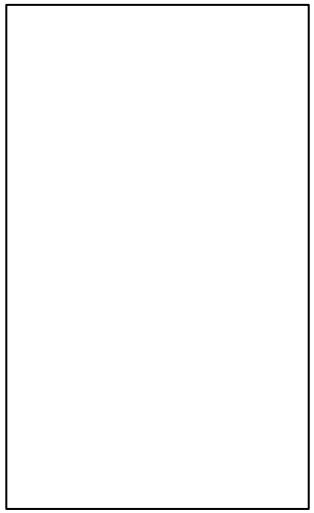
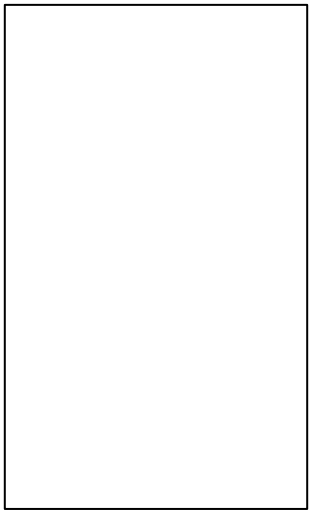
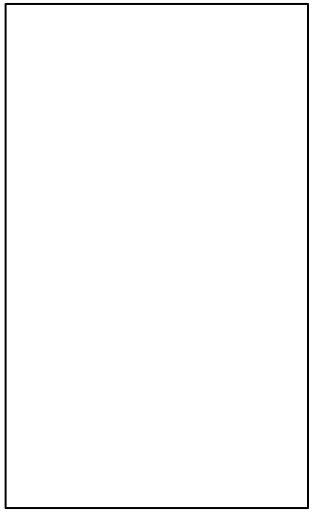
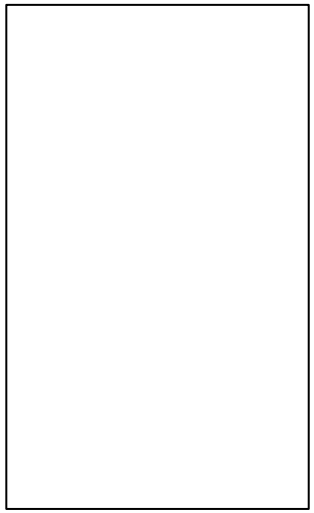
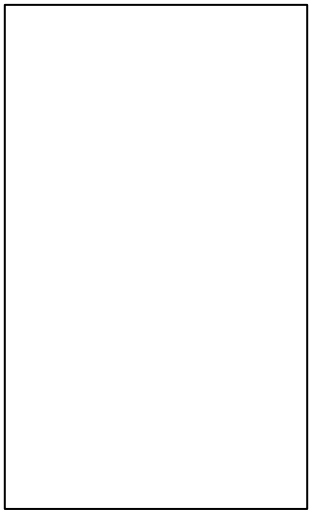
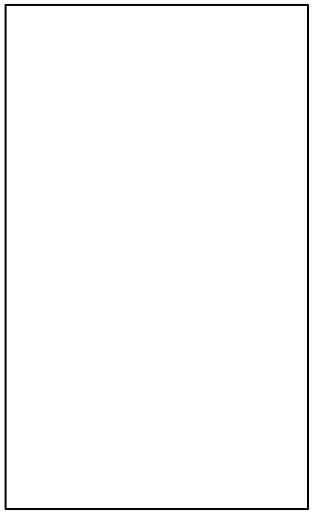
Fibonacci (iterativ)

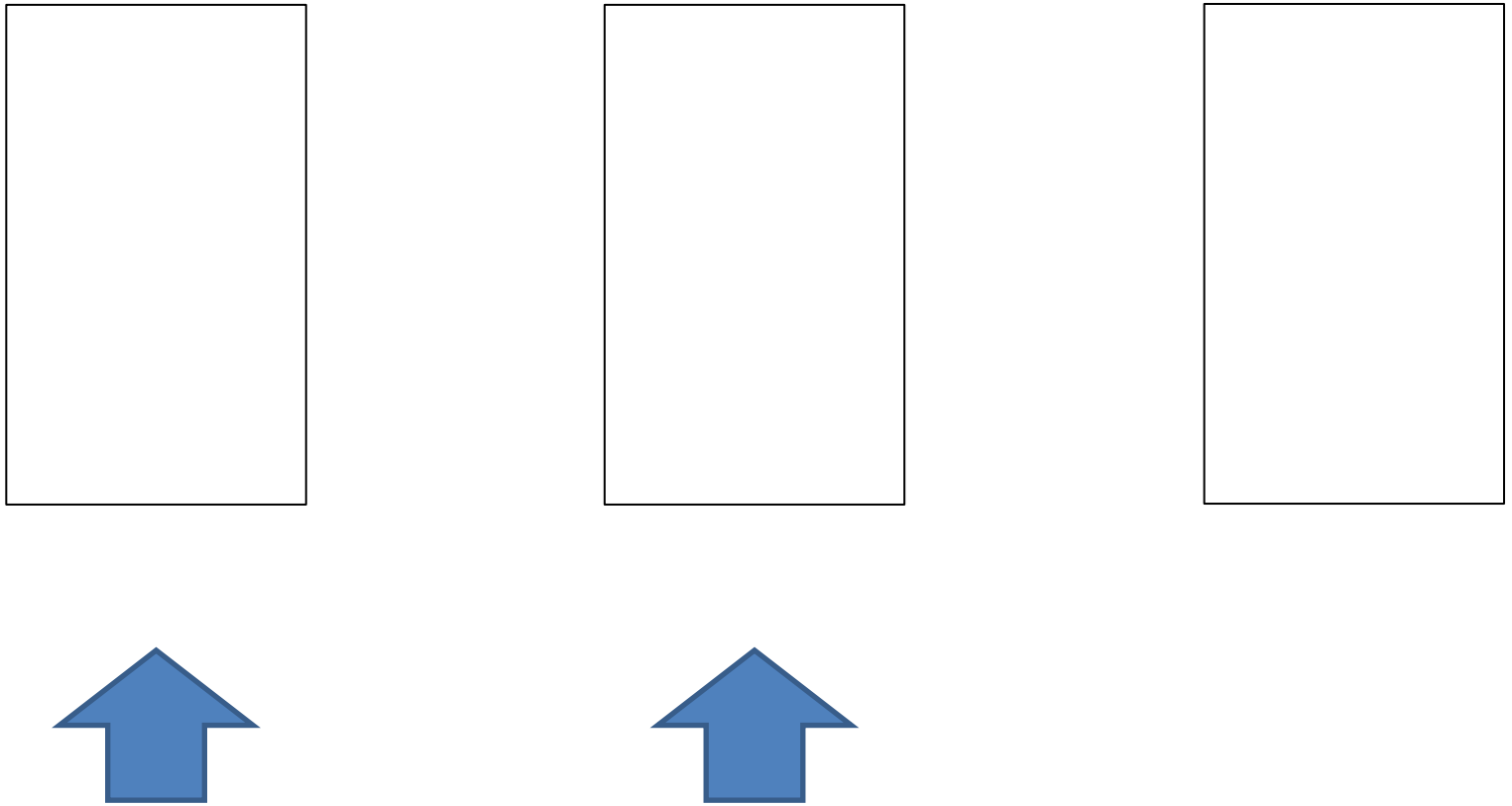
$f(0) = 0, \quad f(1) = 1, \quad f(n) = f(n-1) + f(n-2)$

```
fib := proc(n :: integer)
  local i, f, f1, f2;
  f1 := 0; f2 := 1; f := n;
  for i from 2 to n do
    f := f1 + f2;
    f1 := f2;
    f2 := f;
  end do;
  return f;
end proc :
```

