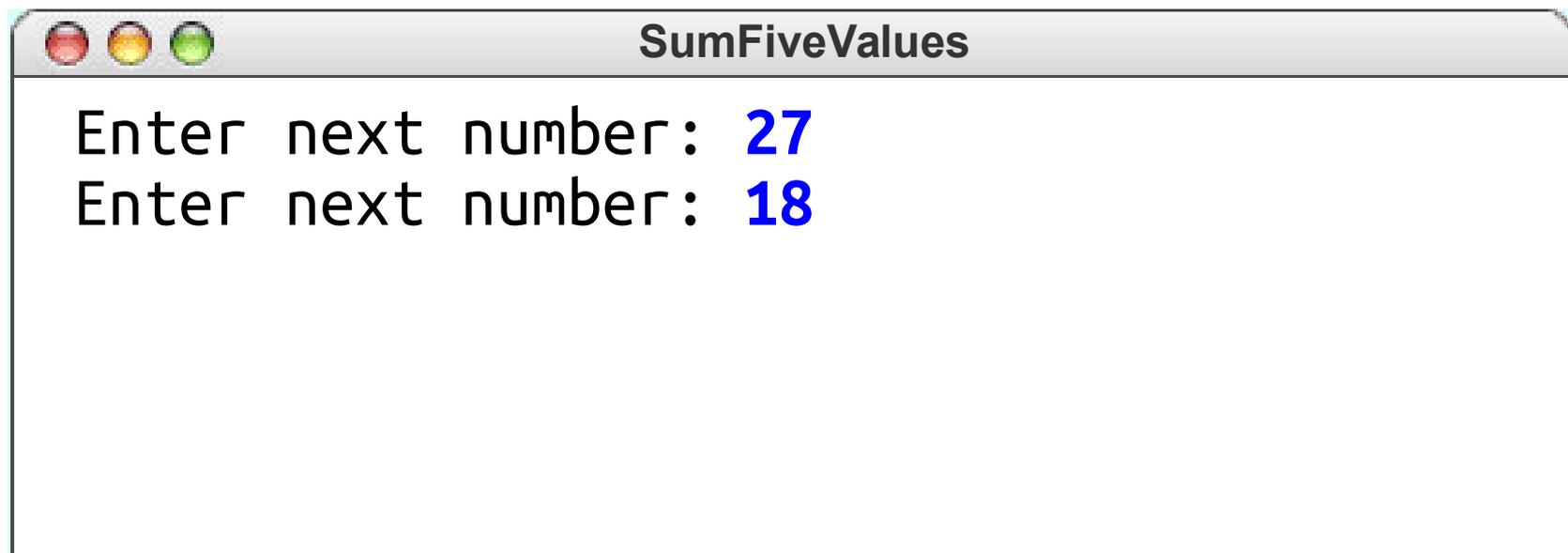
total 45

nextValue 18



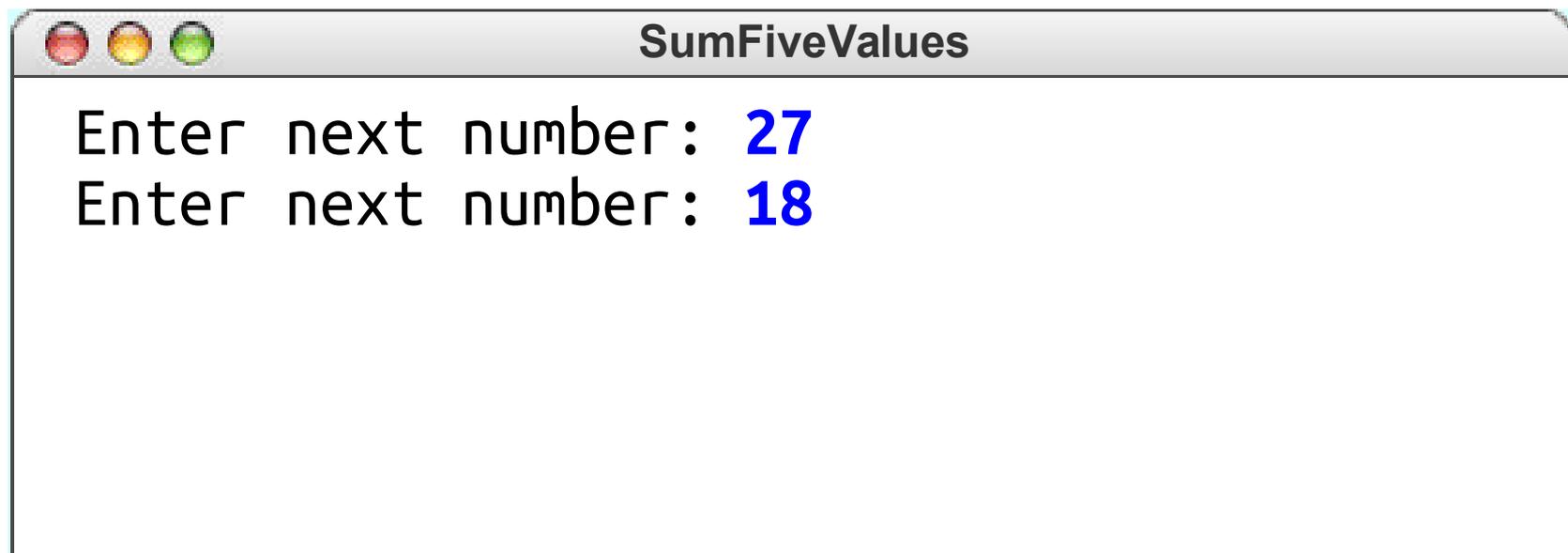
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 45



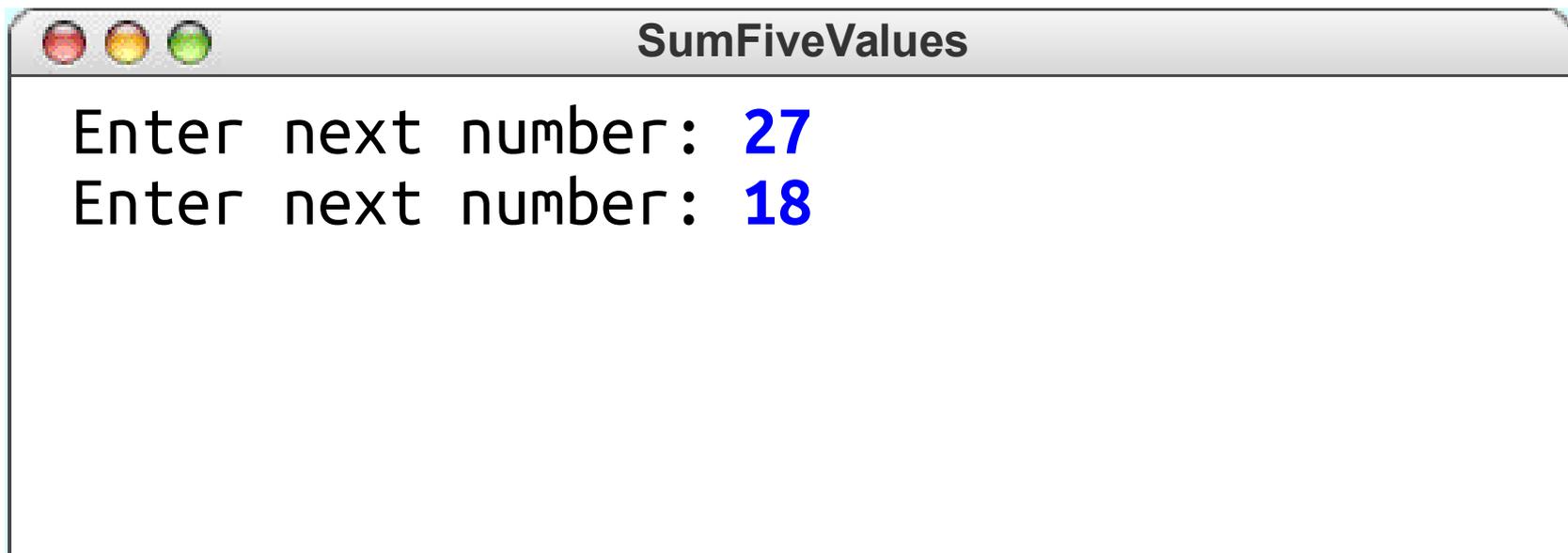
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 45



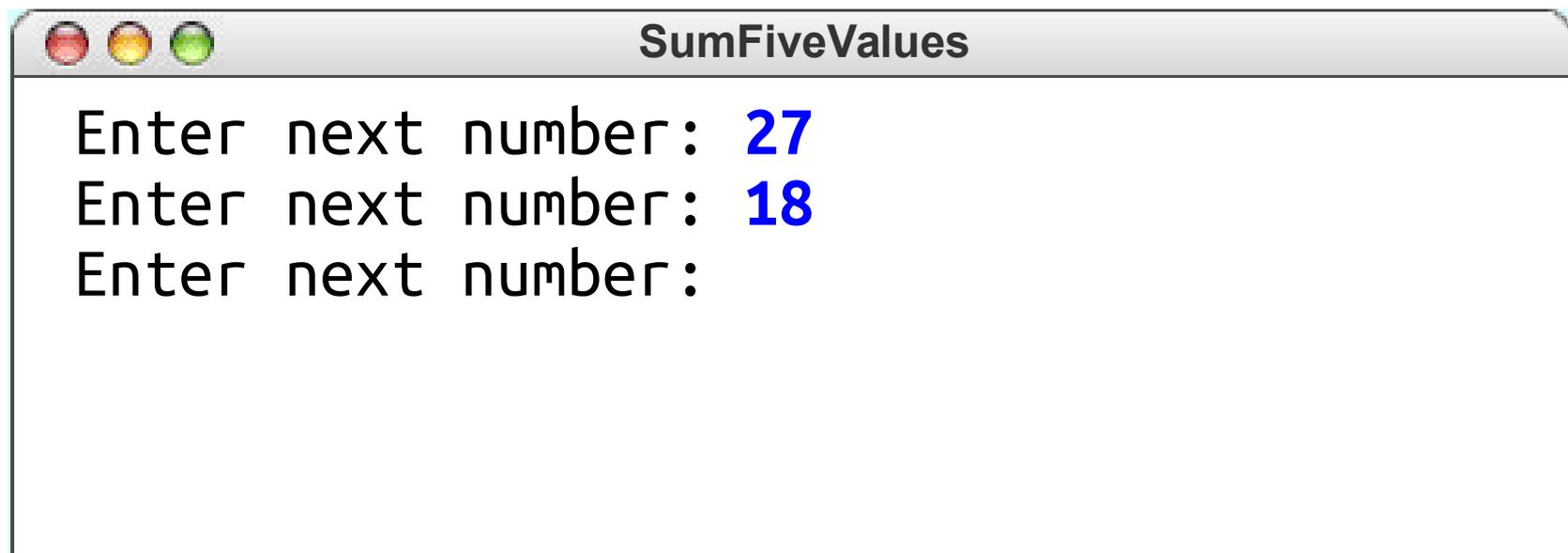
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 45



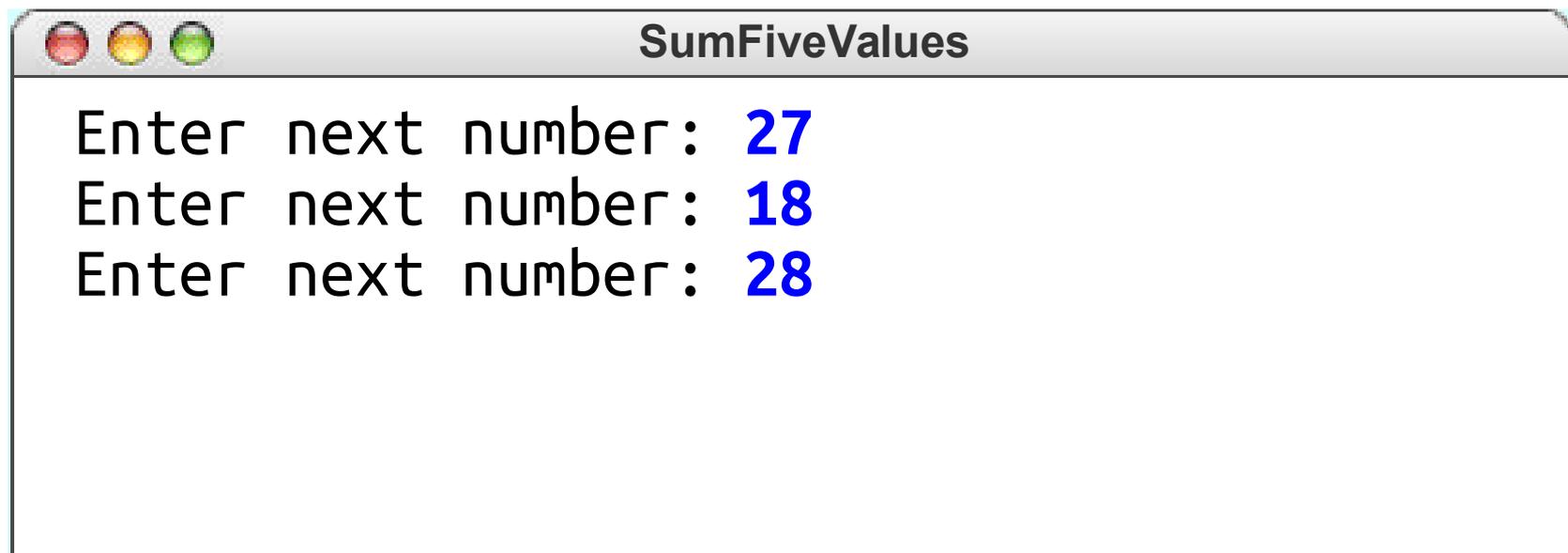
```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 45



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

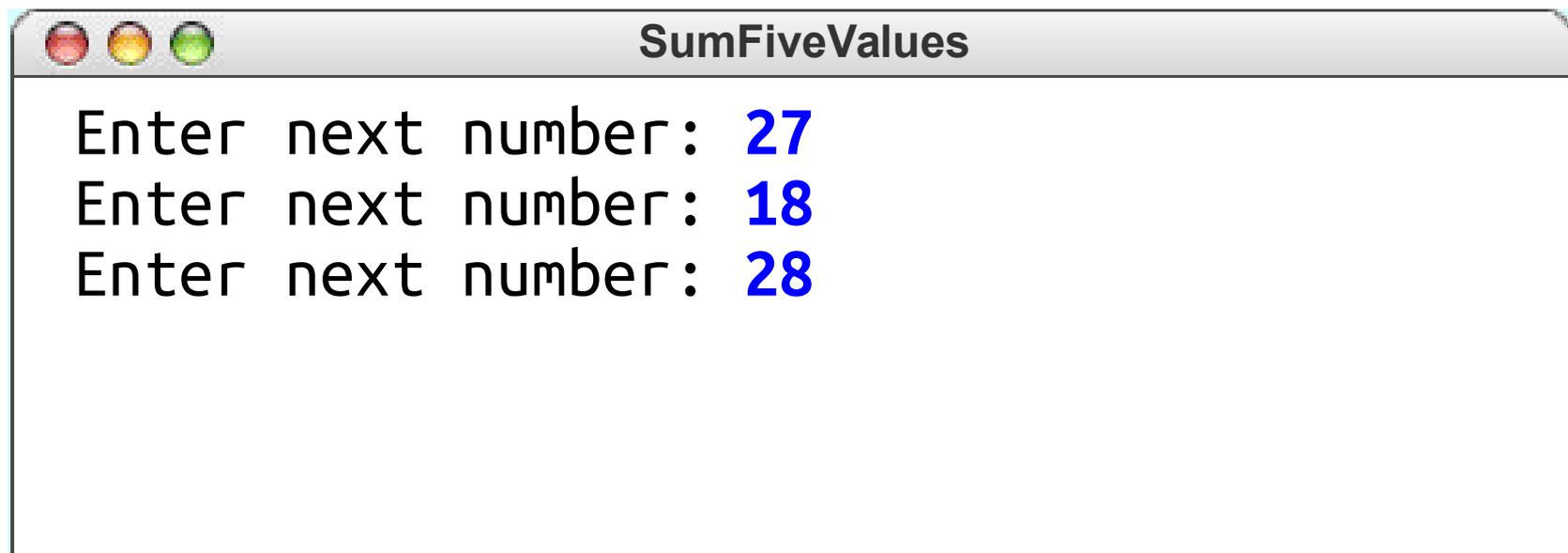
total 45



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 45

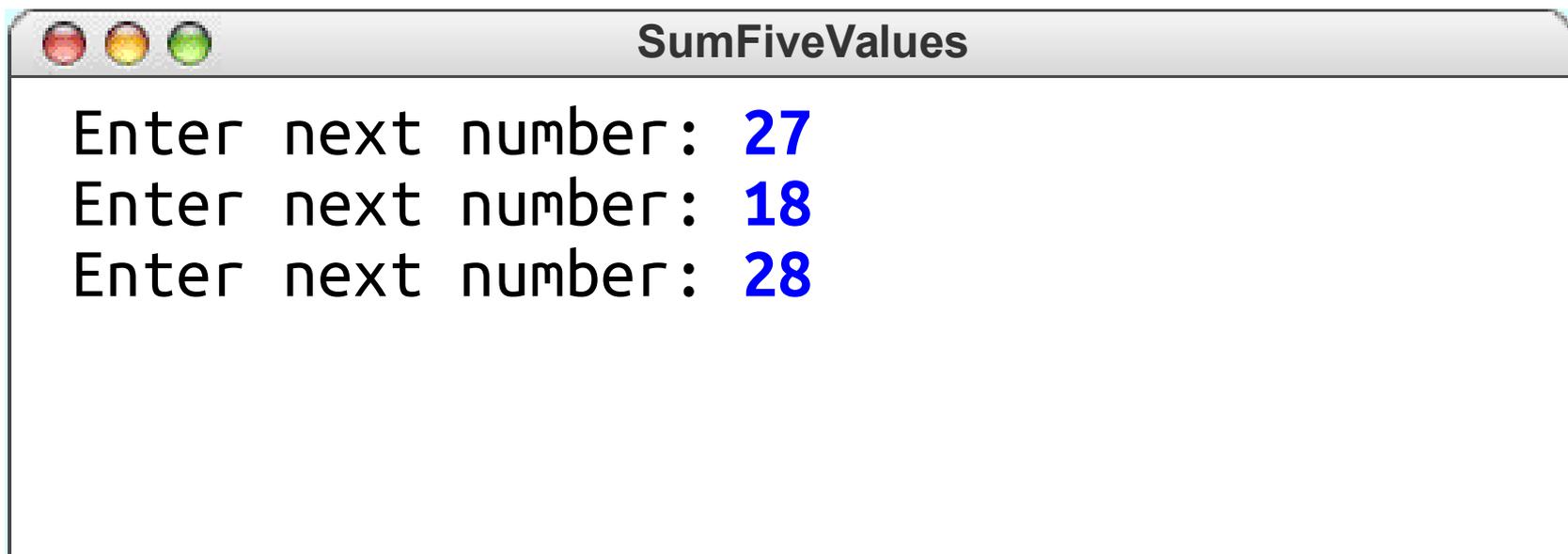
nextValue 28