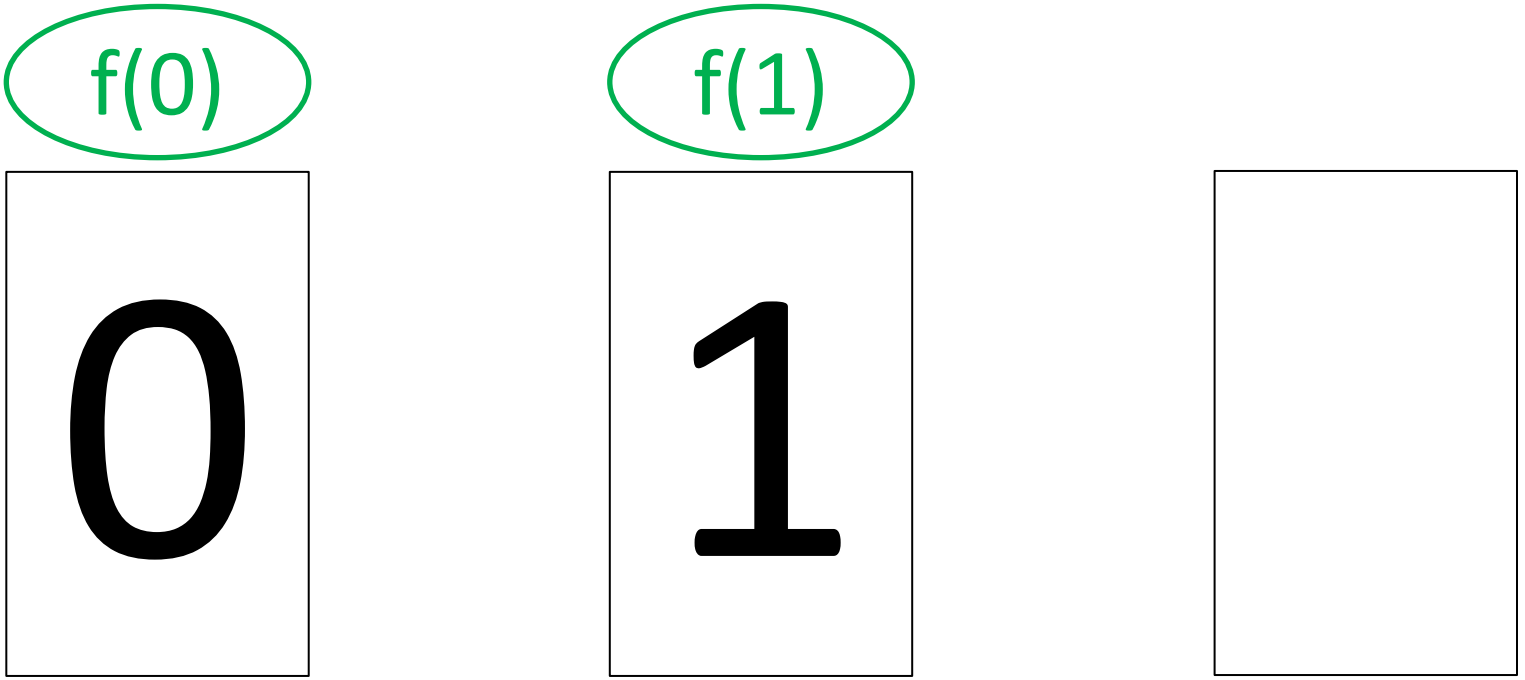




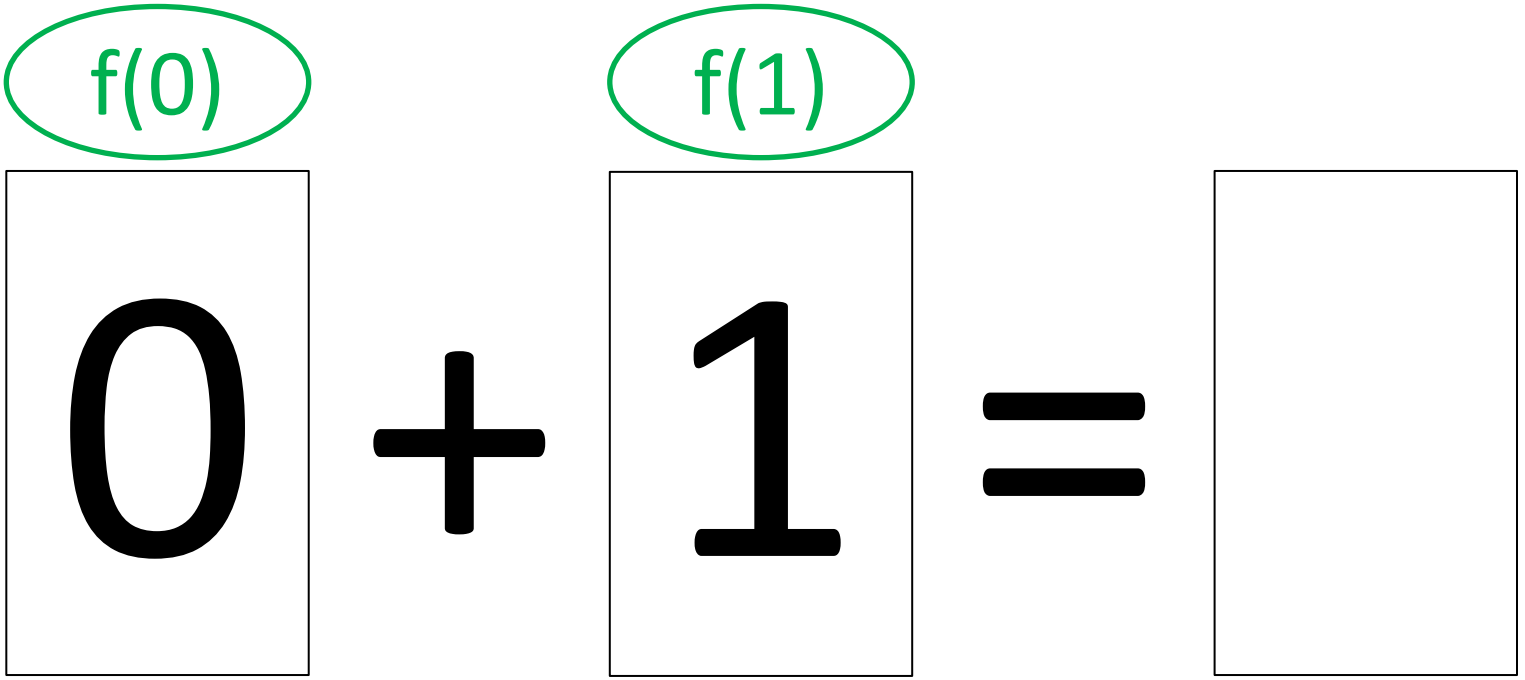




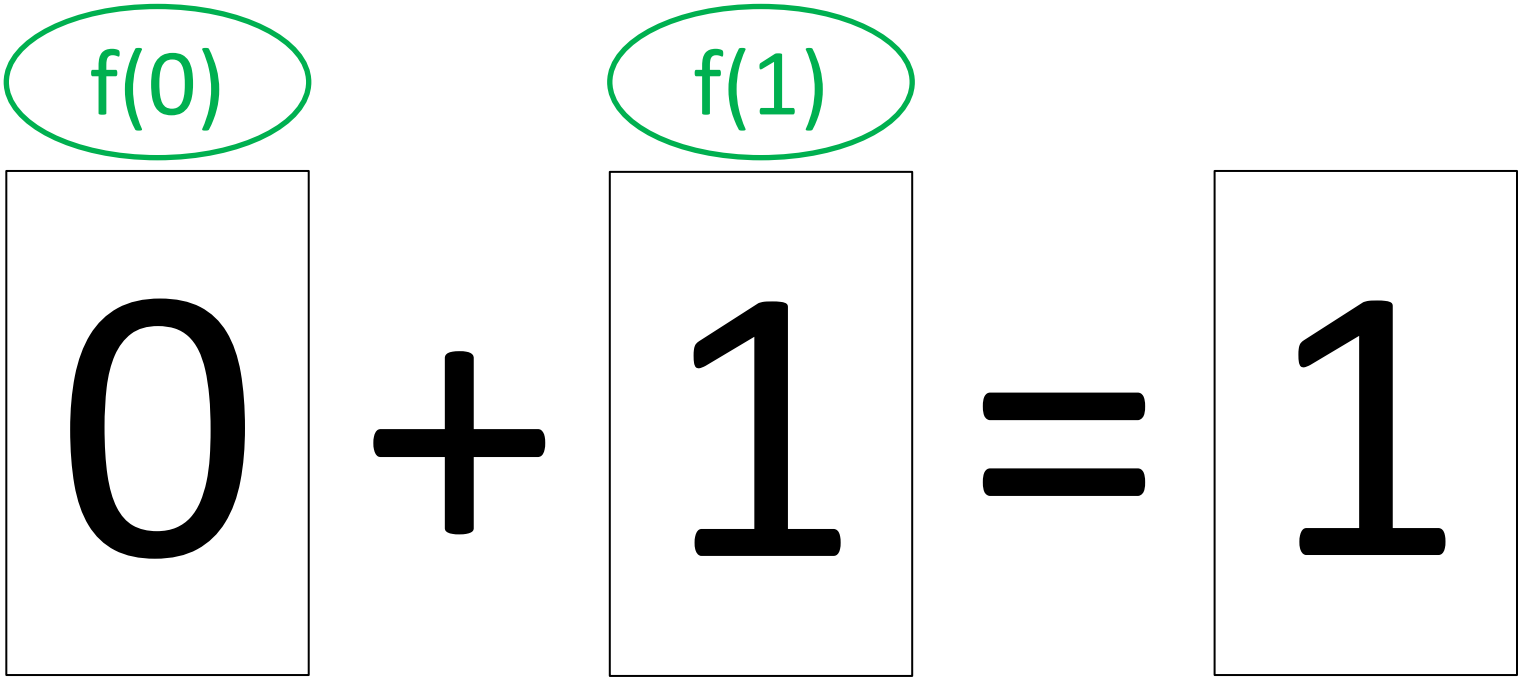
f(0) = 0 f(1) = 1

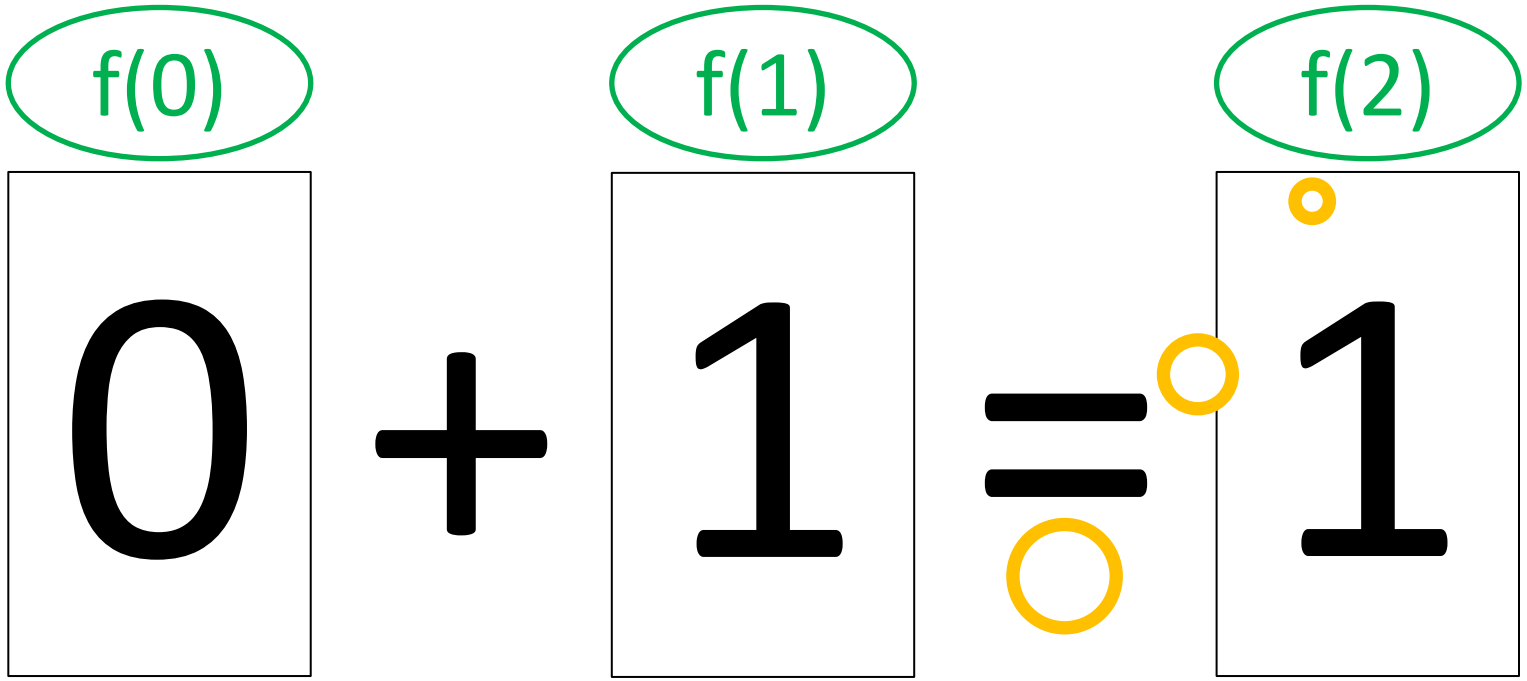


f(0) = 0 f(1) = 1