```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 45

nextValue 28

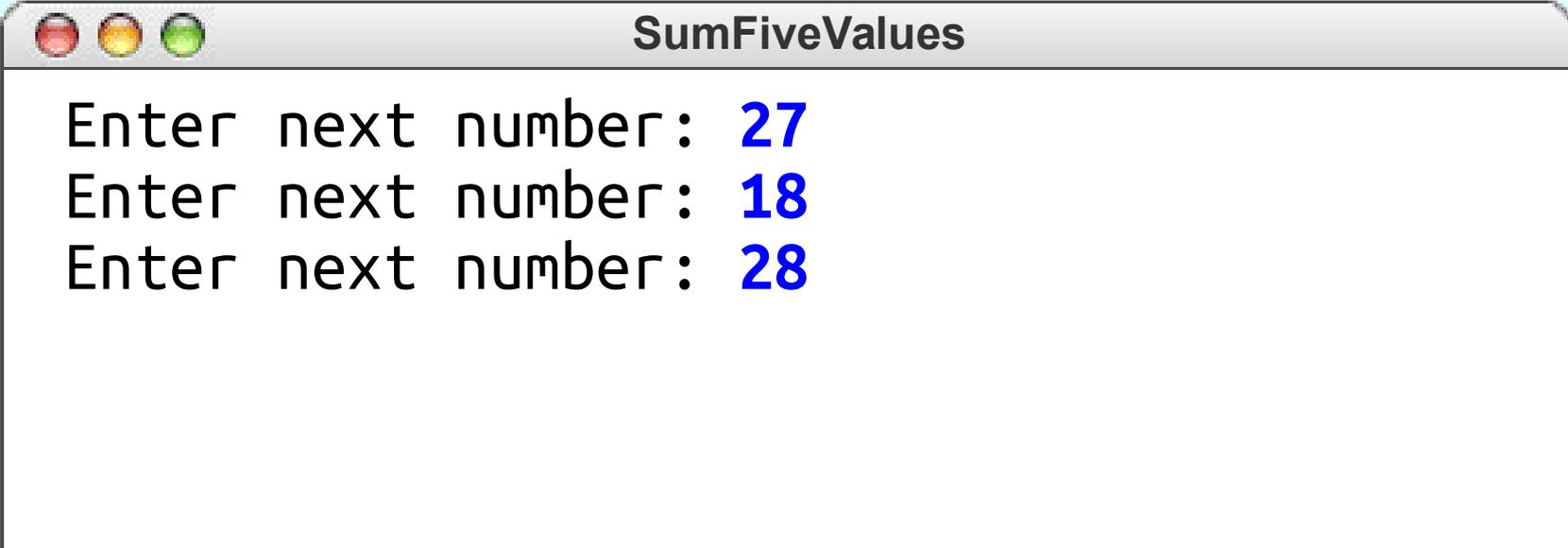


```
SumFiveValues  
Enter next number: 27  
Enter next number: 18  
Enter next number: 28
```

```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total 73

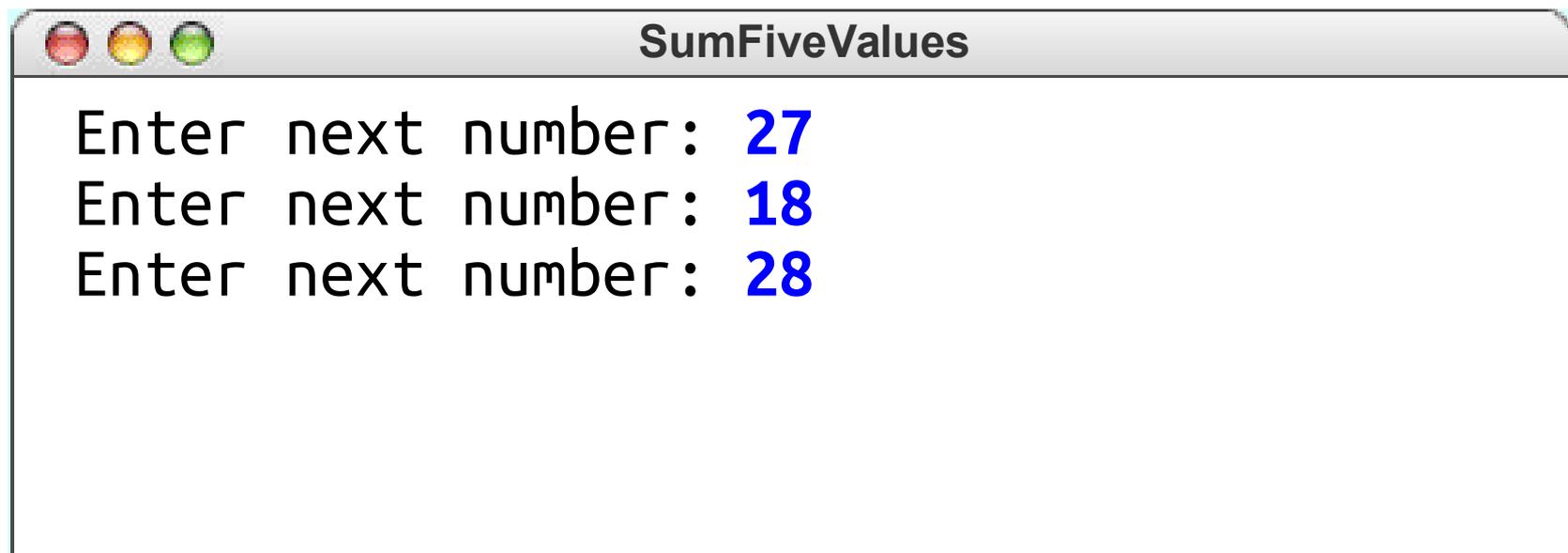
nextValue 28



```
SumFiveValues  
Enter next number: 27  
Enter next number: 18  
Enter next number: 28
```

```
public void run() {  
    int total = 0;  
    for (int i = 0; i < 5; i++) {  
        int nextValue = readInt("Enter next number: ");  
  
        /* Add the next value to the total. */  
        total = total + nextValue;  
    }  
  
    println("The total is " + total);  
}
```

total **73**



Aggregating Information

- It's extremely common to aggregate information across multiple iterations of a loop.
- General pattern:
 - If information needs to persist across loop iterations, store it in a variable defined outside the loop.
 - If information only needs to survive for a single iteration of the loop, define it inside the loop.

Scope

- Each variable has a ***scope*** where it can be accessed and how long it lives.

```
for (int i = 0; i < 5; i++) {  
    int y = i * 4;  
    println(y);  
}
```

```
println(i); // Error!
```

```
println(y); // Error!
```

Scope

- Each variable has a ***scope*** where it can be accessed and how long it lives.
- Variables declared *outside* a loop persist across all loop iterations.
- Variables declared *inside* a loop persist only for a single iteration.
- The loop counter in a for loop persists as long as the loop runs, then disappears.

A Useful Shorthand

- Commonly, programs contain code like this:

```
x = x + 1;
```

```
z = z / 14;
```

```
y = y * 137;
```

```
w = w - 3;
```

A Useful Shorthand

- Commonly, programs contain code like this:

```
x = x + 1;
```

```
y = y * 137;
```

```
z = z / 14;
```

```
w = w - 3;
```

- The statement

variable = variable op value;

can be rewritten as

variable op= value;

A Useful Shorthand

- Commonly, programs contain code like this:

```
x += 1;
```

```
z /= 14;
```

```
y *= 137;
```

```
w -= 3;
```

- The statement

variable = variable op value;

can be rewritten as

variable op= value;

Another Useful Shorthand

- In the special case of writing

variable = ***variable*** + 1;

we can instead write

variable++;

- In the special case of writing

variable = ***variable*** - 1;

we can instead write

variable--;

++: Seem Familiar?

- Hmm... haven't we seen this ++ thing somewhere before?
- How about in

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

- What does this mean?

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

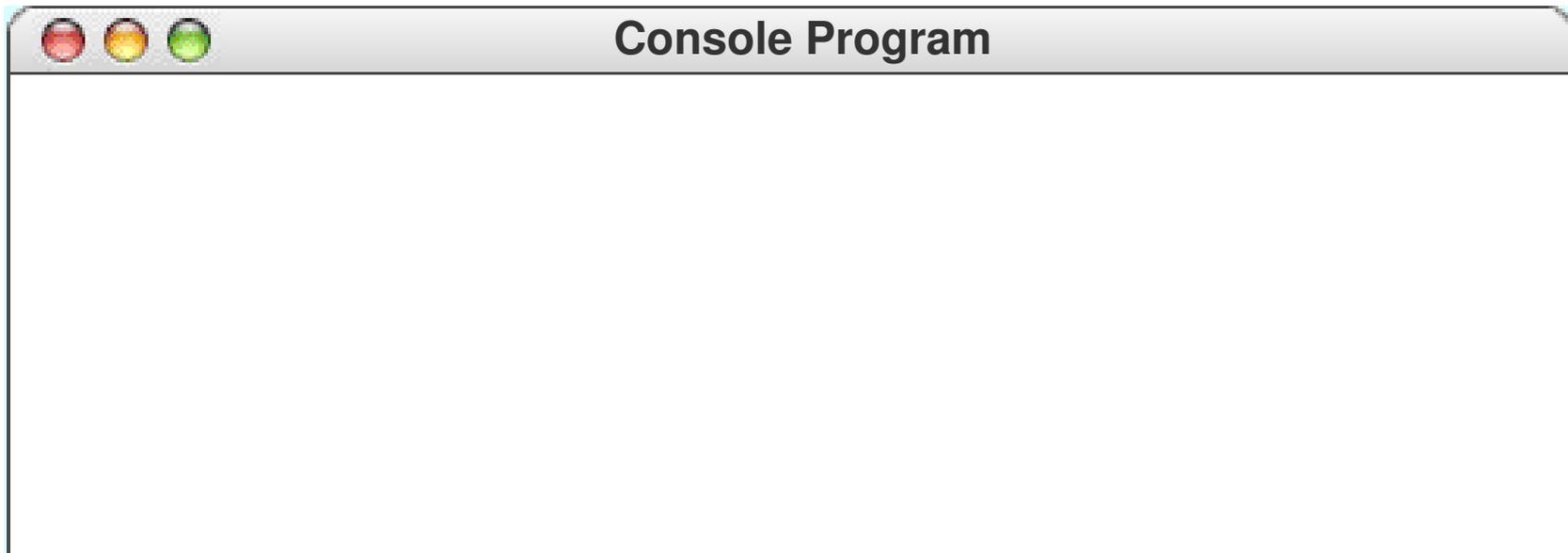
This is called the **step** or **increment** and is performed at the end of each loop iteration.

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

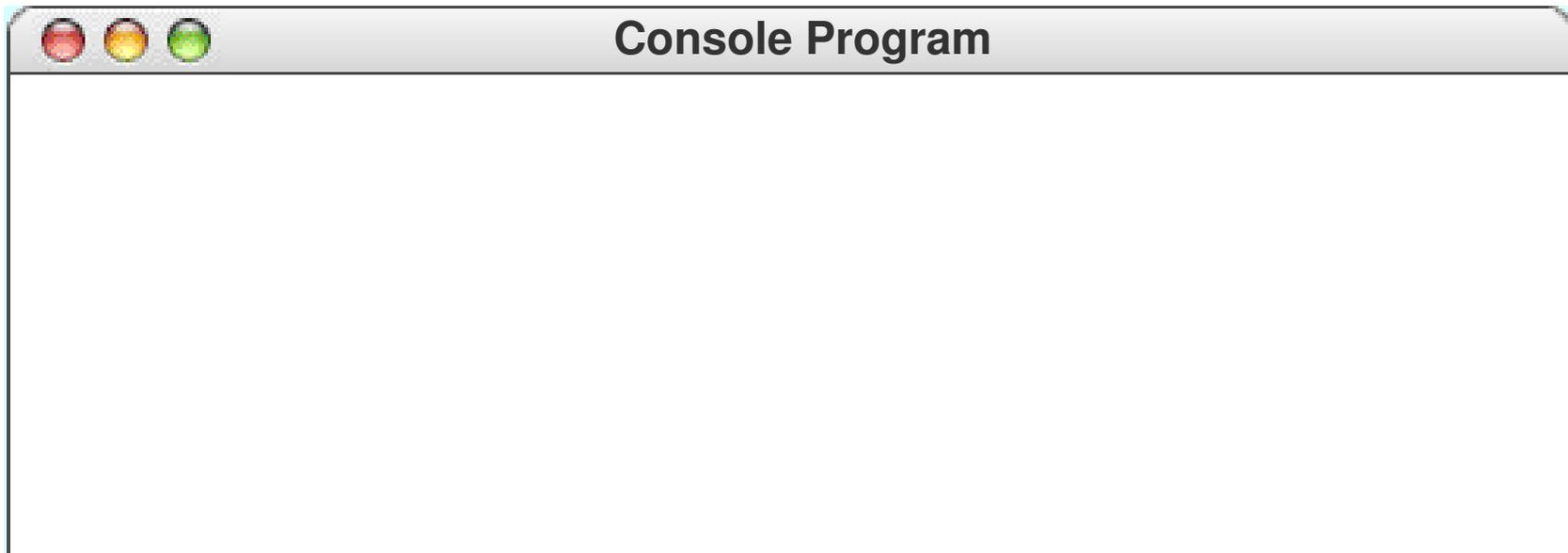
This is called the **loop condition** or **termination condition**. The loop will check whether this statement is true before each iteration of the loop.

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

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



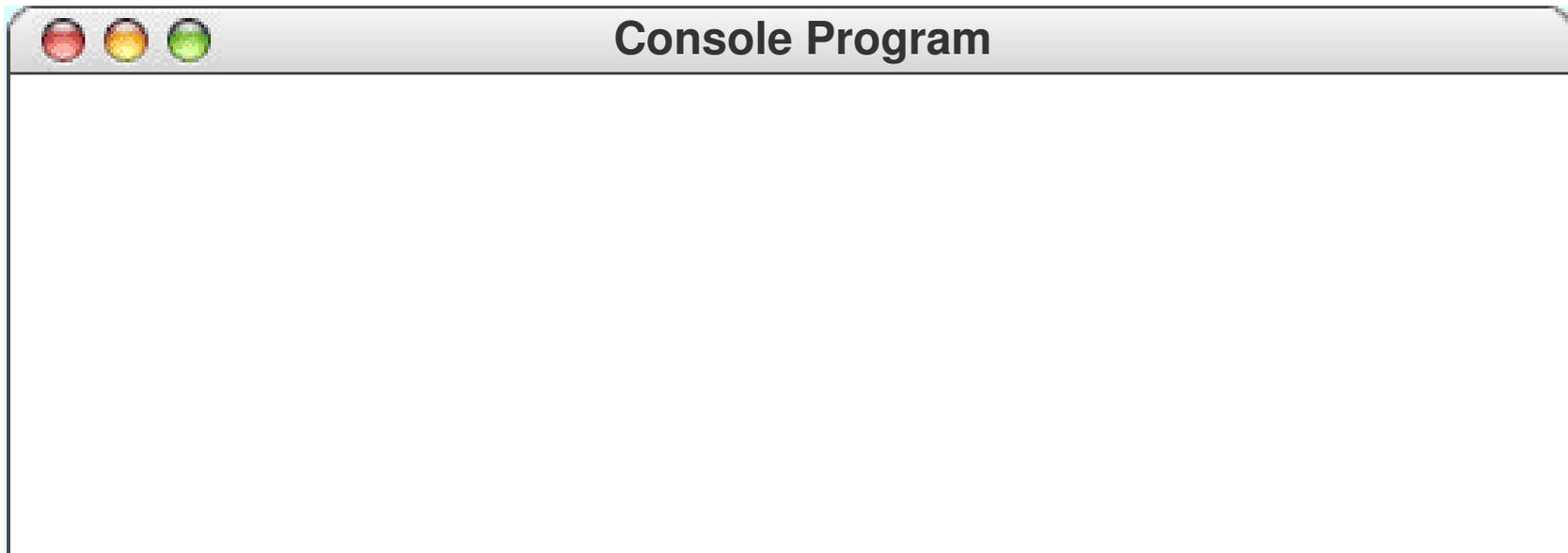
```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Ohhh");
```



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

int i

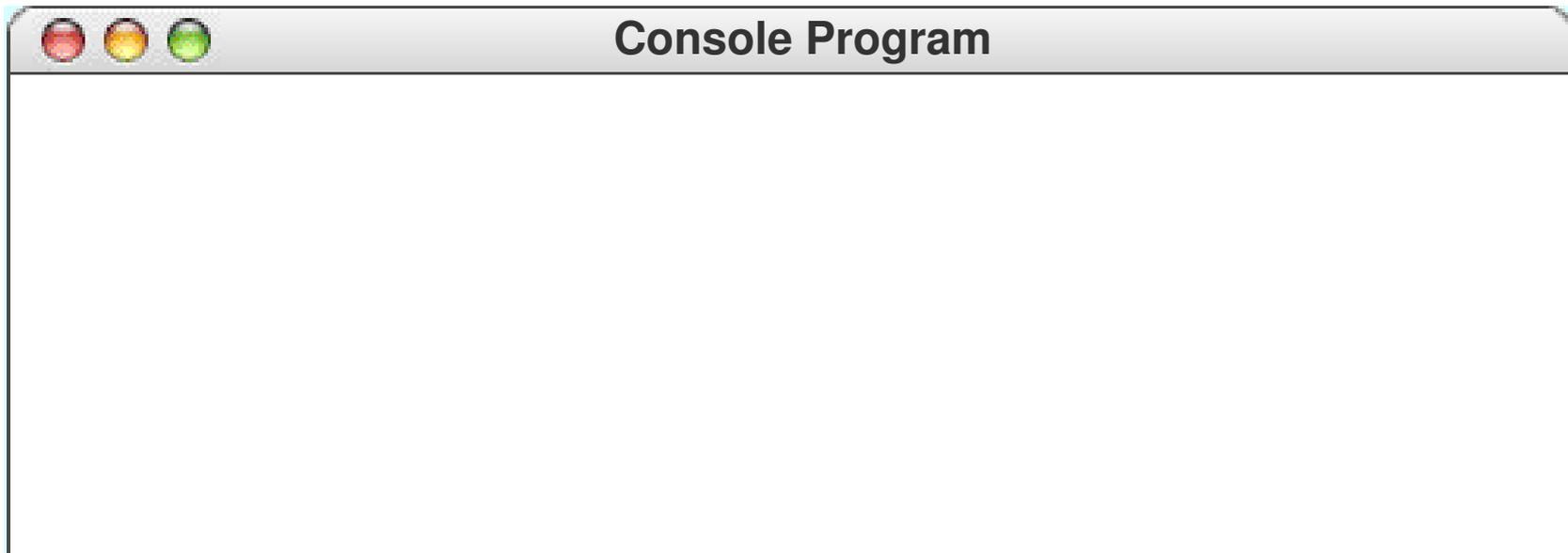
0



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

int i

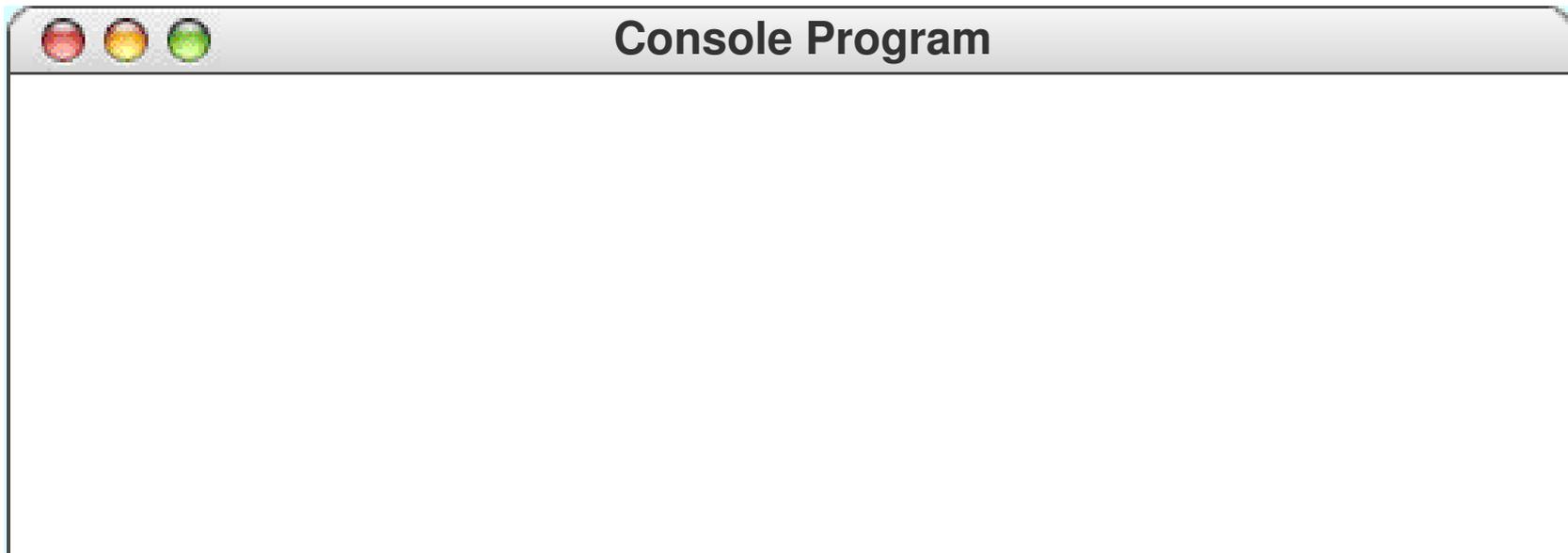
0



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

int i

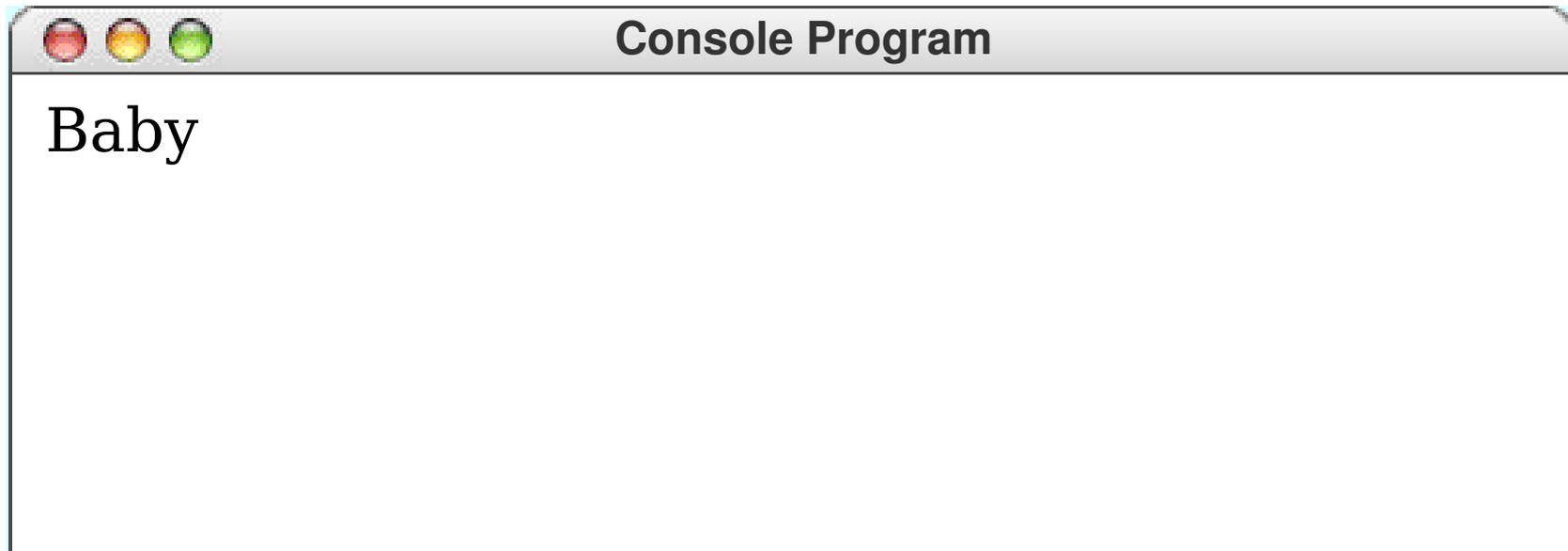