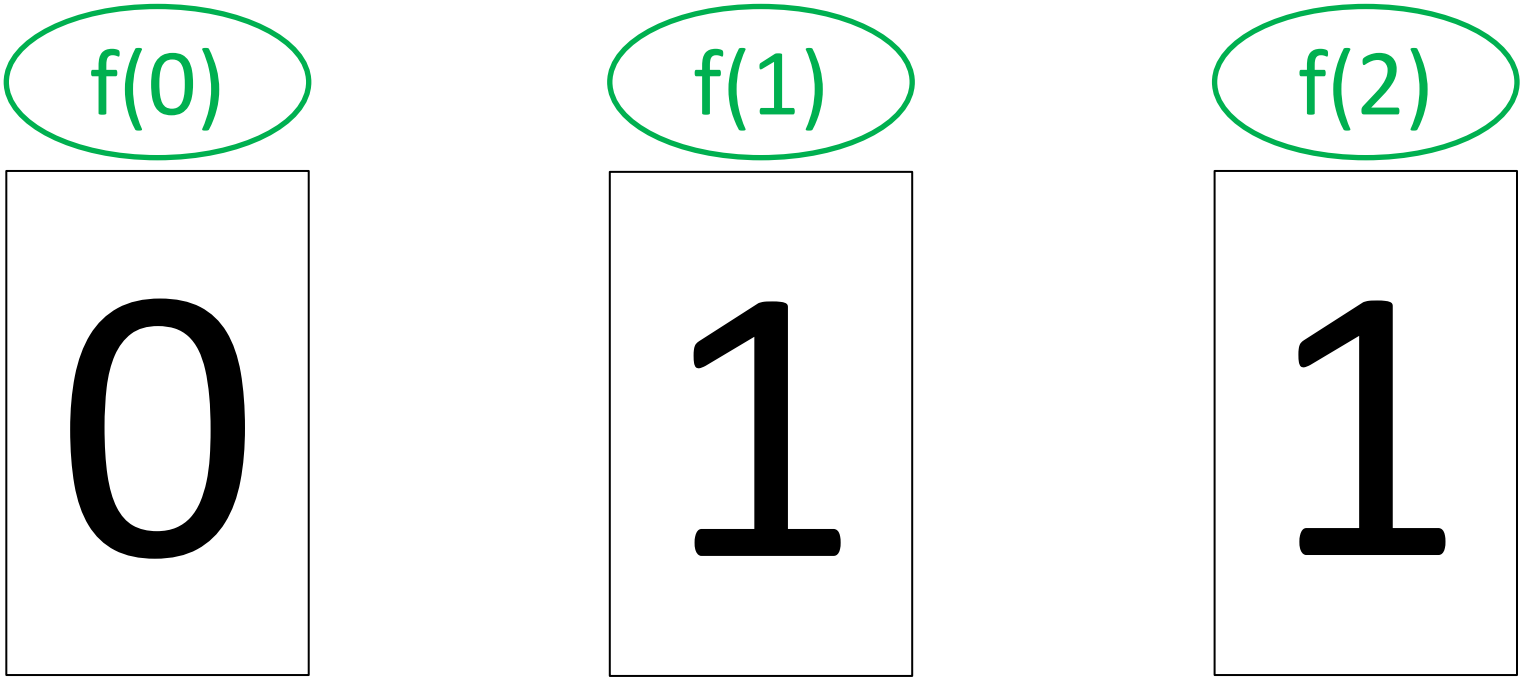

f(0)
f(1)
0
1


f(0)
f(1)
0
+
1
=


f(0)
f(1)
0
+
1
=
1



Schleife beginnt
bei n = 2.


f(0)
0
f(1)
1
f(2)
1

f(0)

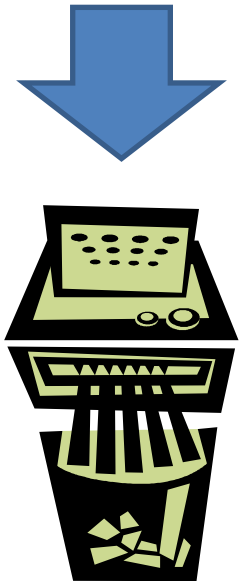
0

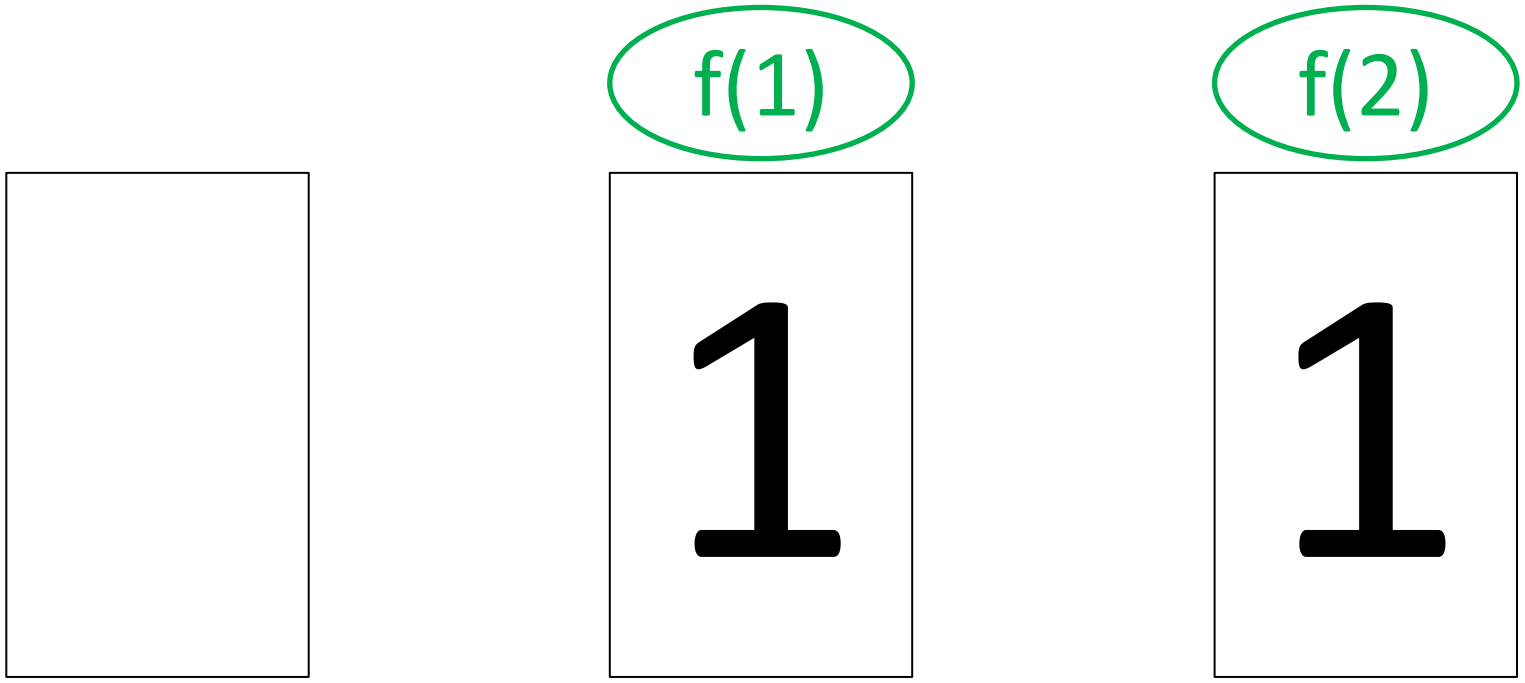
f(1)

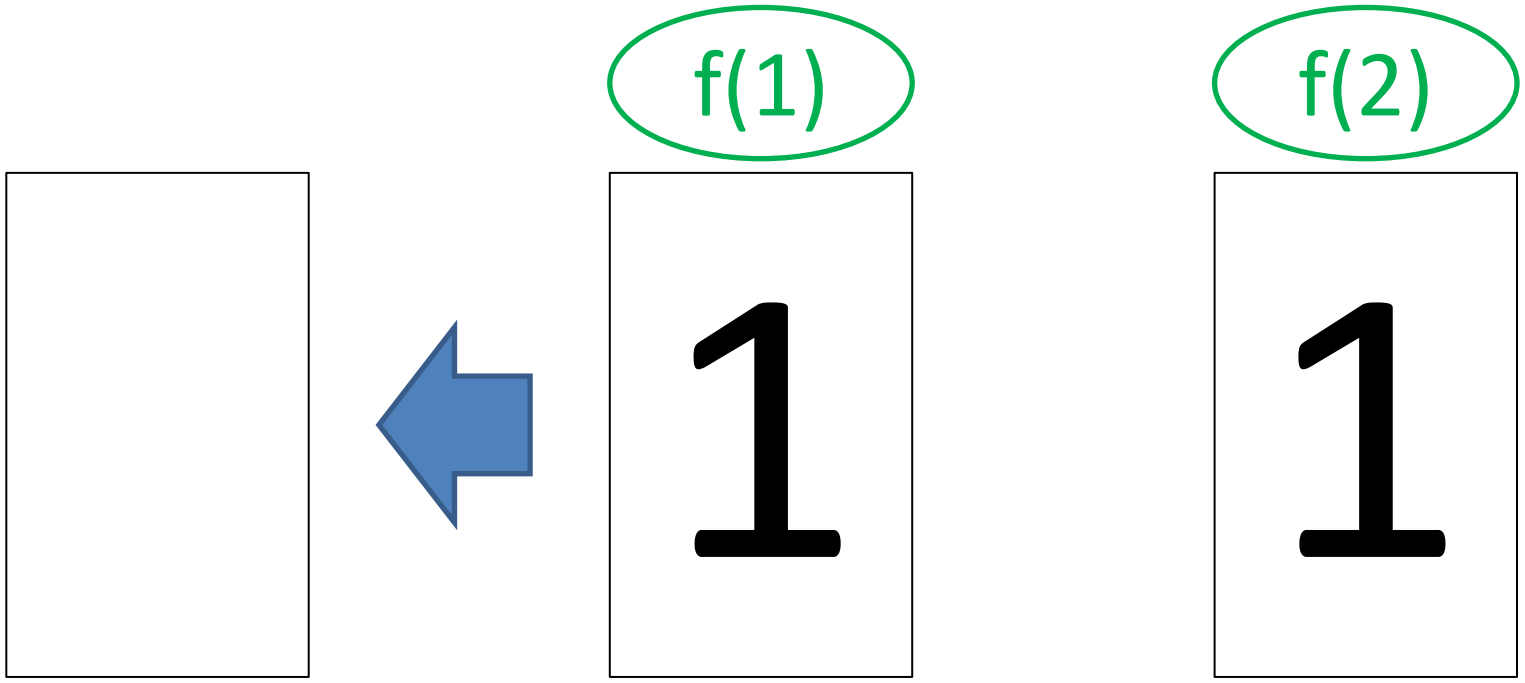
1

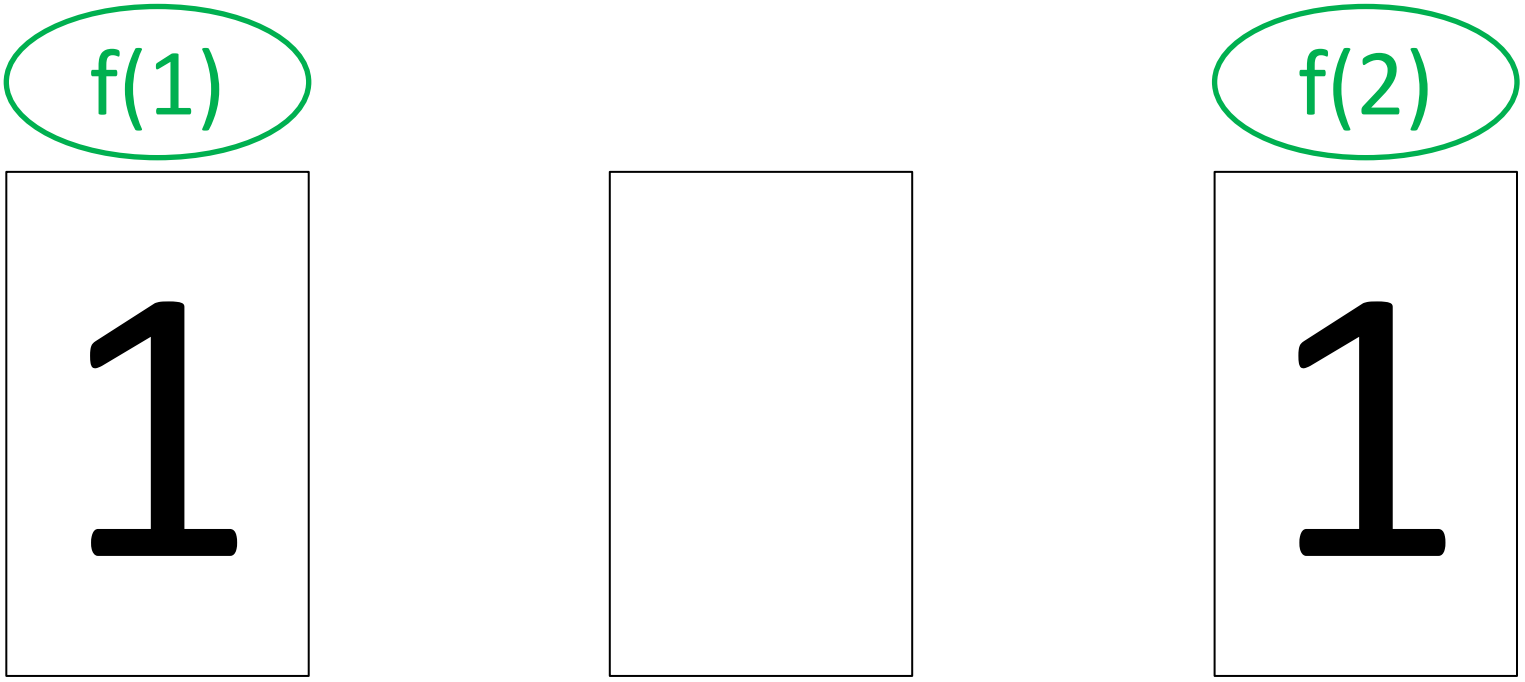
f(2)