0



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

int i

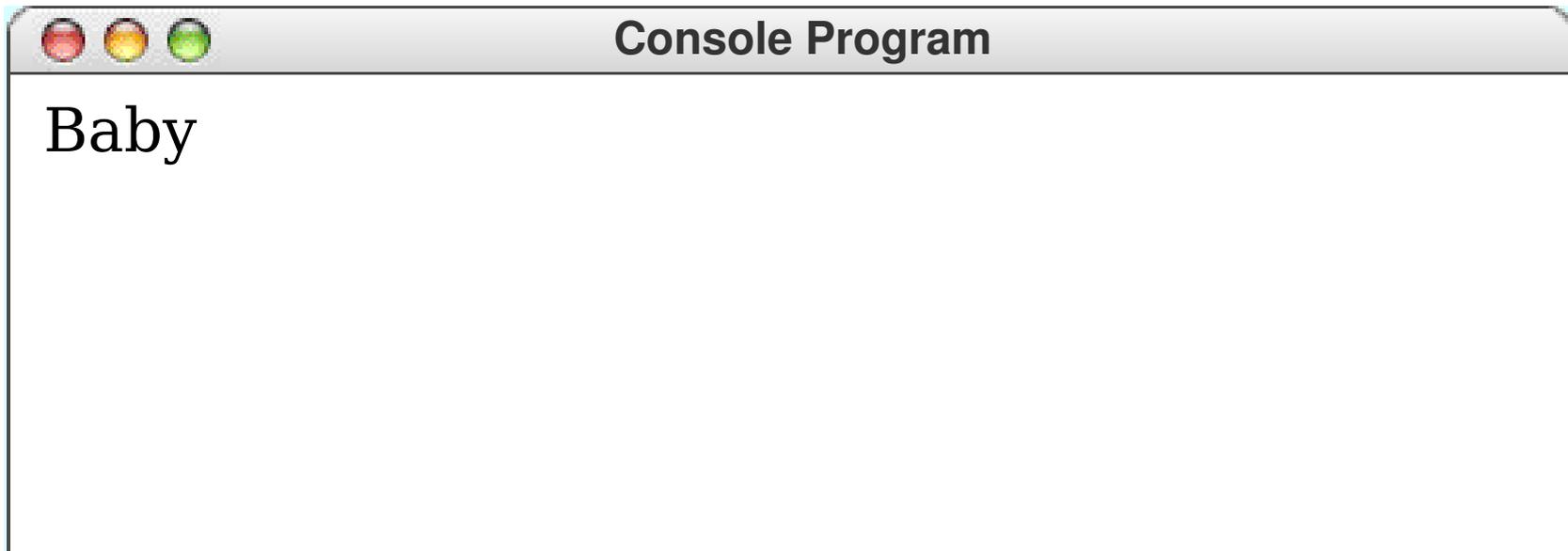
0



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

int i

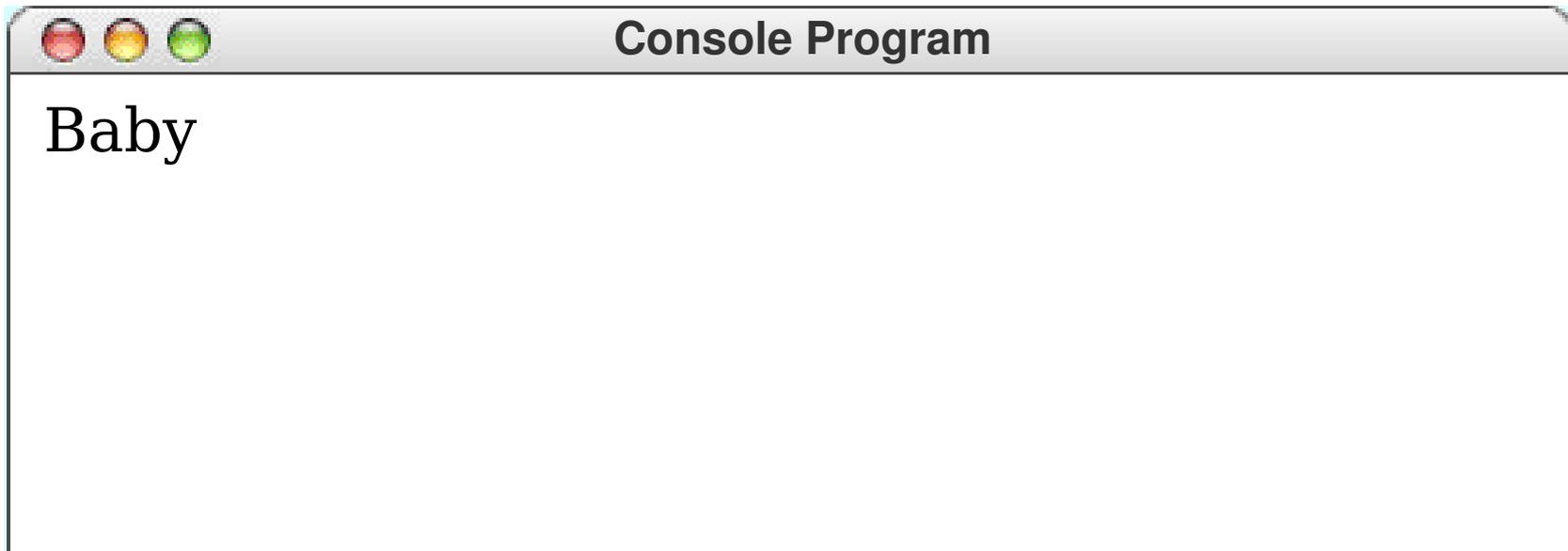
0



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

int i

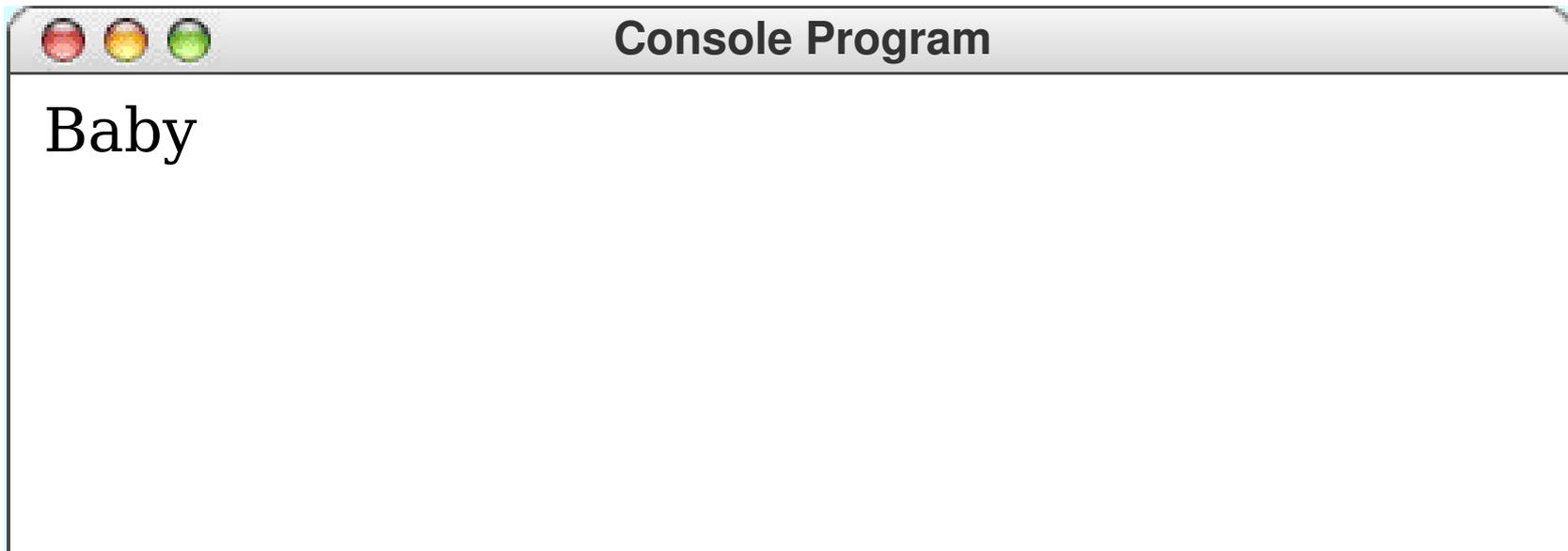
1



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

int i

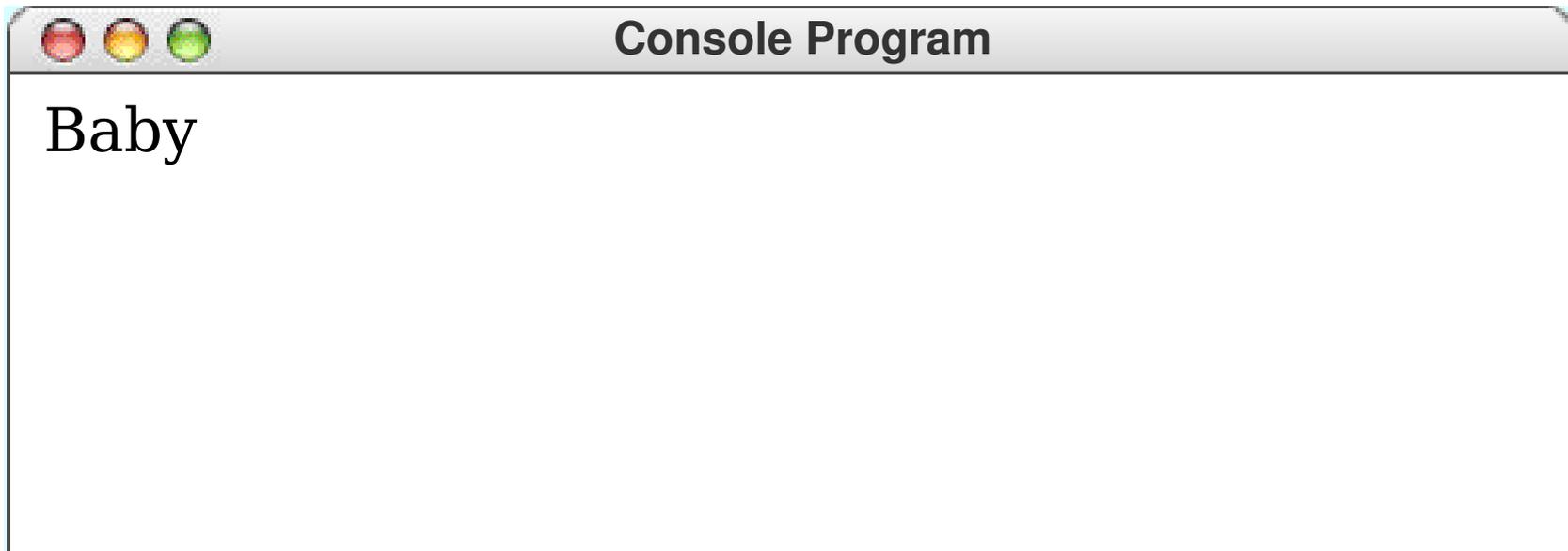
1



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

int i

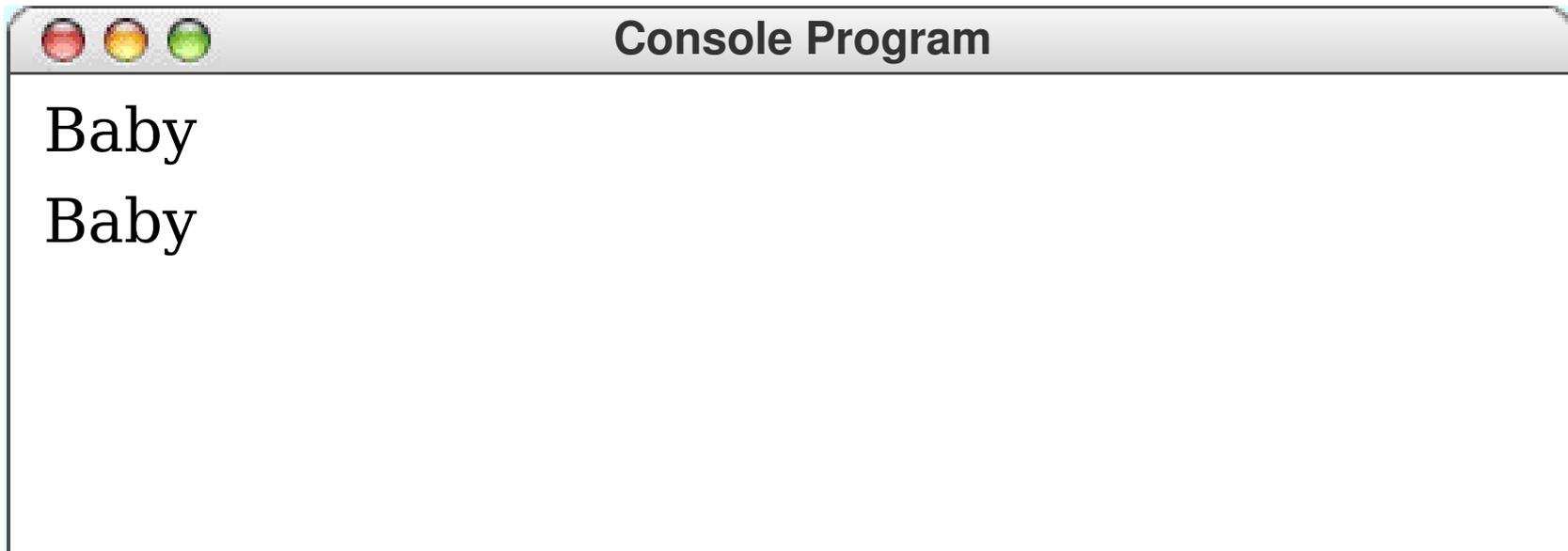
1



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

int i

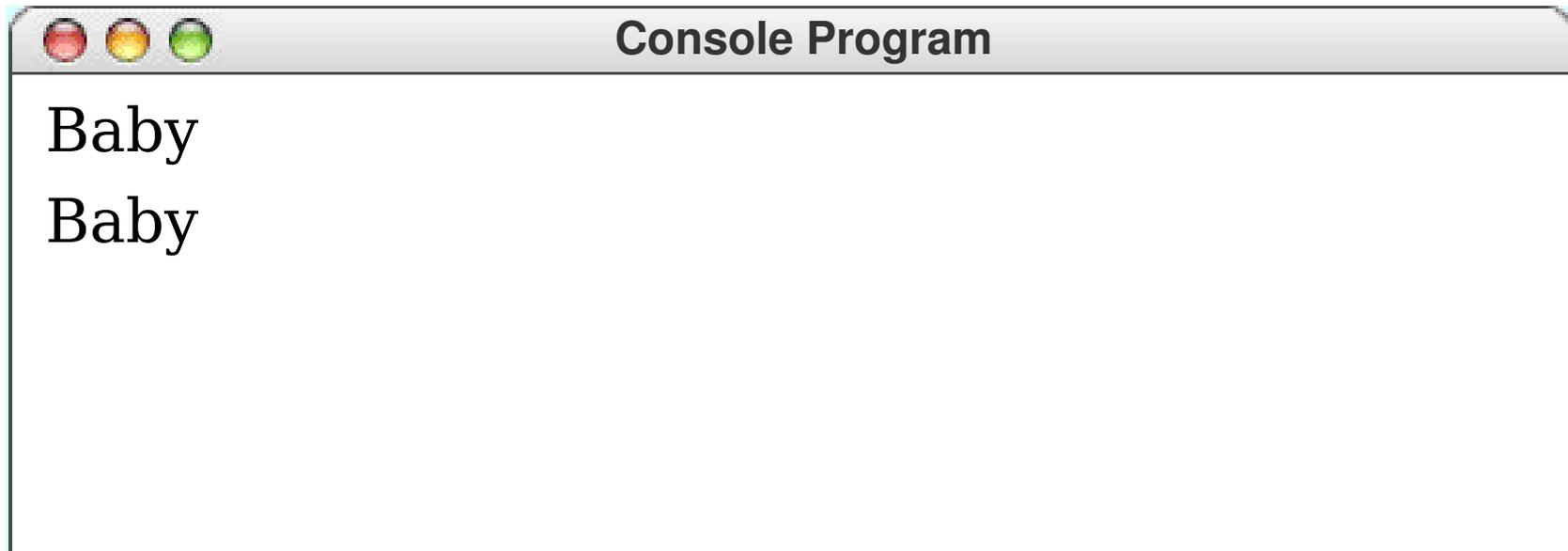
1



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

int i

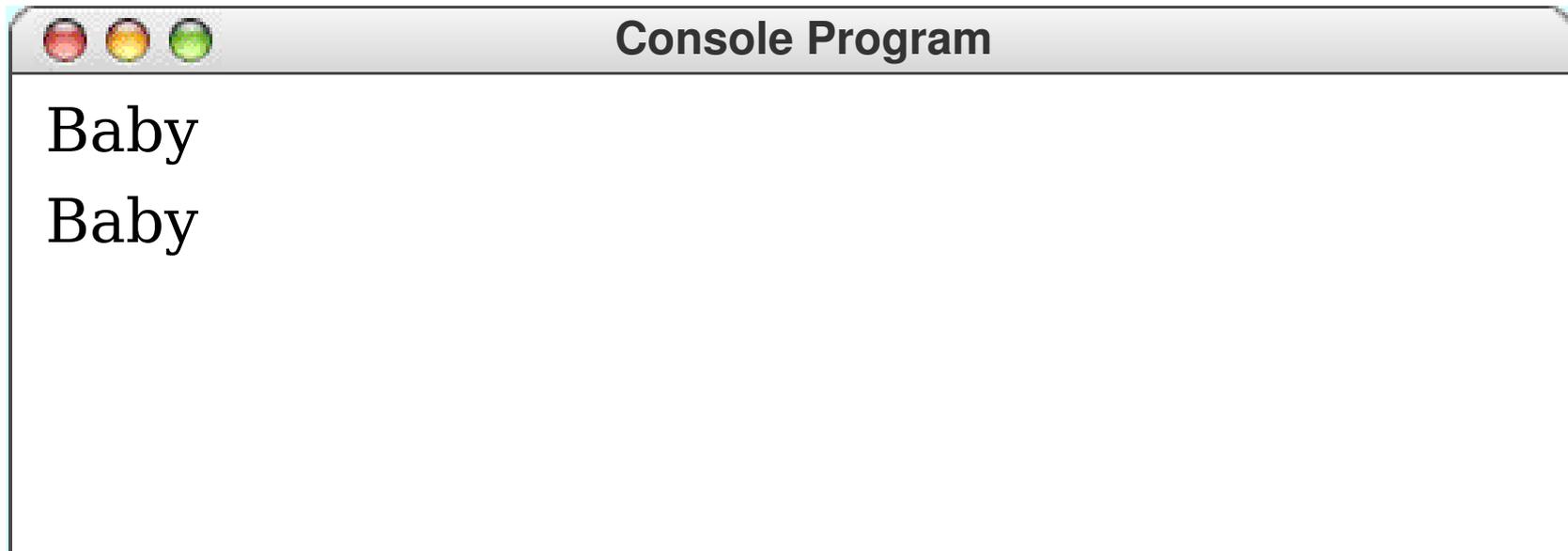
1



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

int i

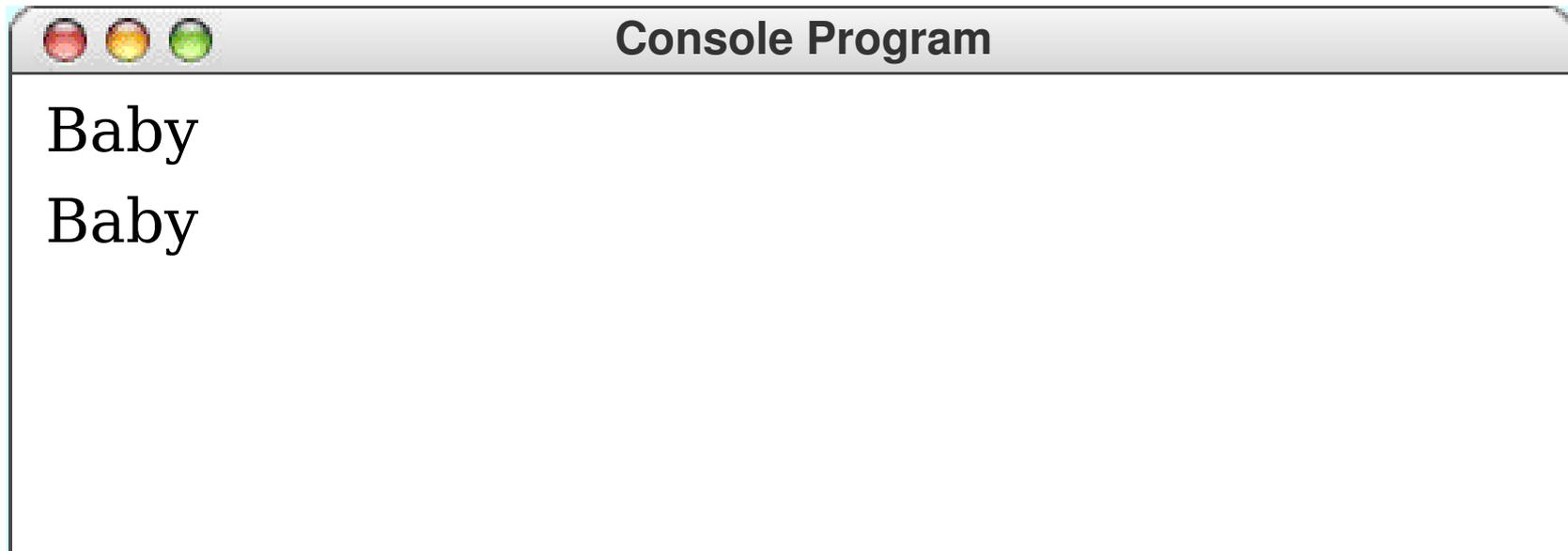
2



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

int i

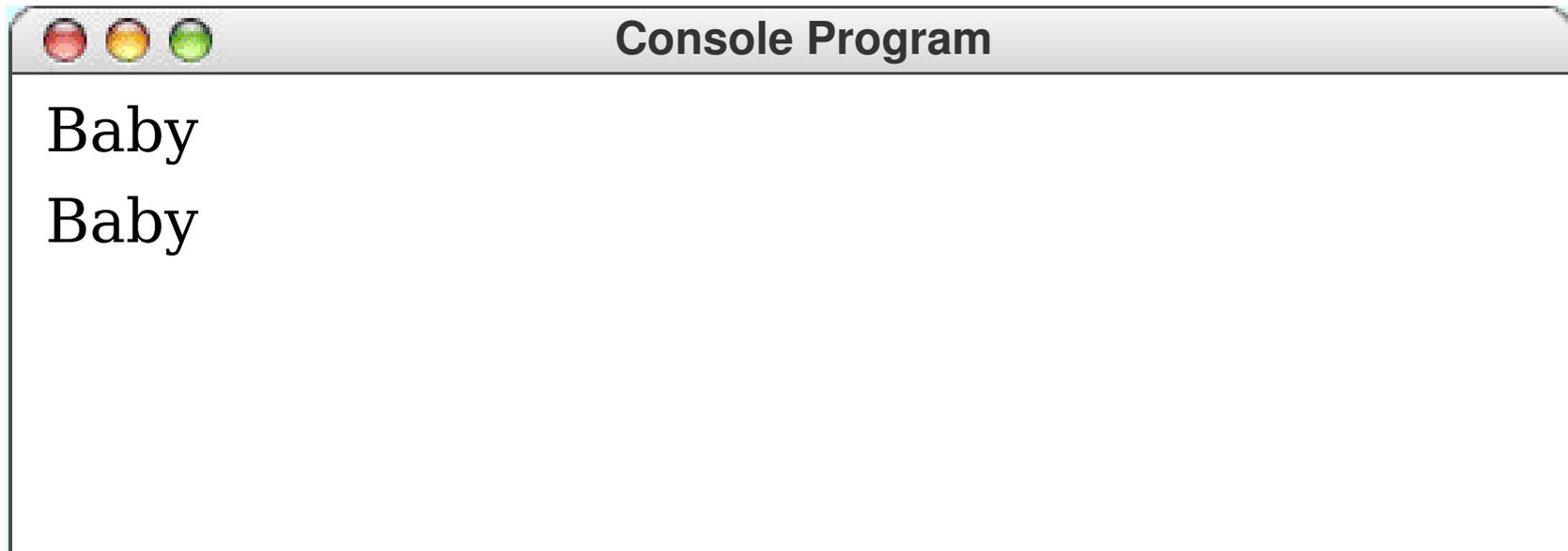
2



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

int i

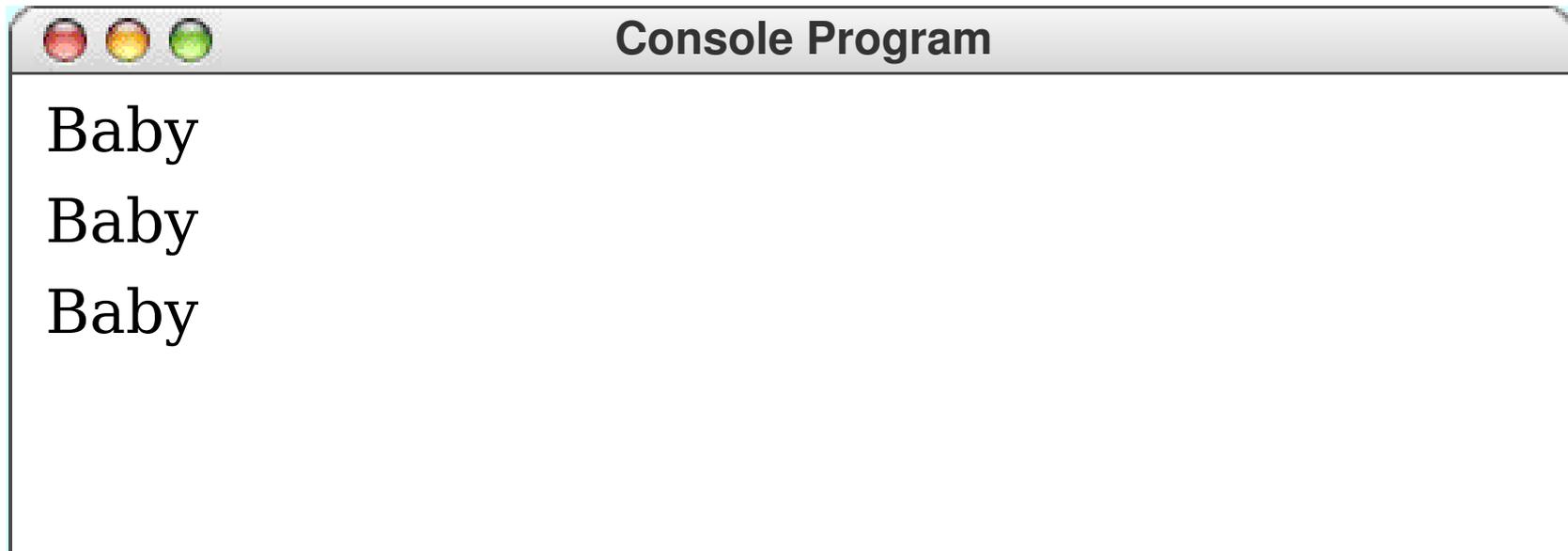
2



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

int i

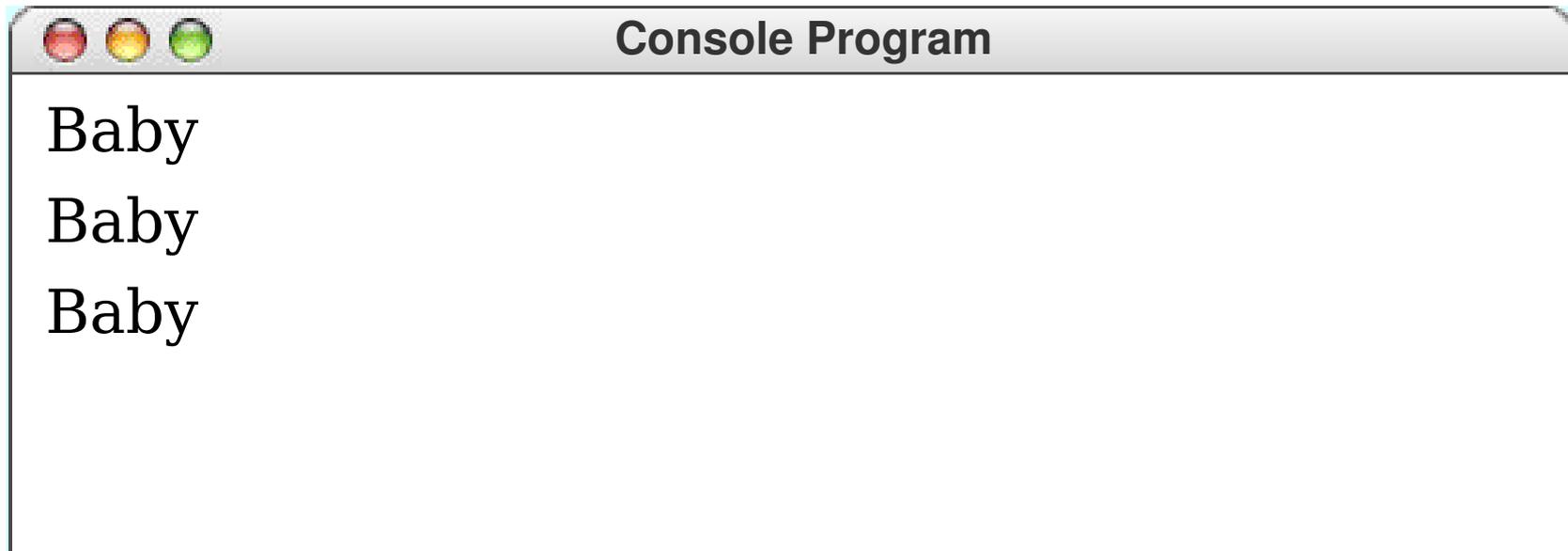
2



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

int i

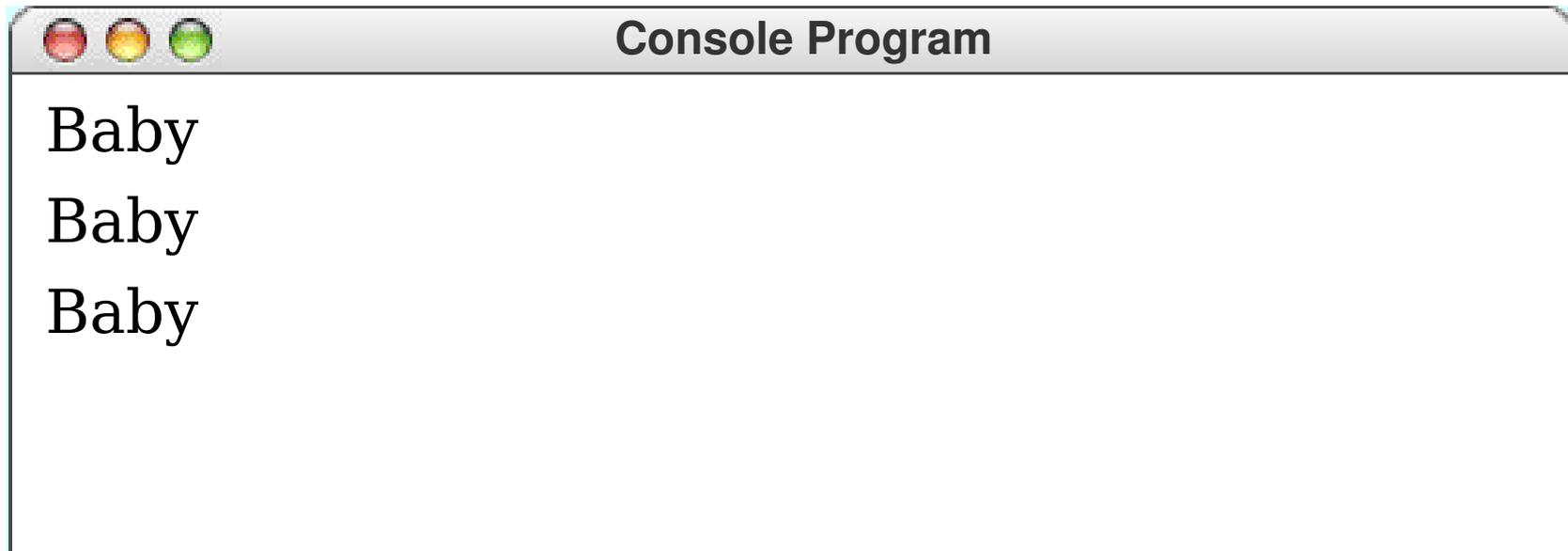
2



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

int i

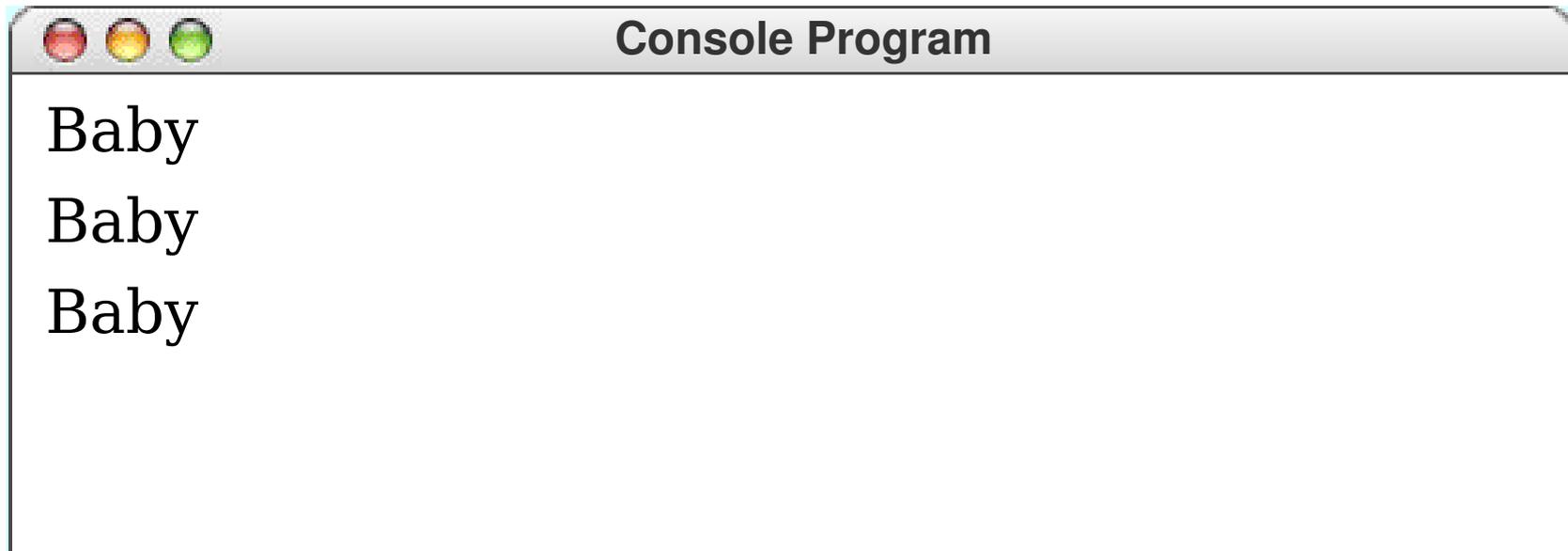
3



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

int i

3

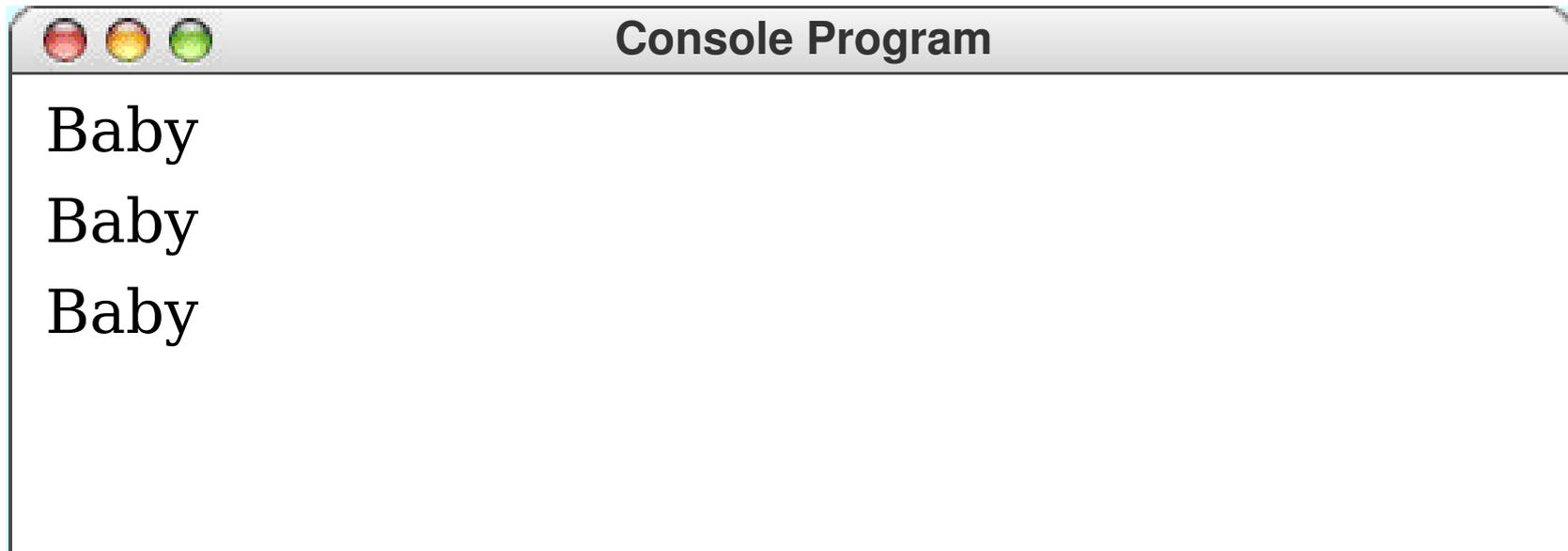