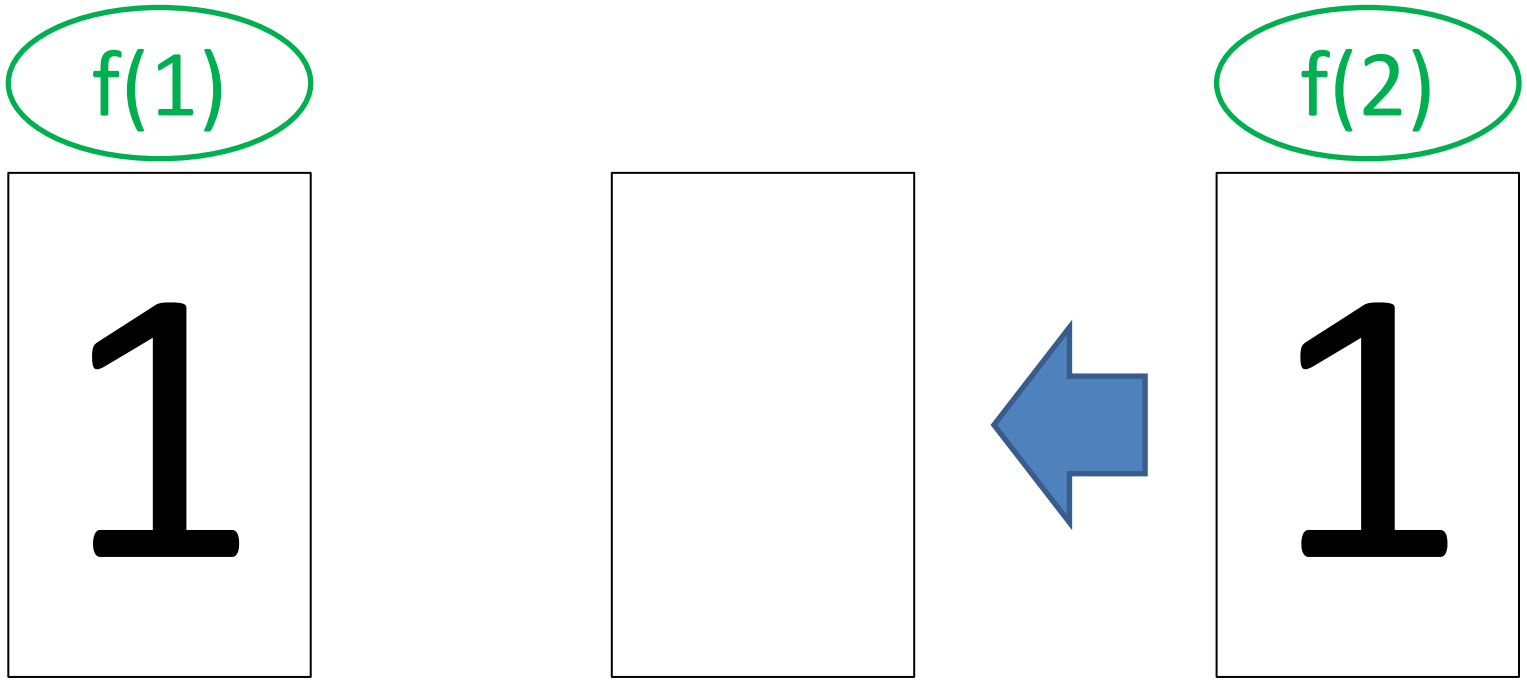
1

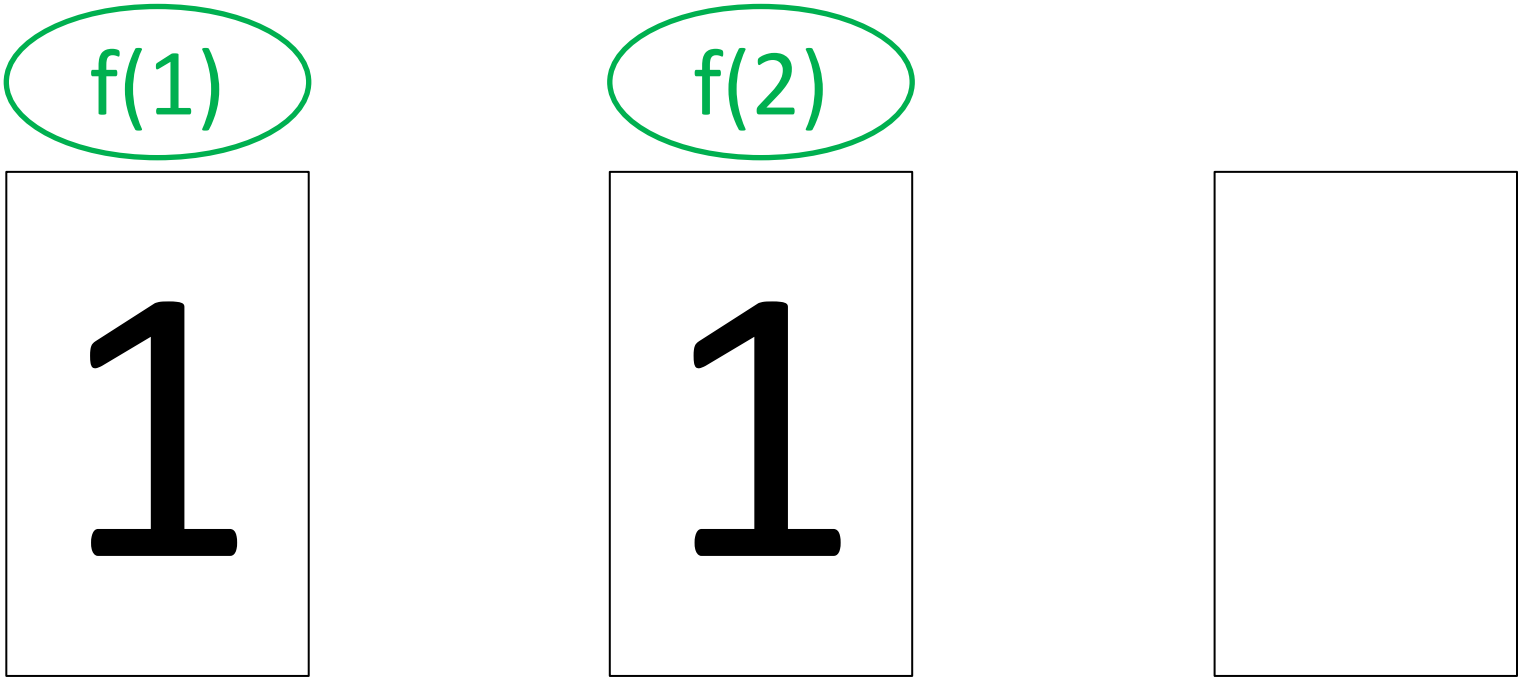


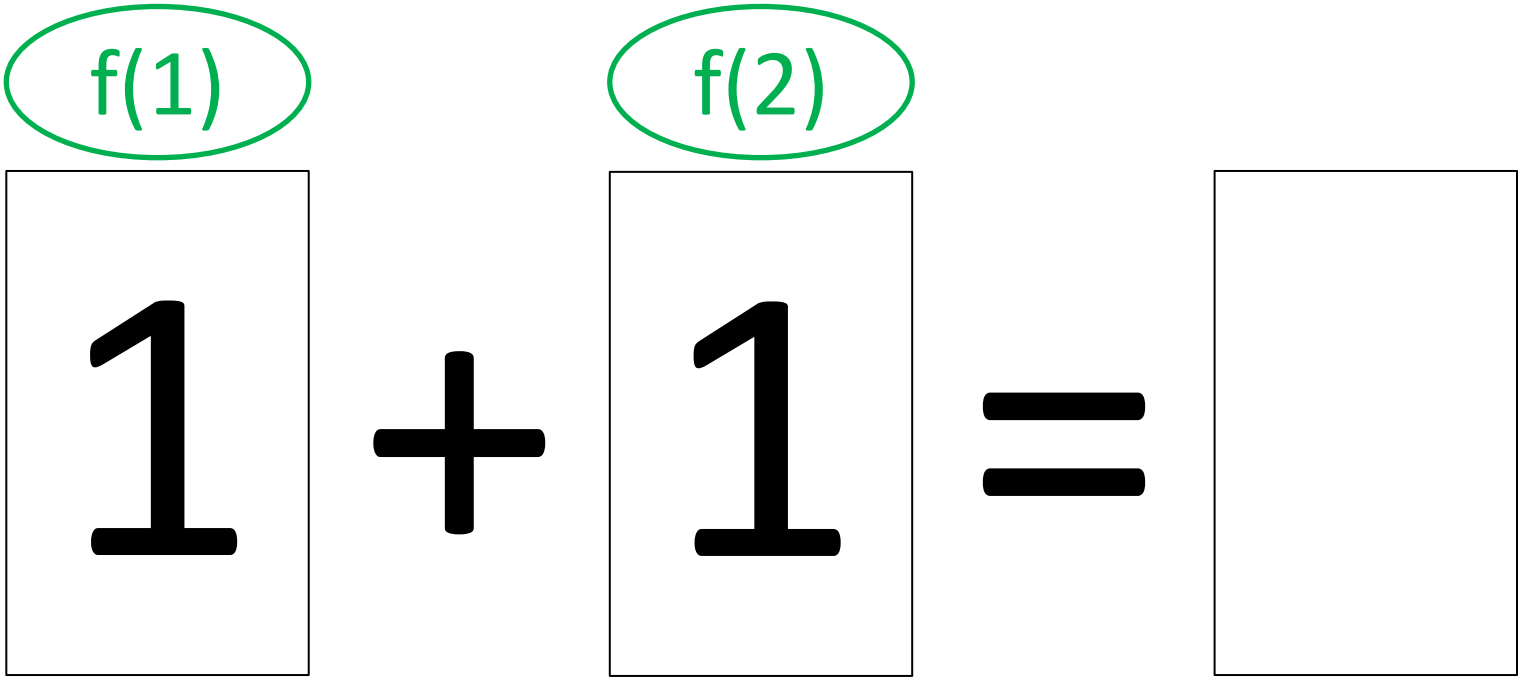

f(1)
f(2)
1
1

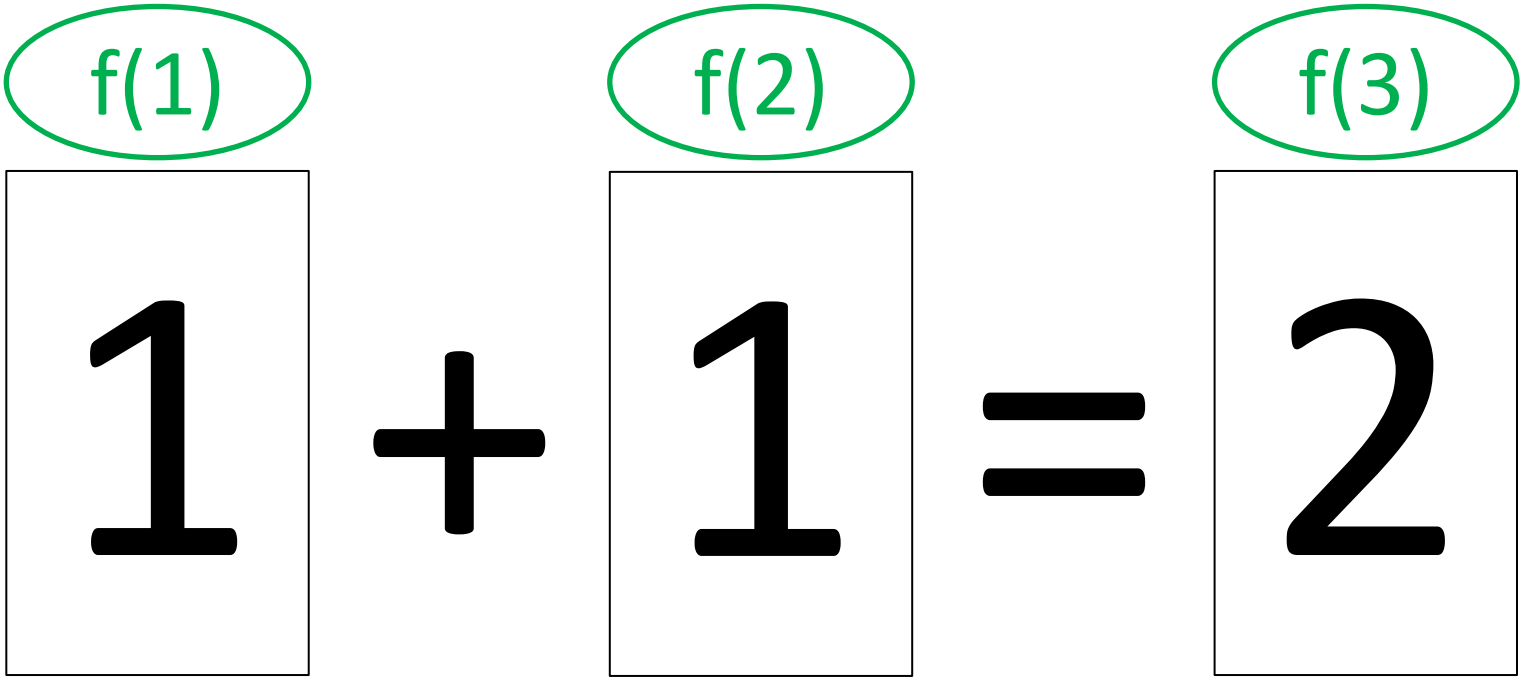

f(1)
f(2)
1
1

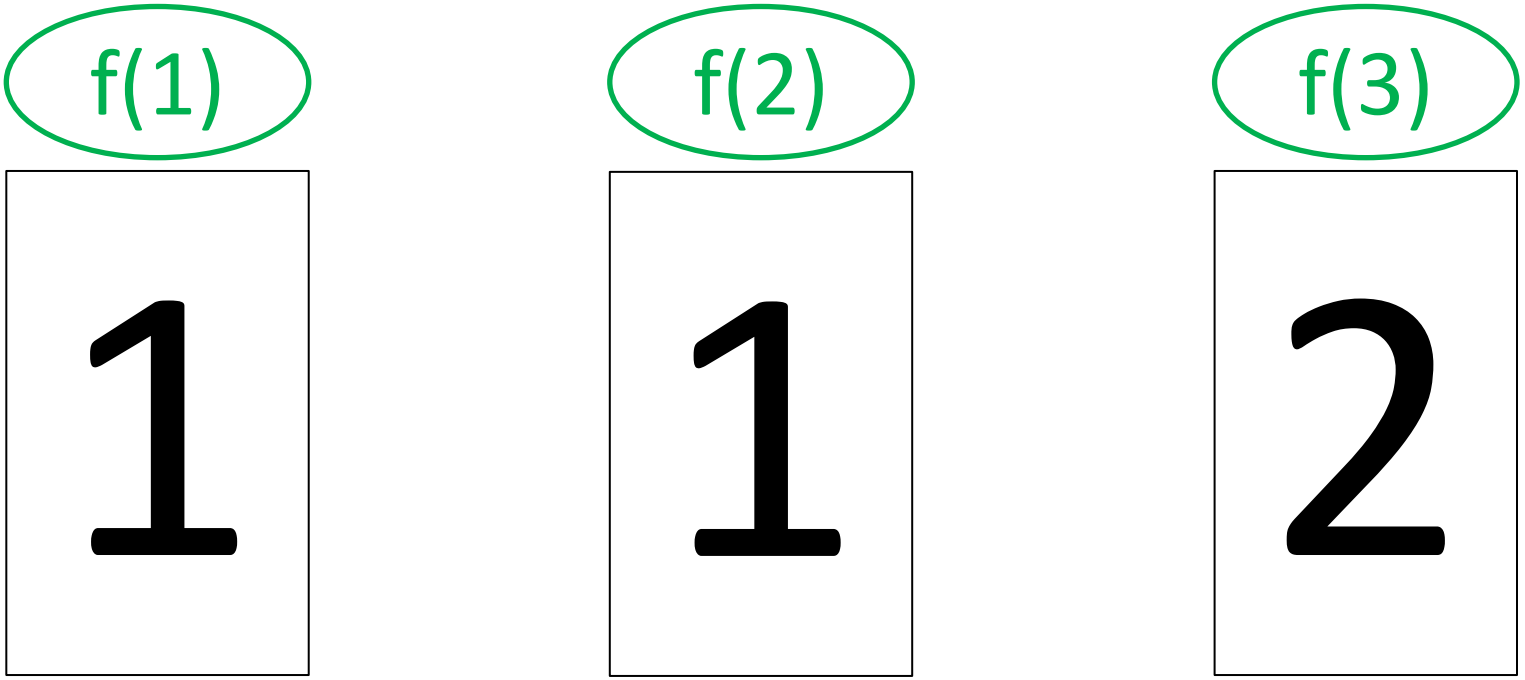

f(1)
1
f(2)
1


f(1)
1
f(2)
1


f(1)
1
f(2)
1


f(1)
f(2)
1
+
1
=


f(1)
f(2)
f(3)
1
+
1
=
2


f(1)
1
f(2)
1
f(3)
2

f(1)

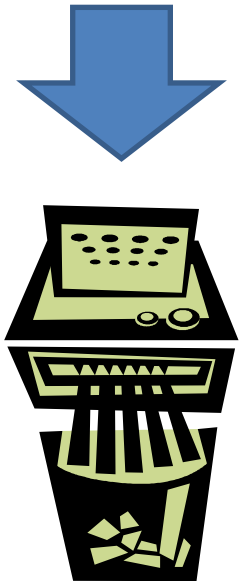
1

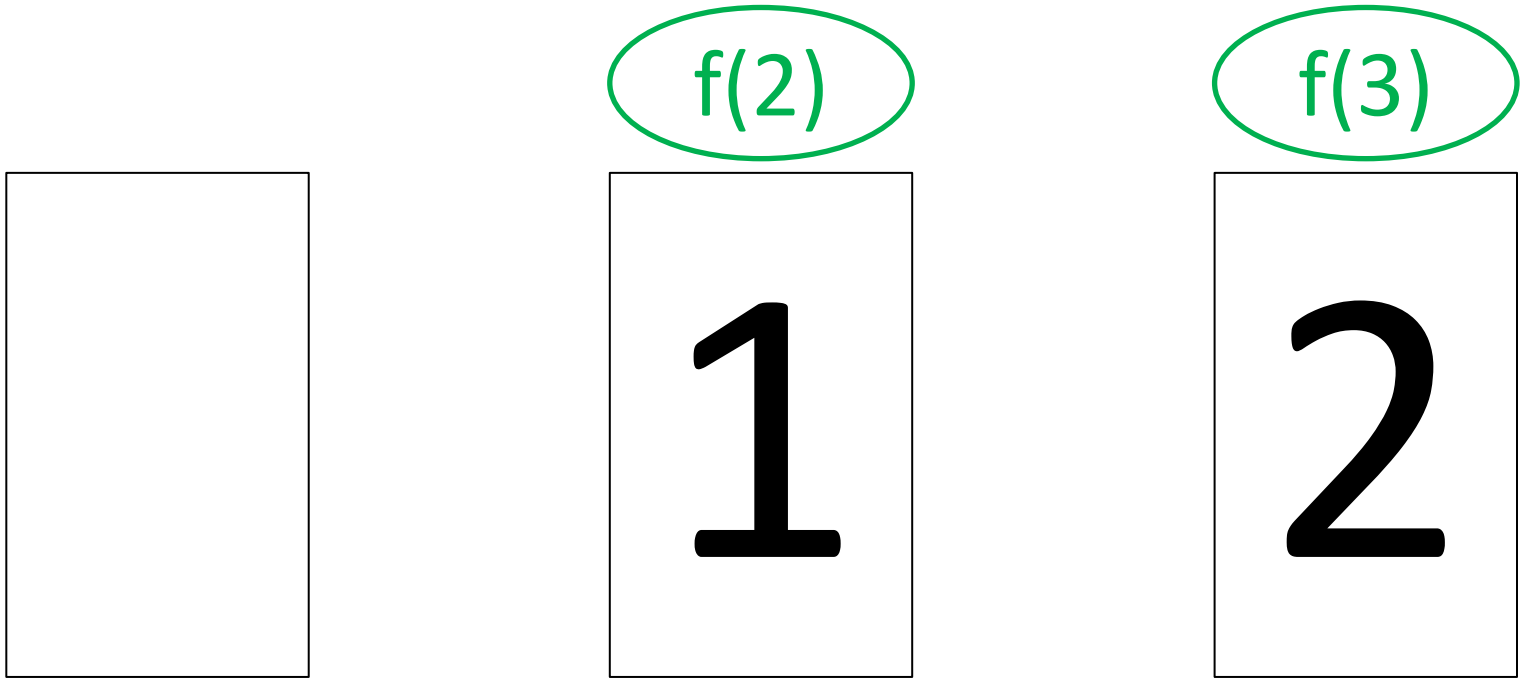
f(2)