```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}
```

```
println("Ohhh");
```

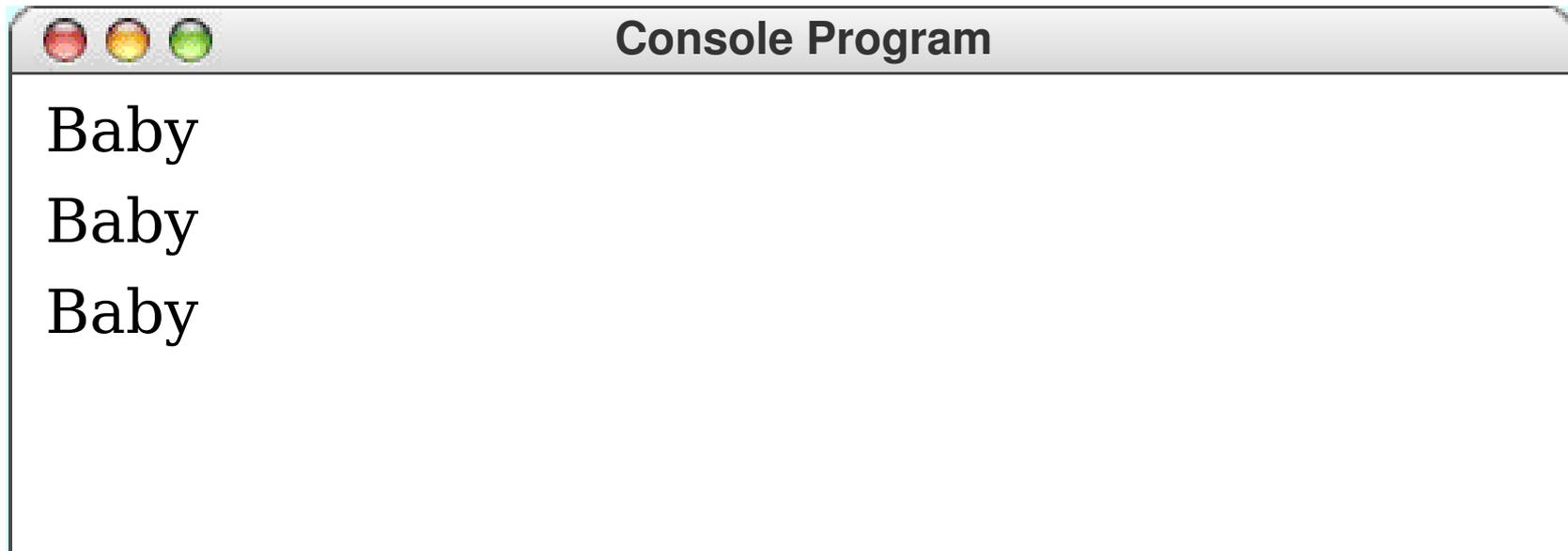
int i

3

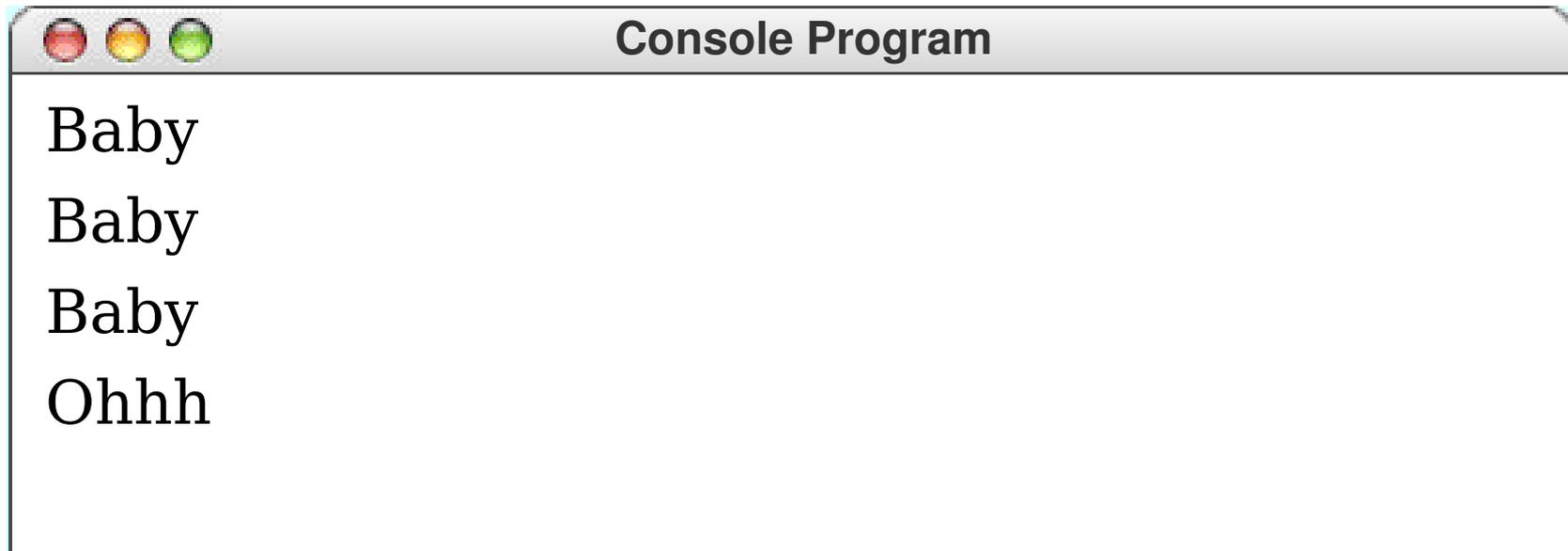


```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}
```

```
println("Ohhh");
```



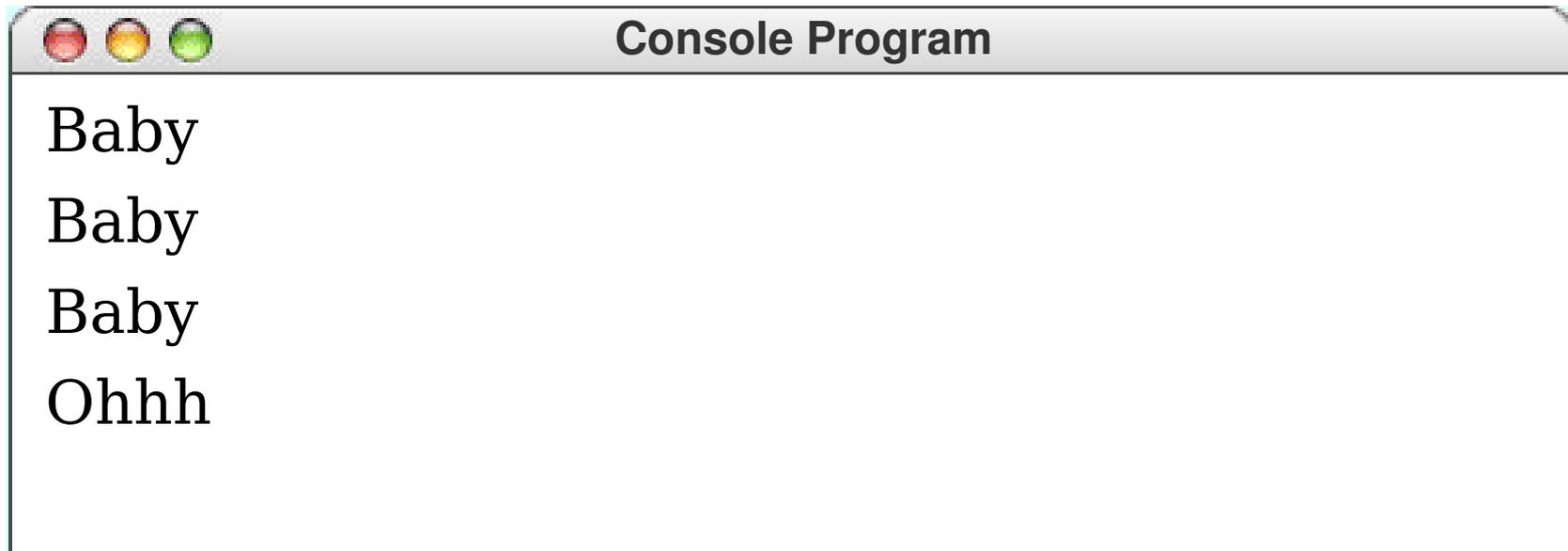
```
for (int i = 0; i < 3; i++) {  
    println("Baby");  
}  
println("Ohhh");
```



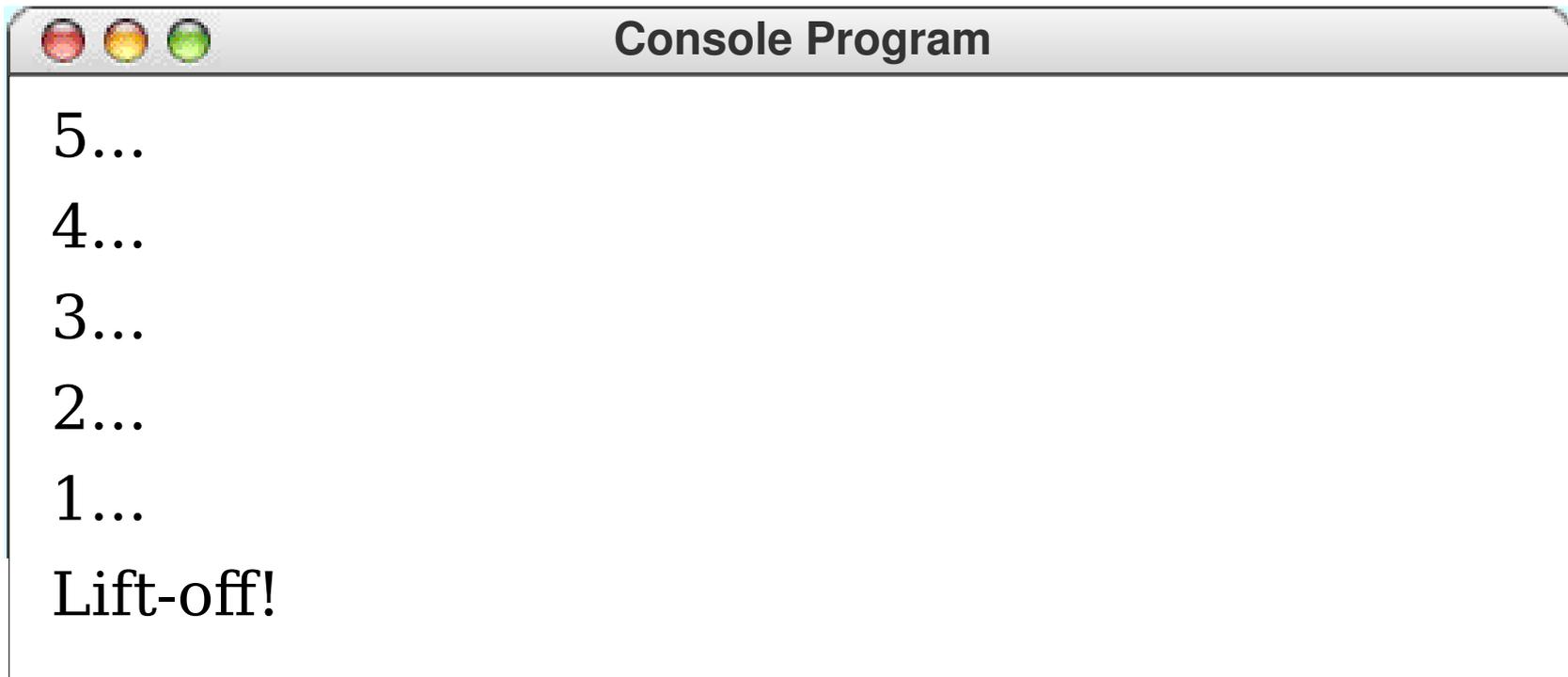
Console Program