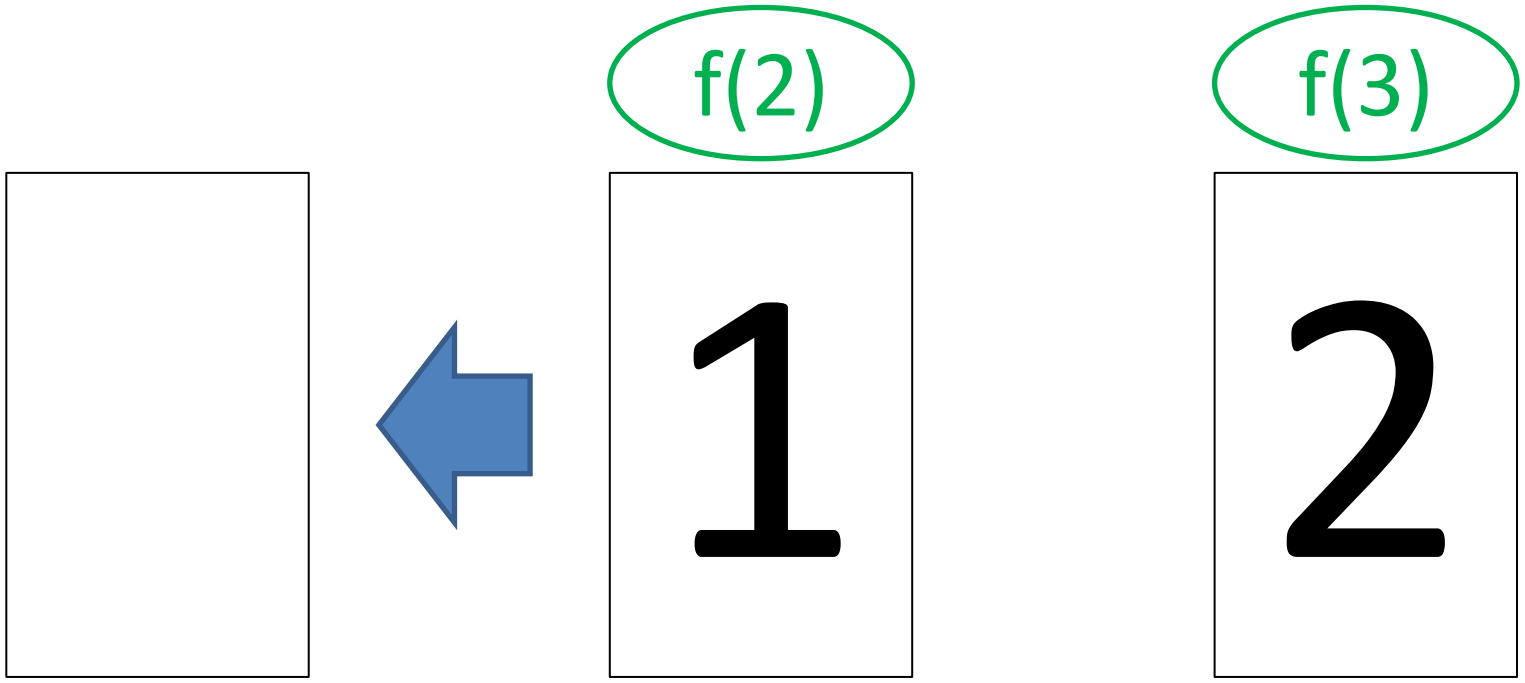
1

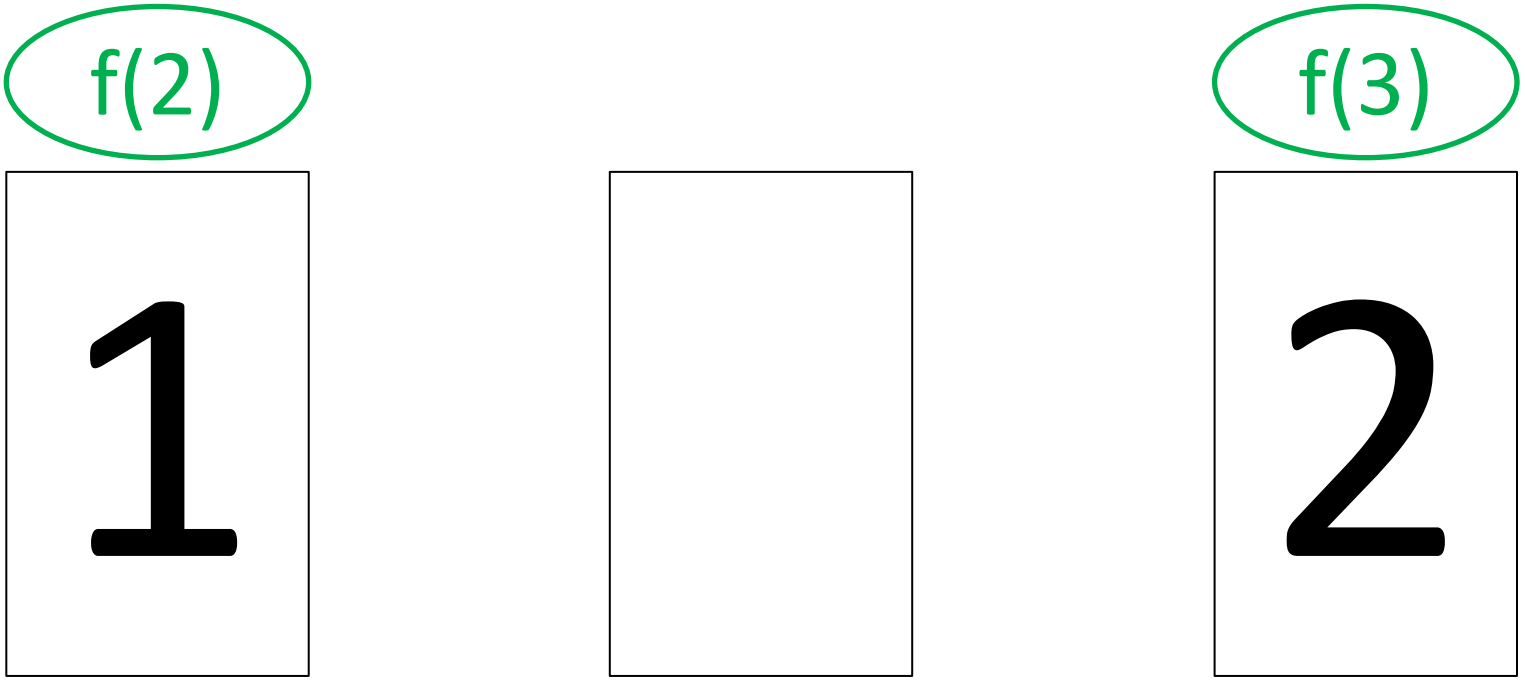
f(3)