```
Baby  
Baby  
Baby  
Ohhh
```

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



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



Console Program

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

Control Statements

if
for
while

Control Statements

if
for
while

The while Loop

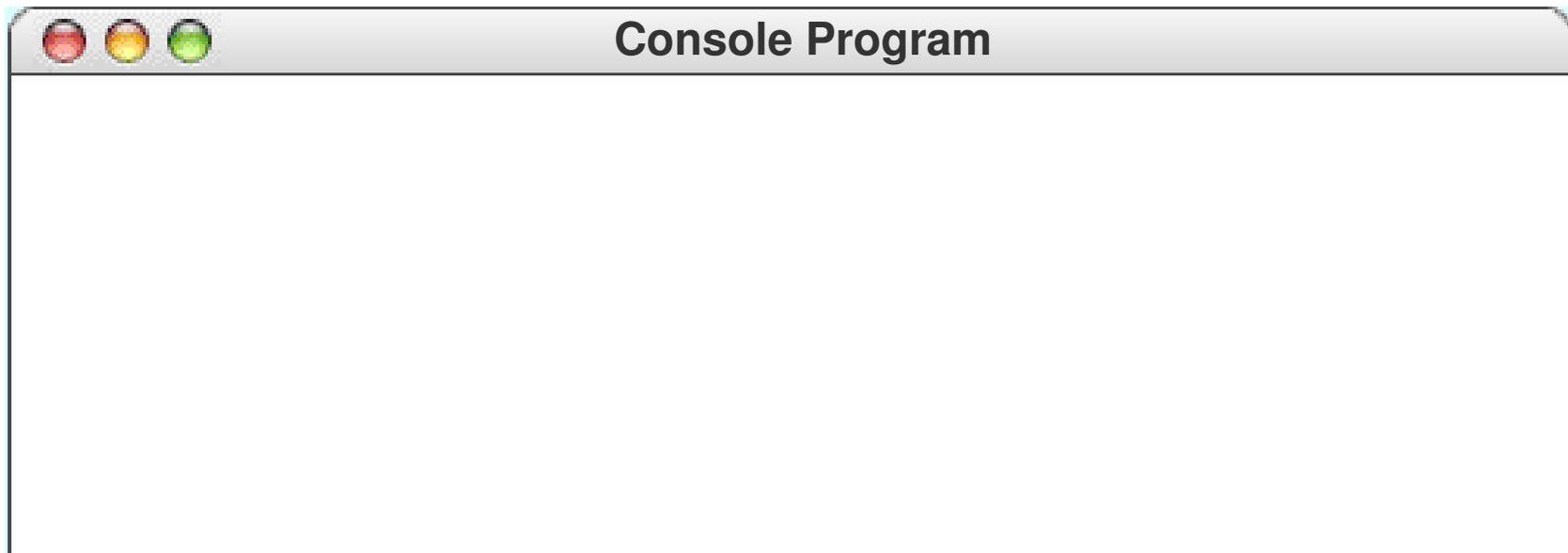
```
while (condition) {  
    ... statements ...  
}
```

- This loop works as follows:
 - Check whether *condition* is true.
 - If so, execute *statements* in their entirety, then repeat this process.
 - If not, move on to whatever comes after the loop.

The while Loop

Example:

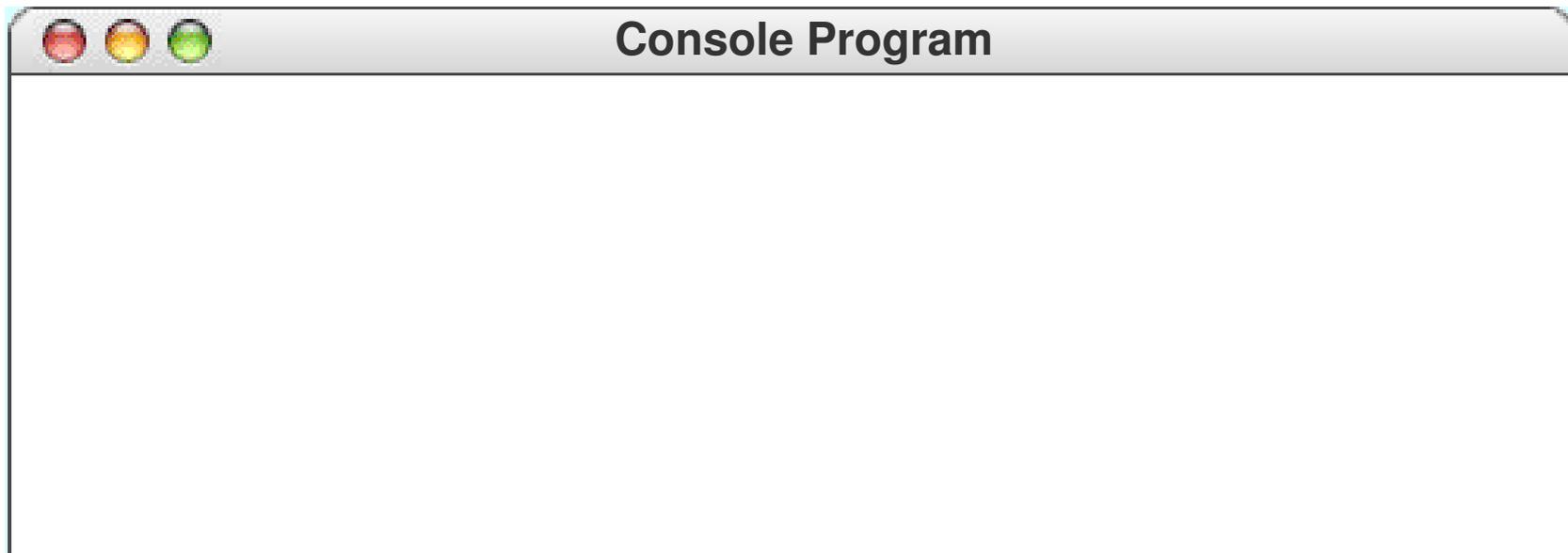
```
int x = 15;  
while (x > 1) {  
    x /= 2;  
    println(x);  
}
```



The while Loop

Example:

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

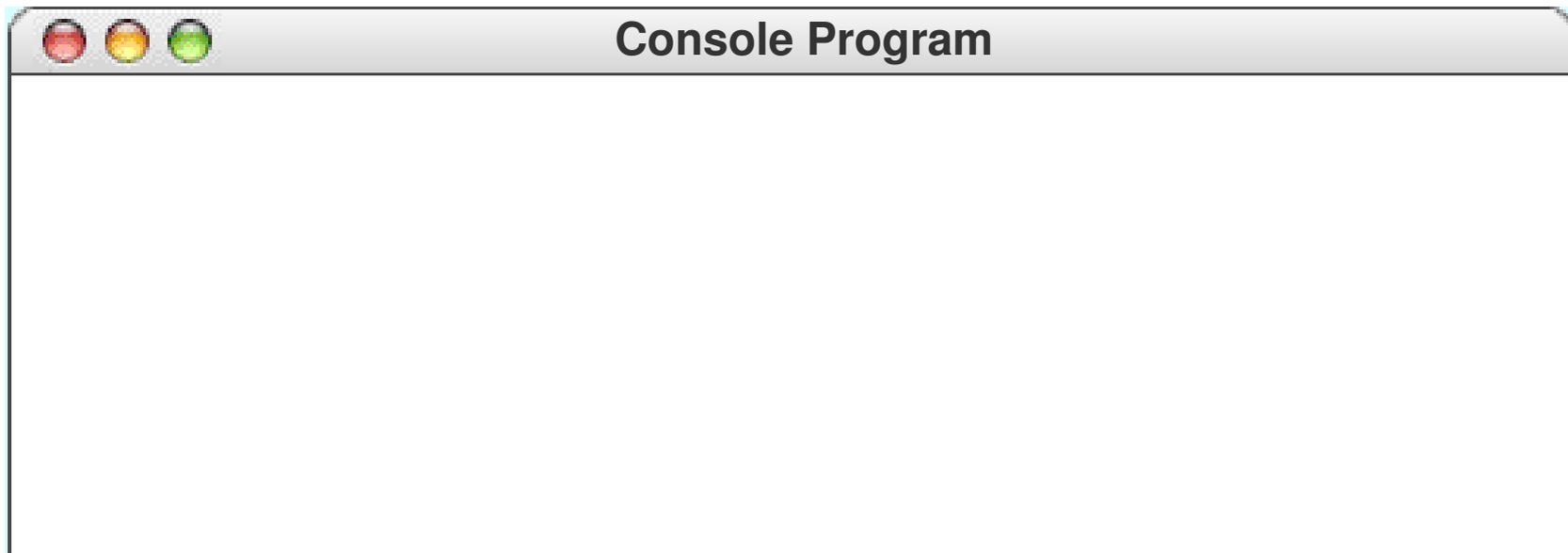


The while Loop

Example:

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

15 int x

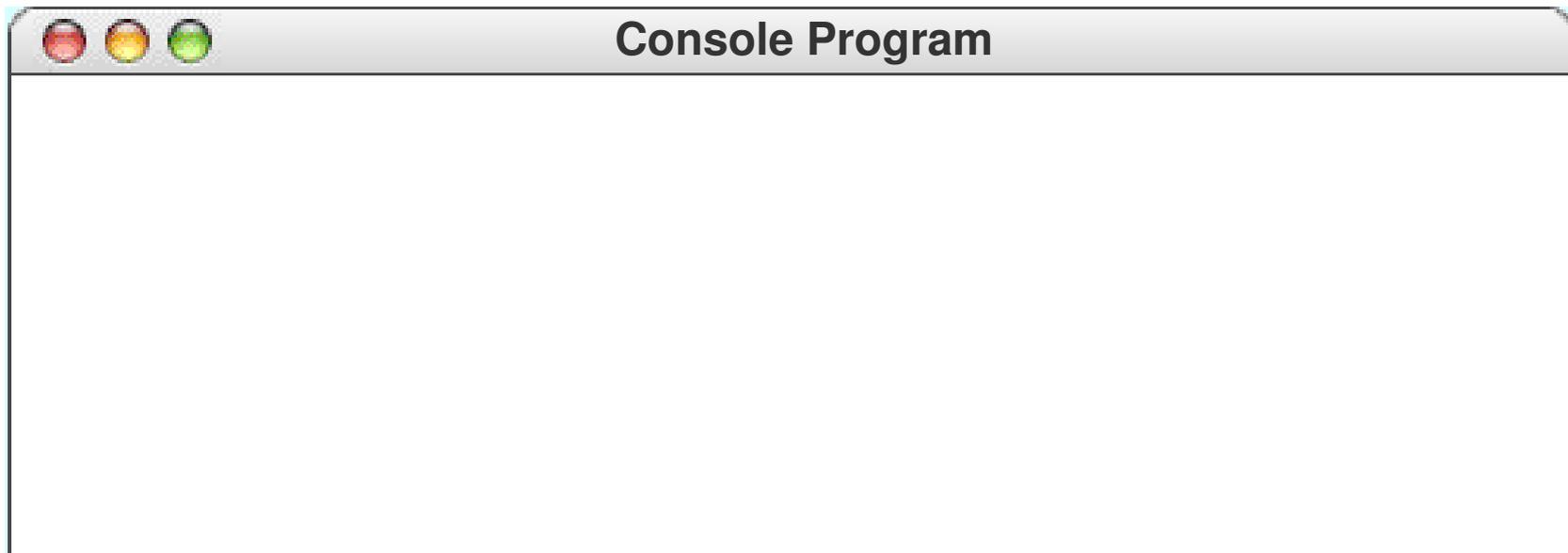


The while Loop

Example:

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

15 int x



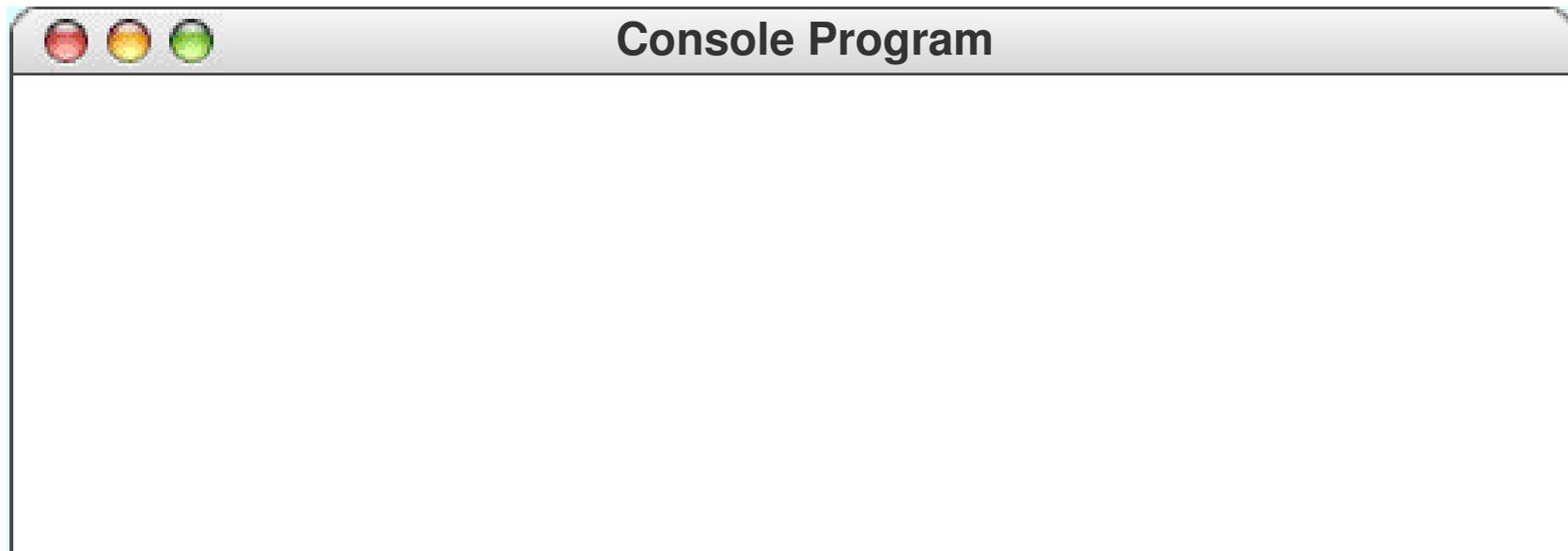
The while Loop

Example:

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

15

int x



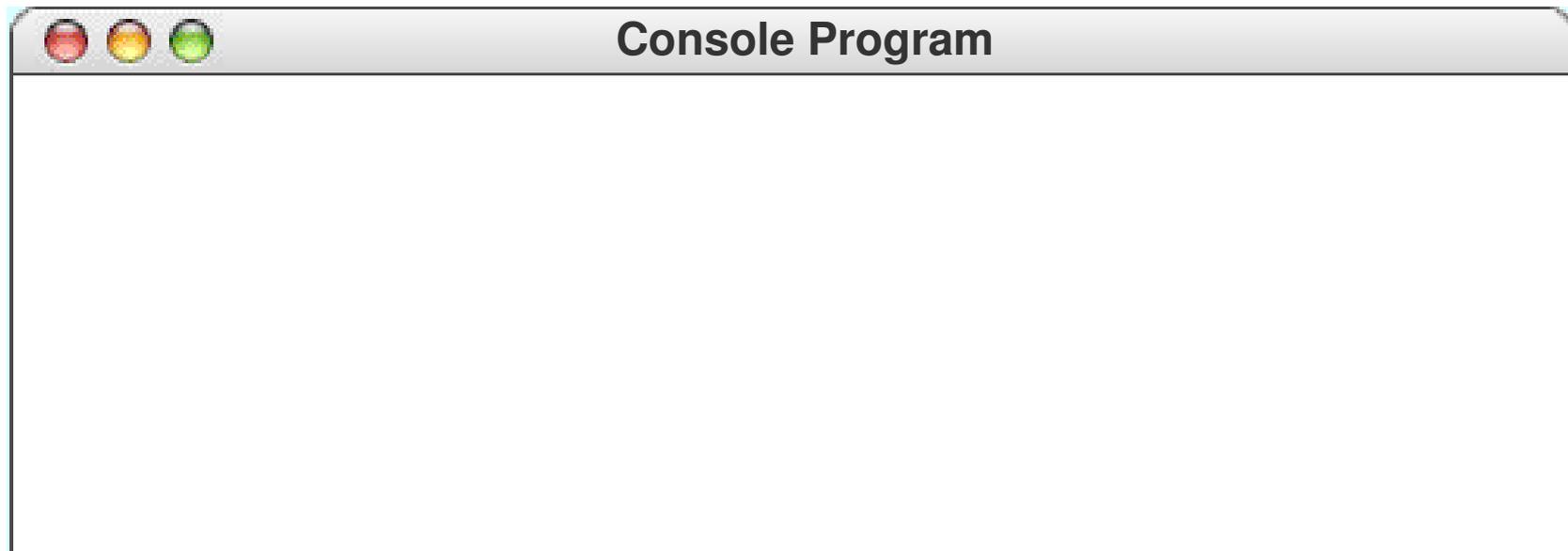
The while Loop

Example:

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

7

int x



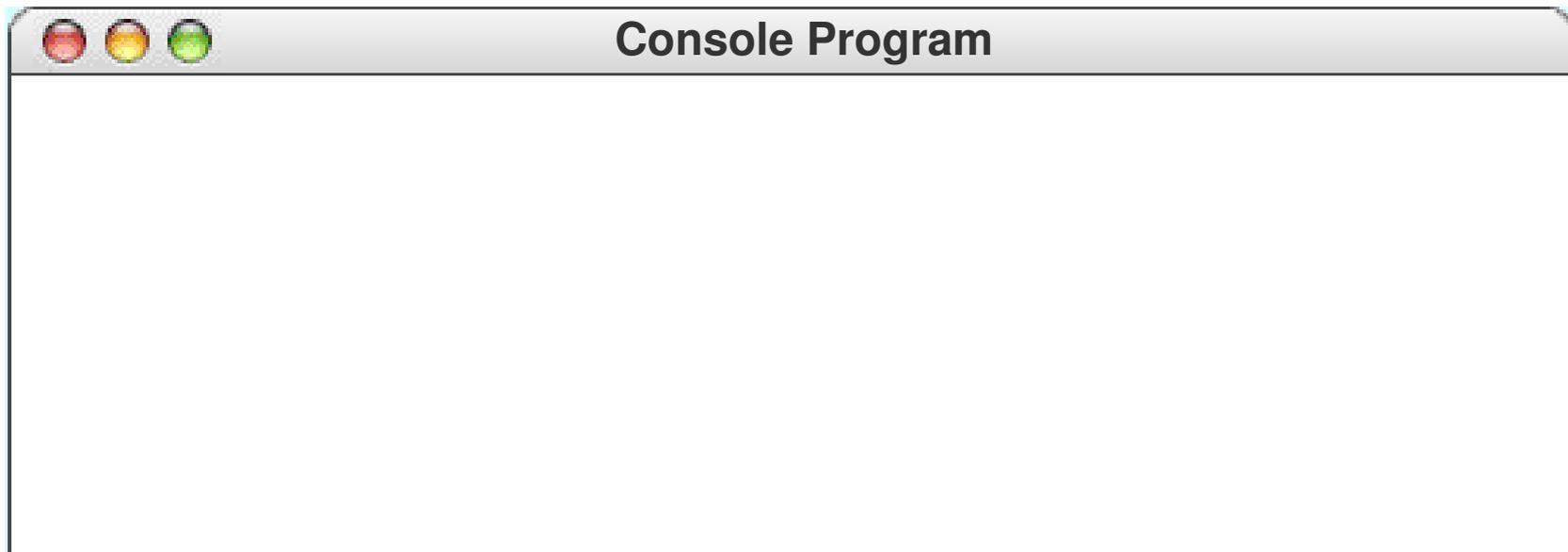
The while Loop

Example:

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

7

int x



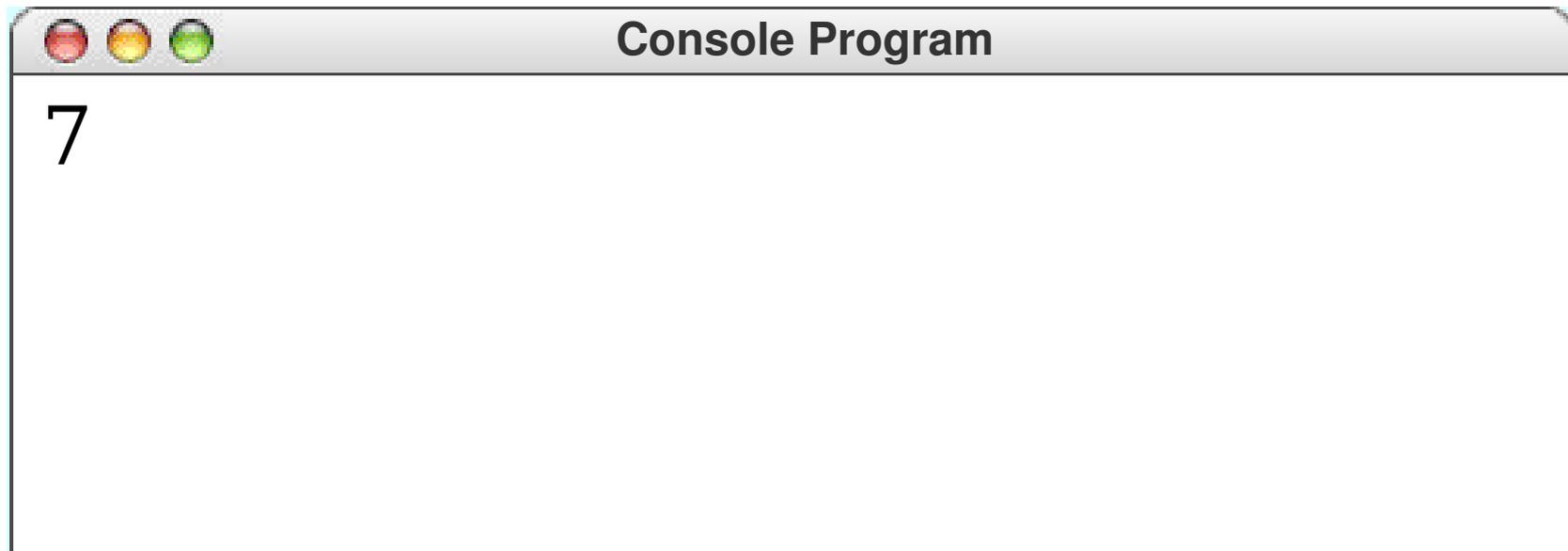
The while Loop

Example:

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

7

int x



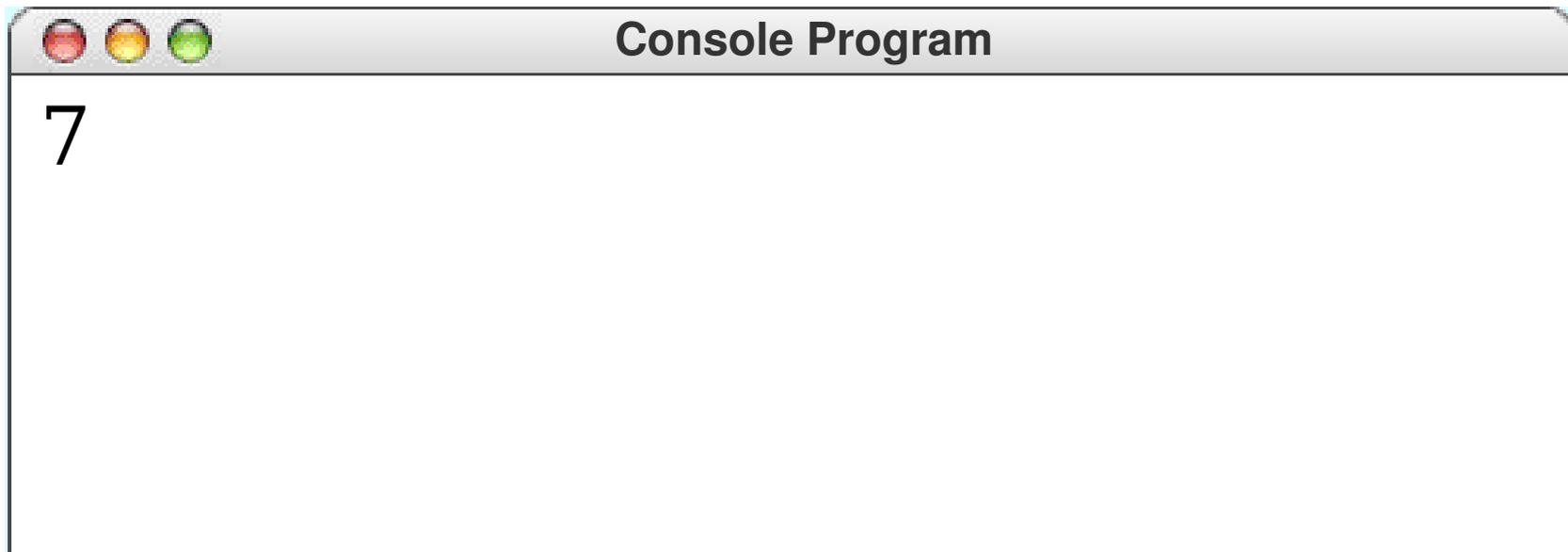
The while Loop

Example:

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

7

int x



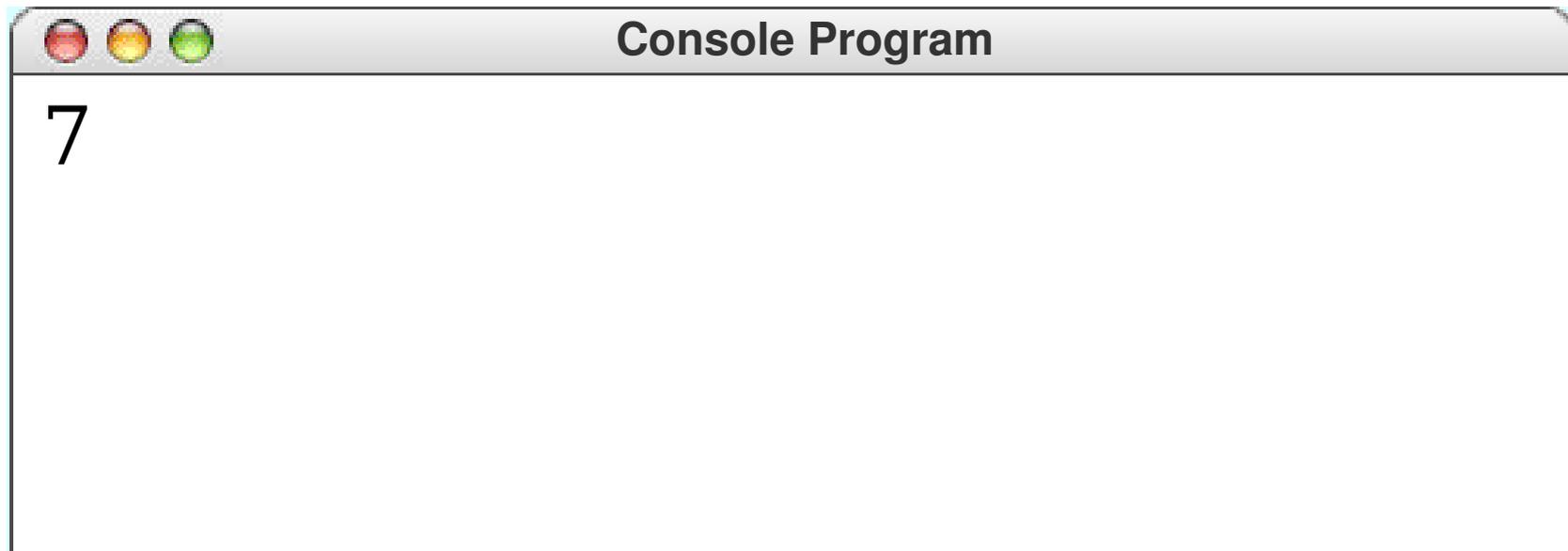
The while Loop

Example:

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

7

int x



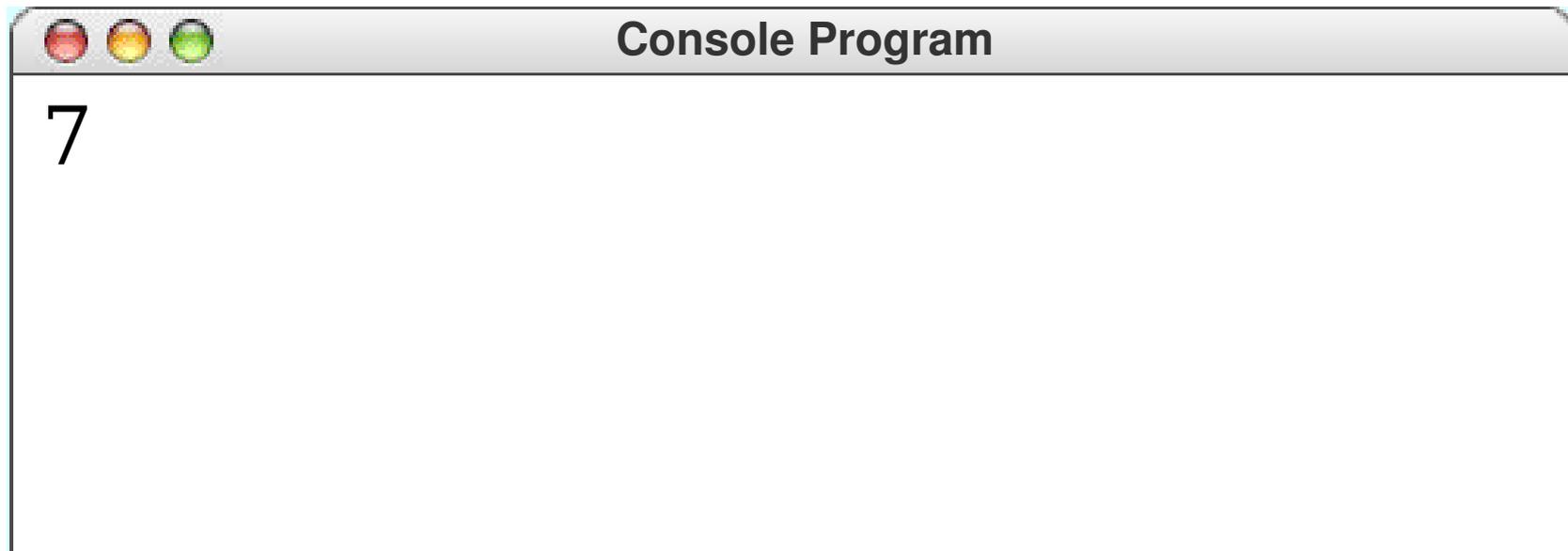
The while Loop

Example:

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

3

int x



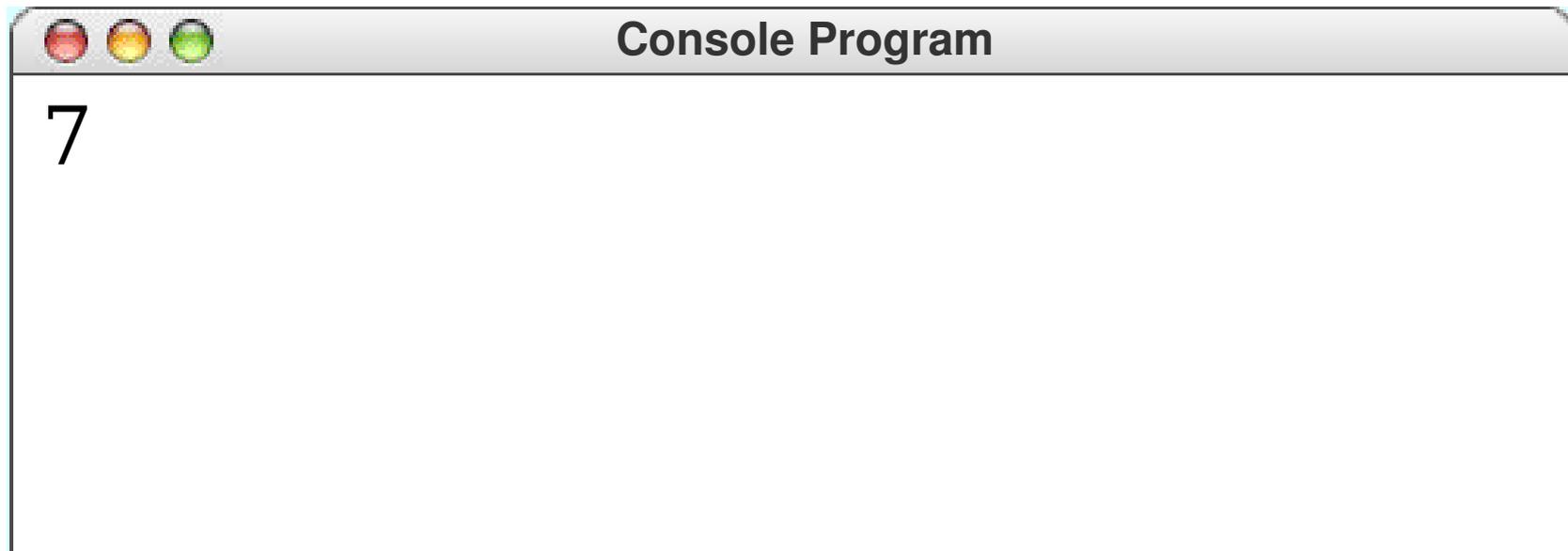
The while Loop

Example:

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

3

int x



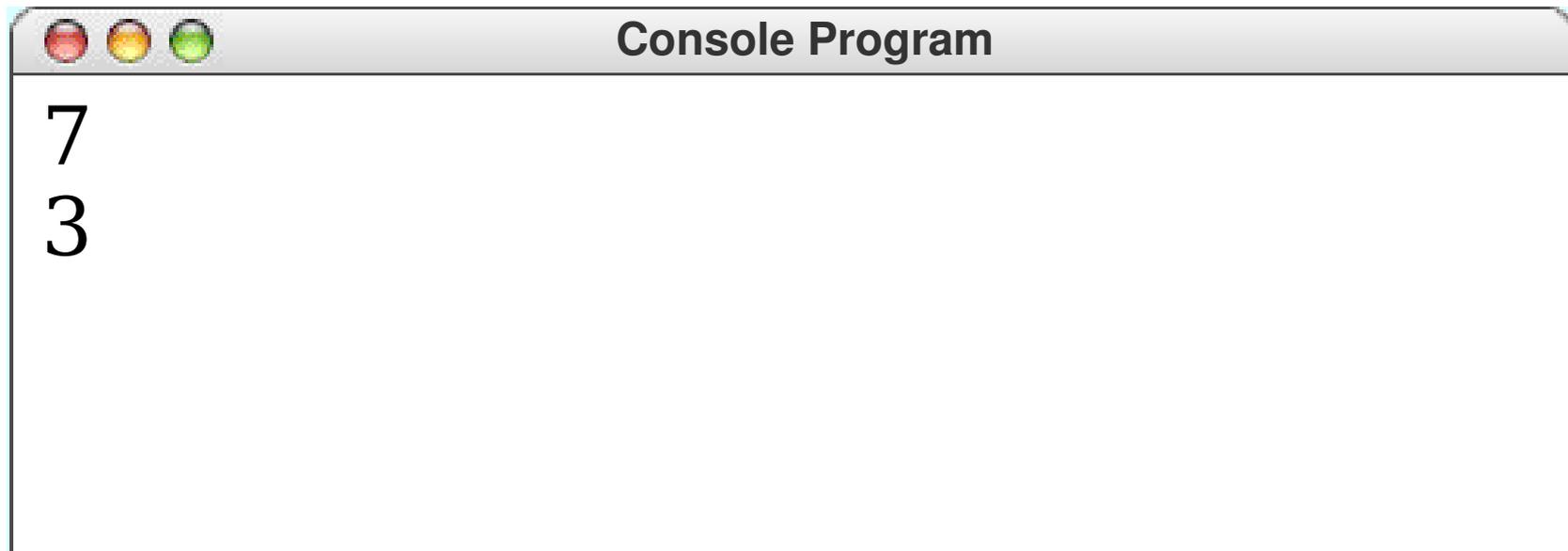
The while Loop

Example:

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

3

int x



Console Program

```
7  
3
```

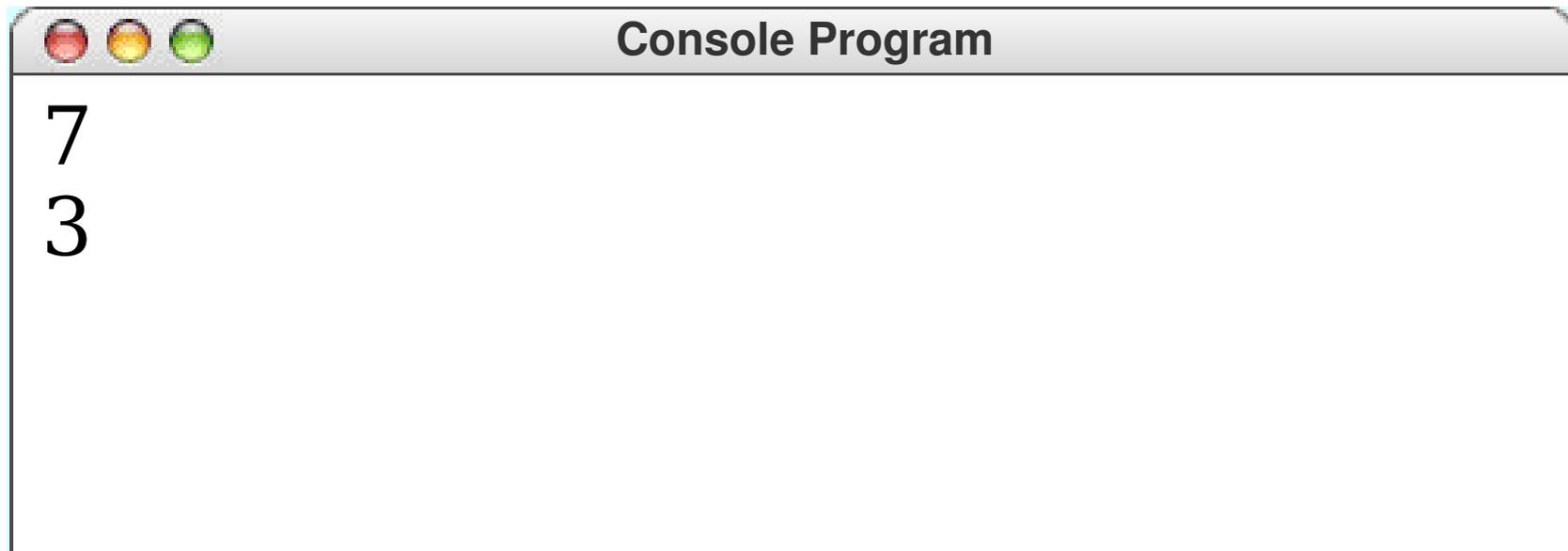
The while Loop

Example:

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

3

int x



Console Program

```
7  
3
```

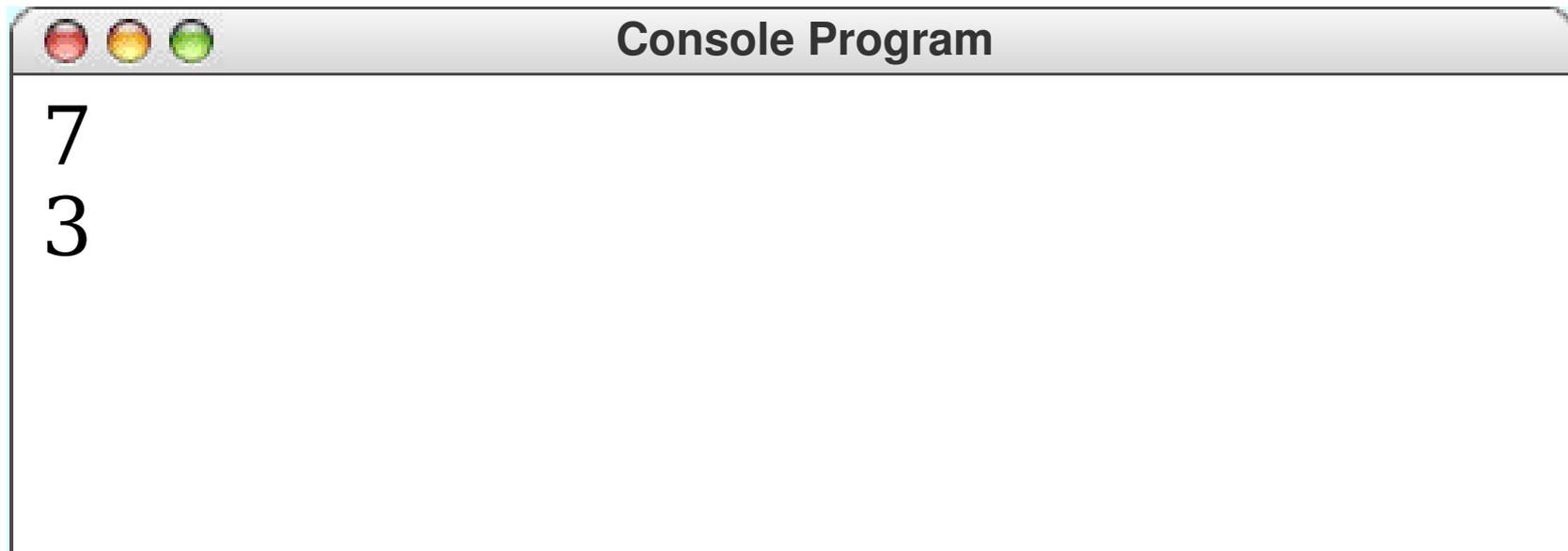
The while Loop

Example:

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

3

int x



Console Program

```
7  
3
```

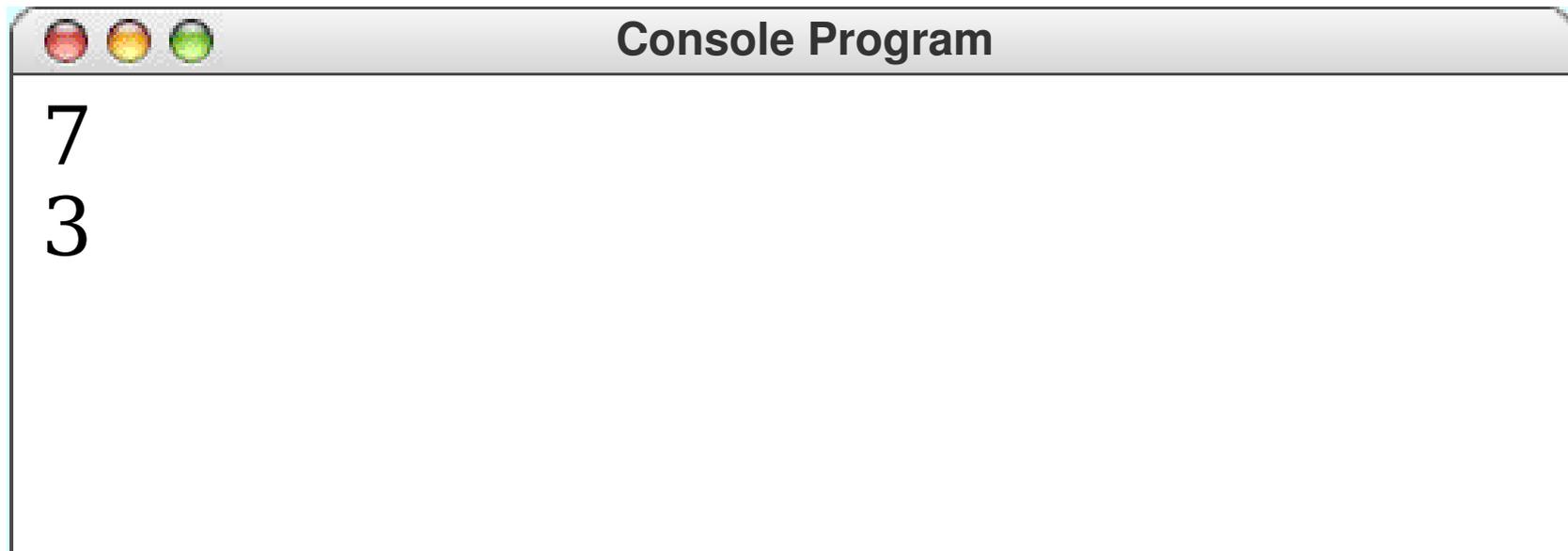
The while Loop

Example:

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

1

int x



Console Program

```
7  
3
```

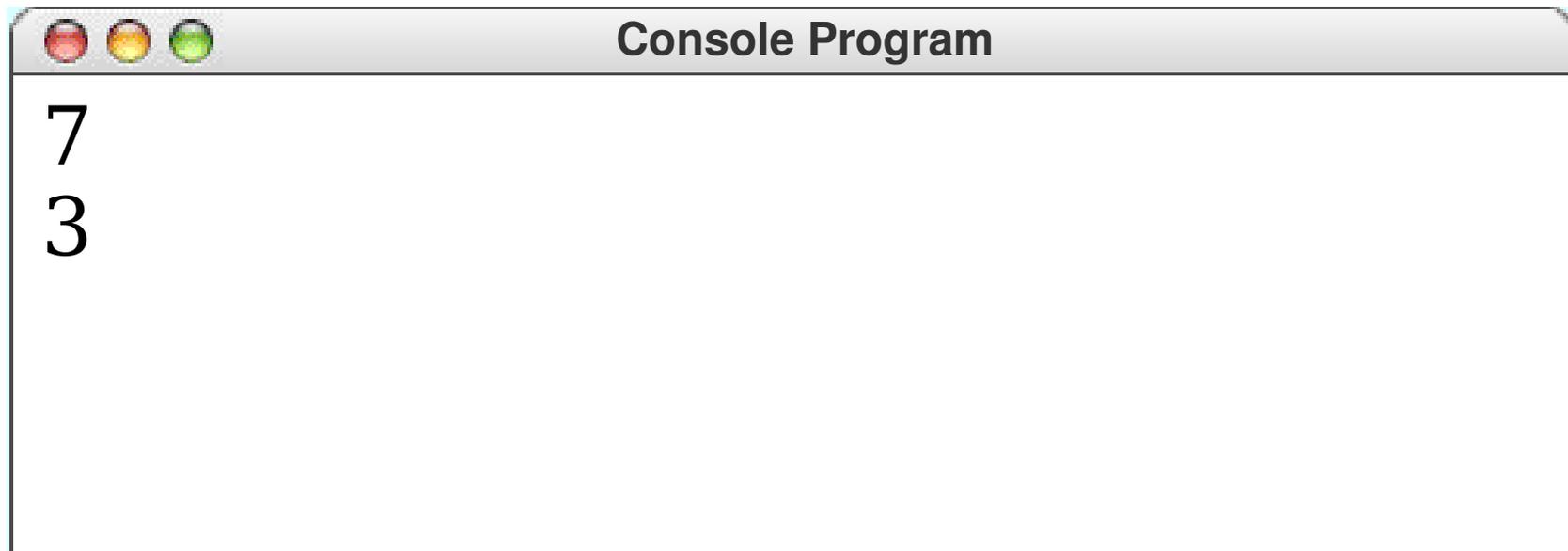
The while Loop

Example:

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

1

int x

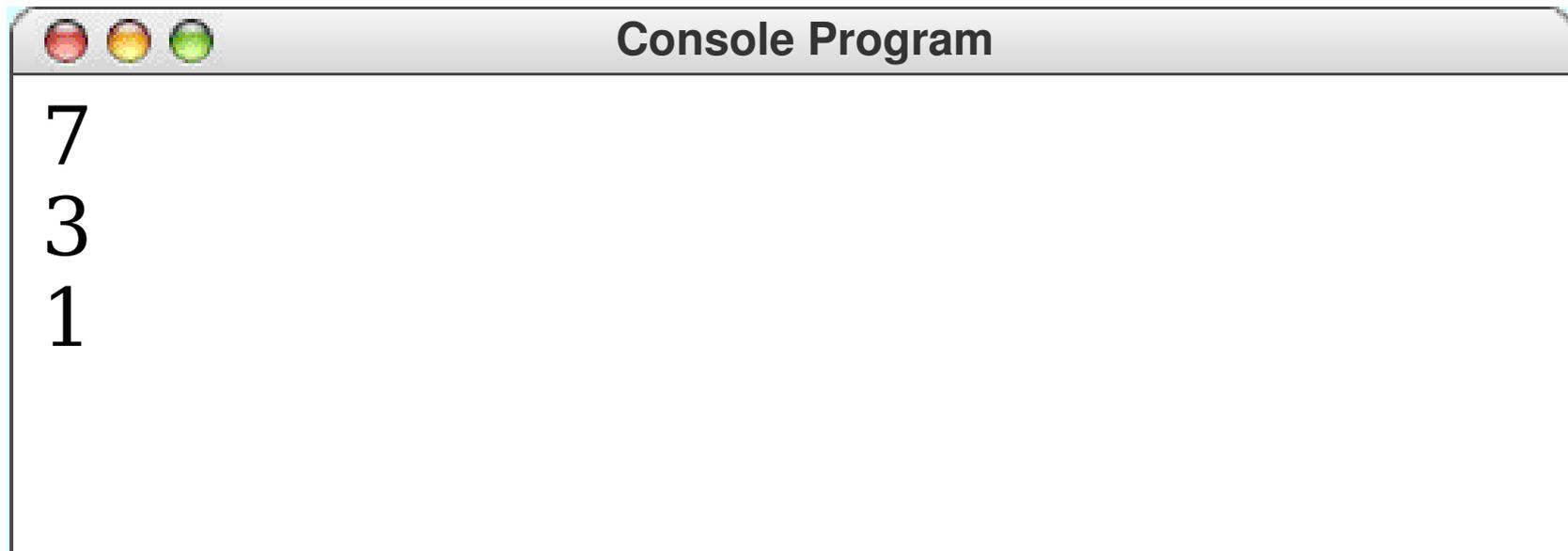


The while Loop

Example:

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

1 int x



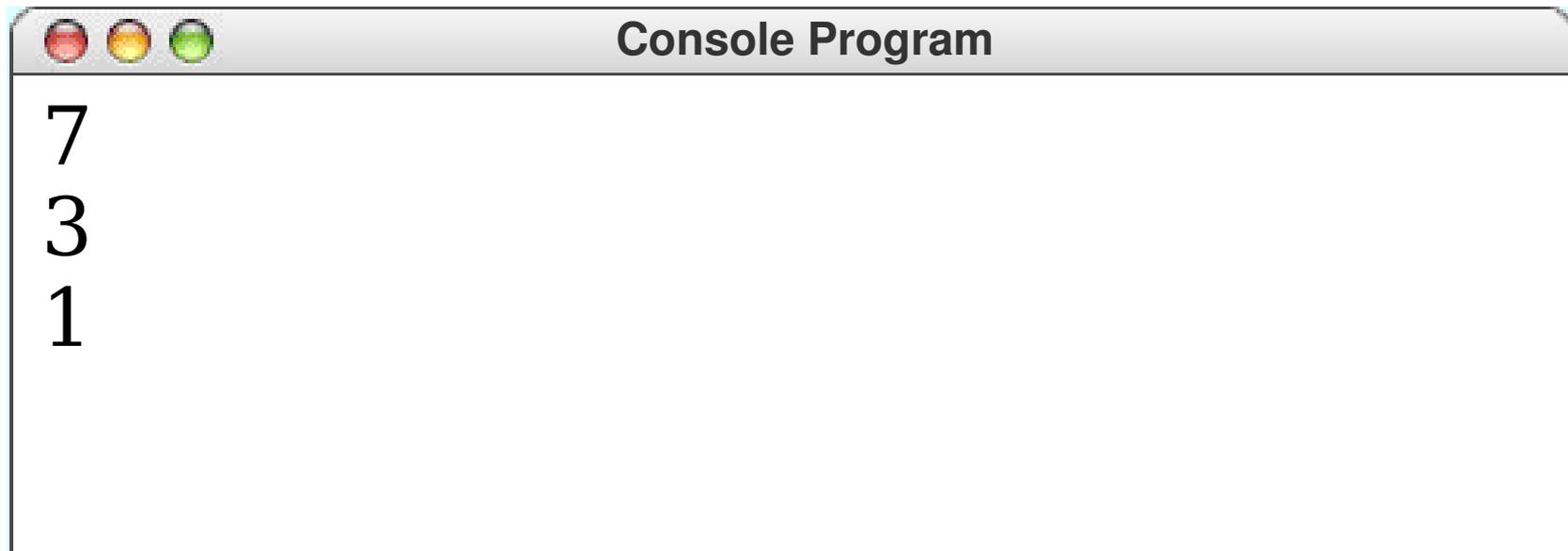
```
Console Program  
7  
3  
1
```

The while Loop

Example:

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

1 int x



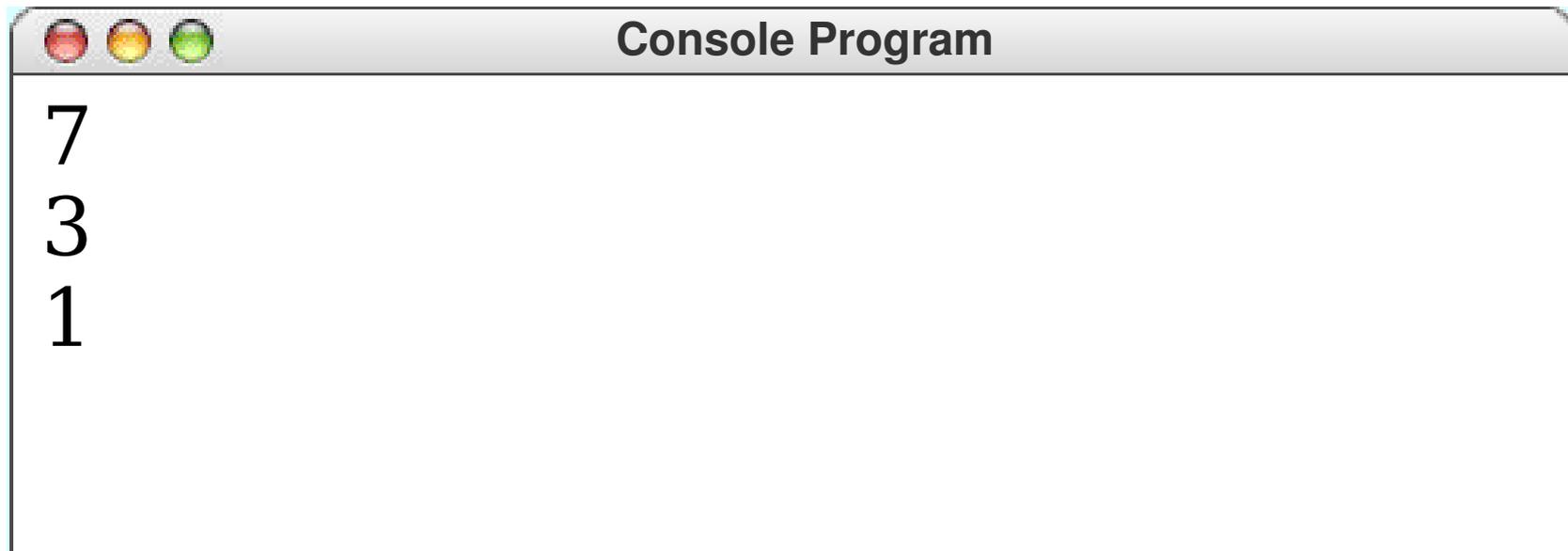
```
Console Program  
7  
3  
1
```

The while Loop

Example:

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

1 int x



```
Console Program  
7  
3  
1
```