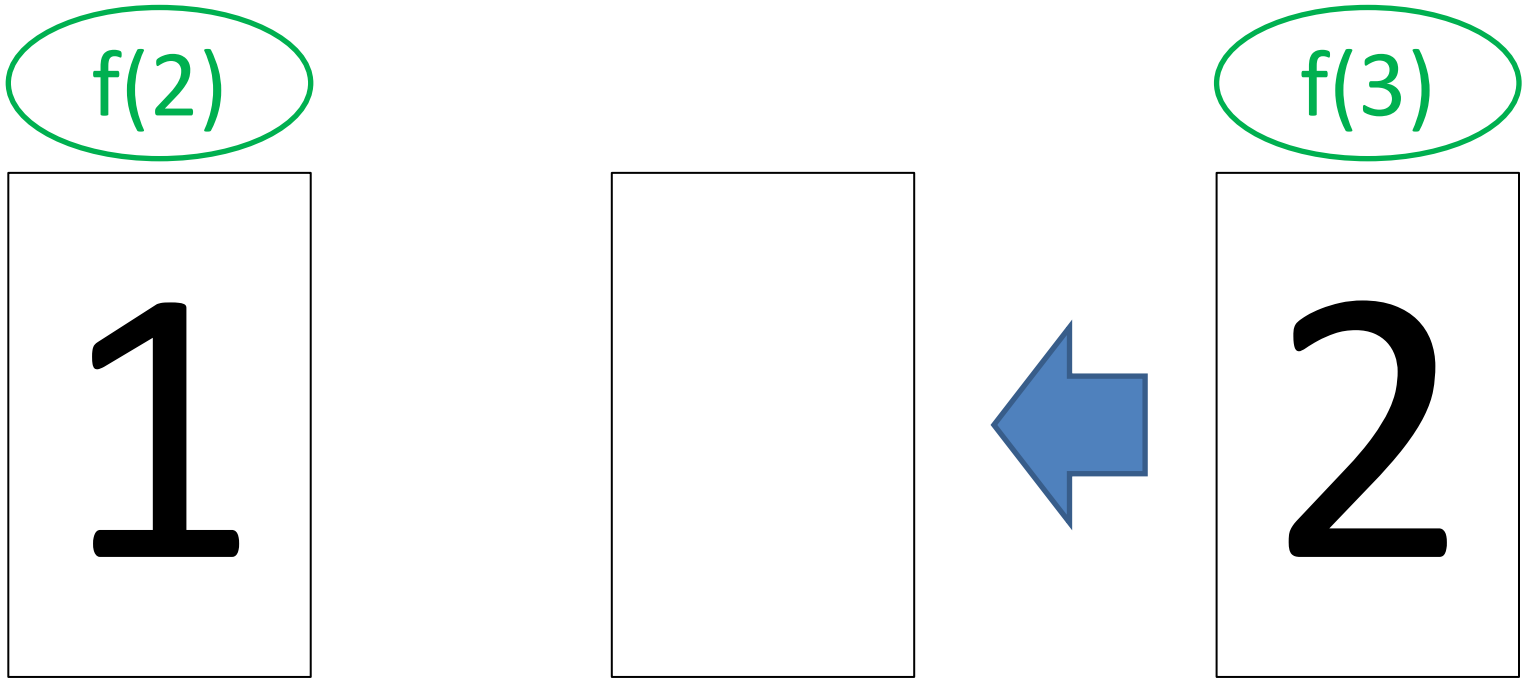
2

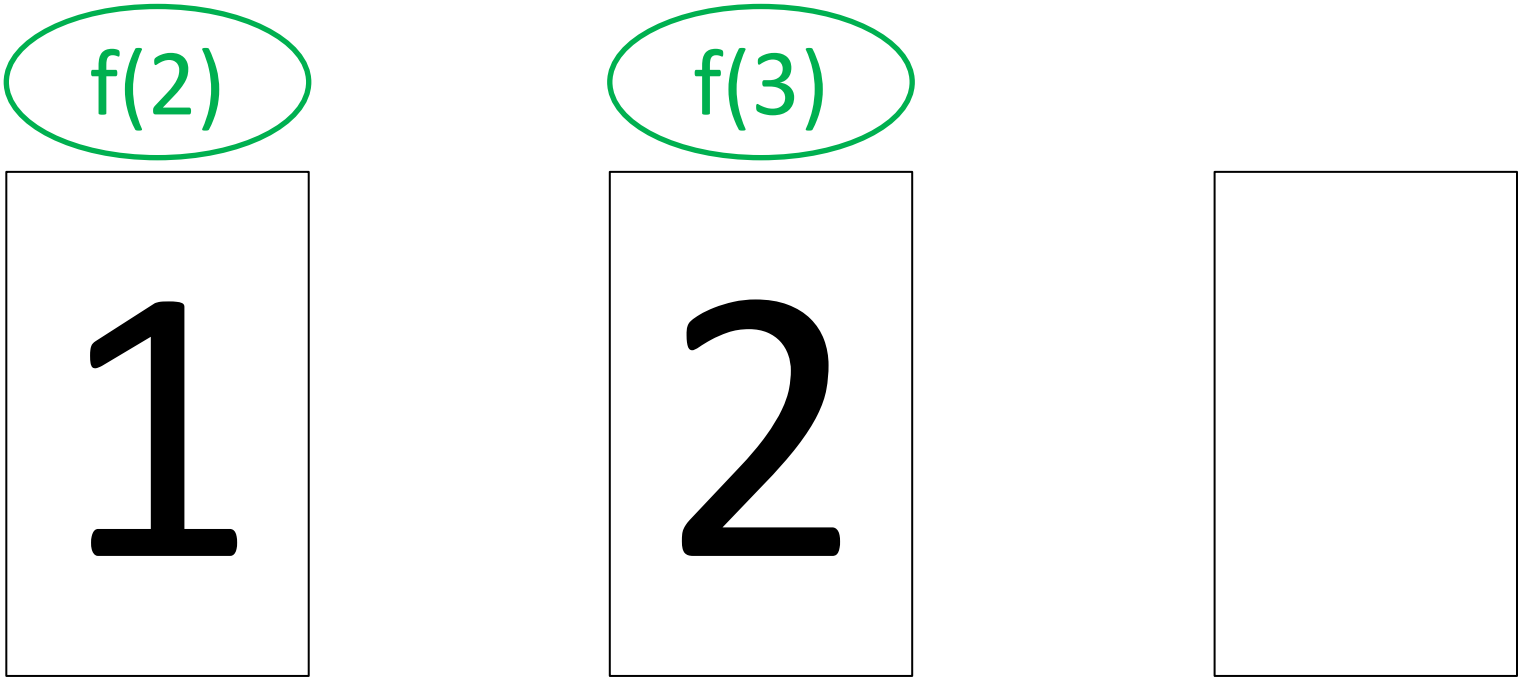



f(2)
f(3)
1
2

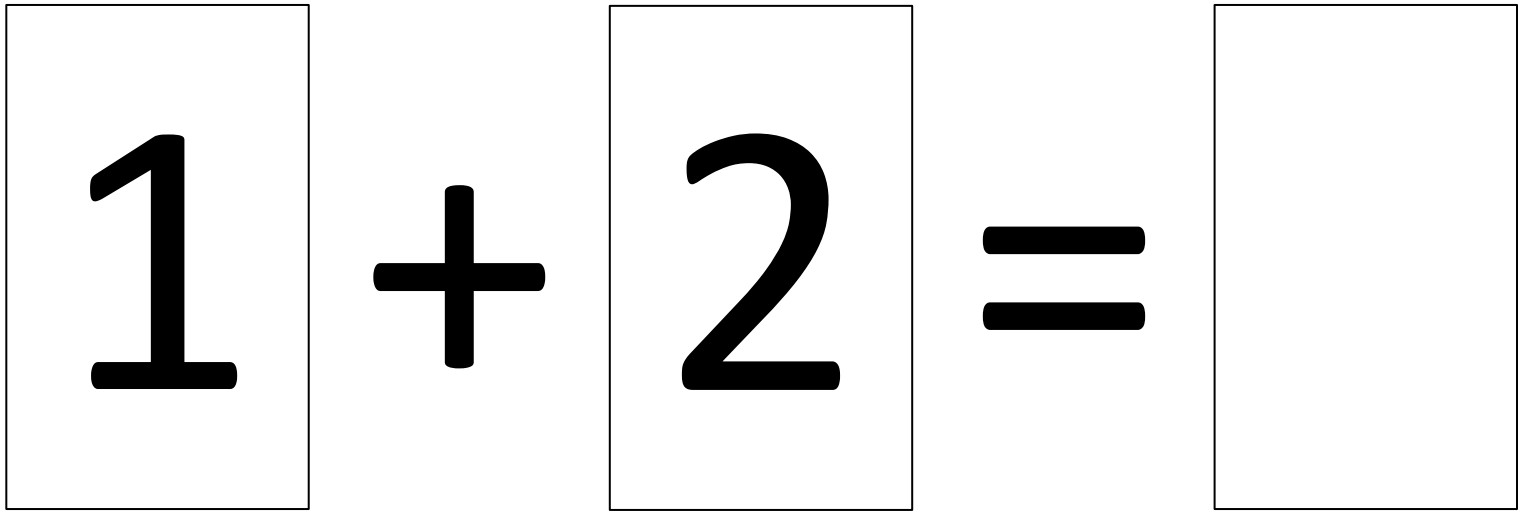

f(2)
f(3)
1
2

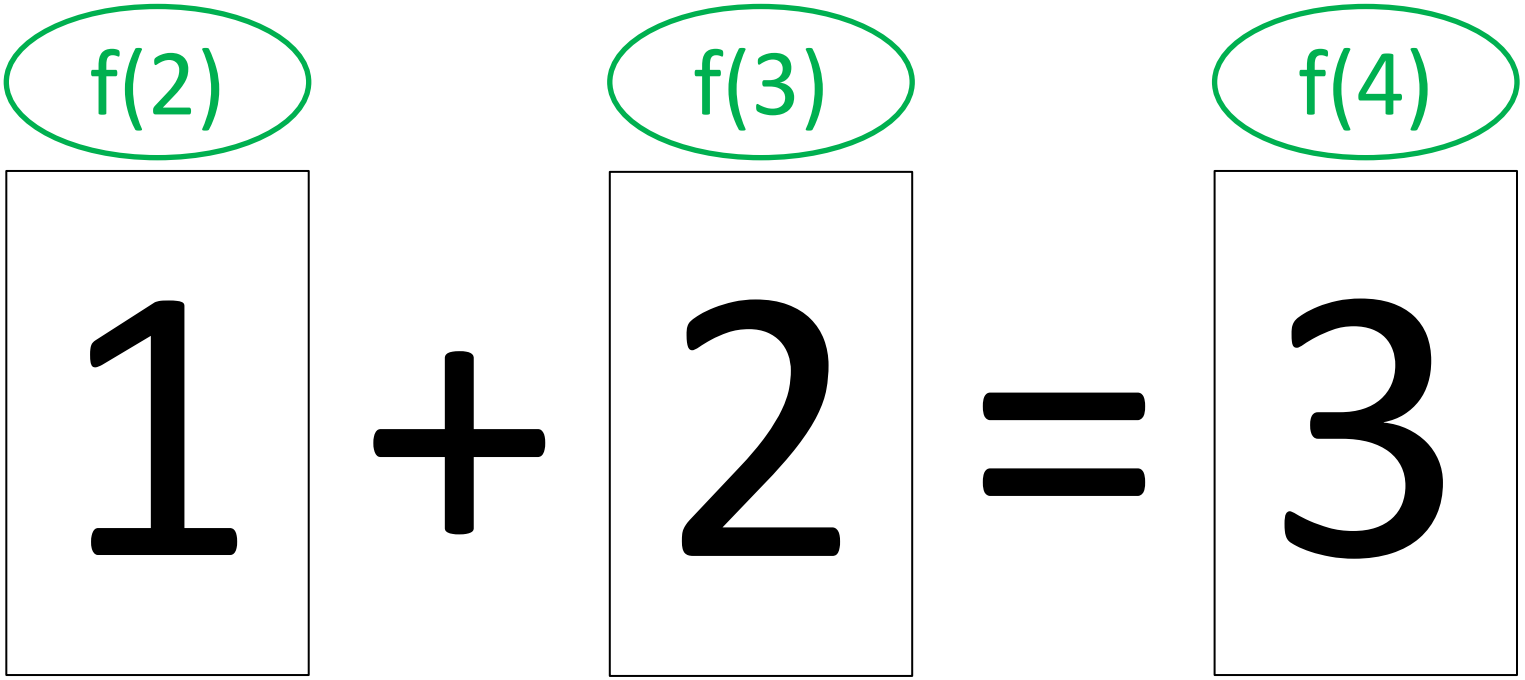

f(2)
1
f(3)
2

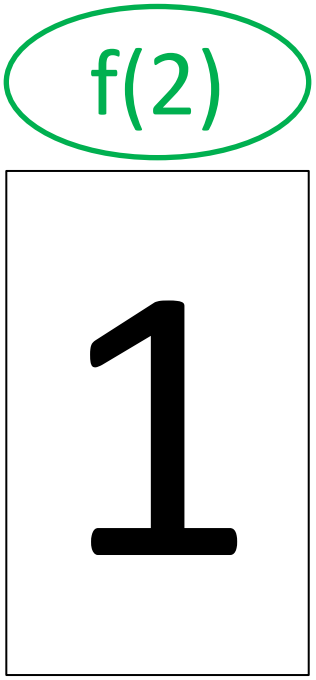

f(2)
f(3)
1
2

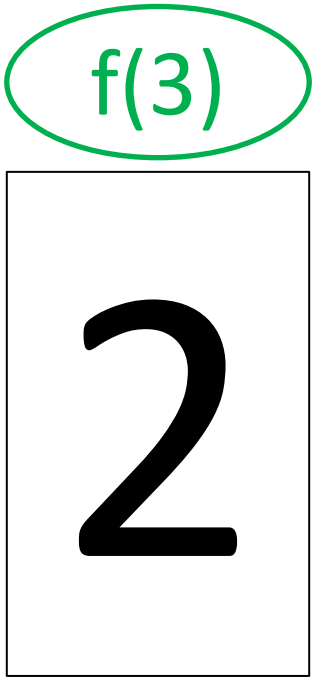

f(2)
f(3)
1
2

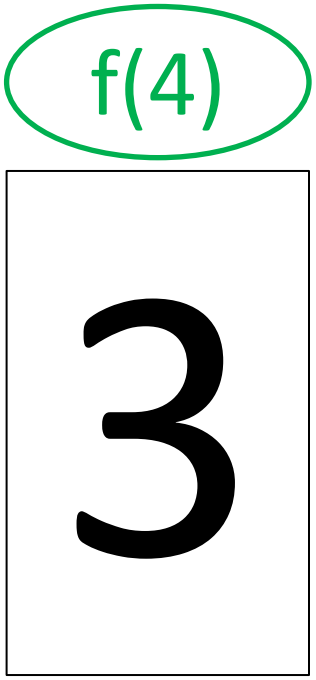
f(2) f(3)




f(2)
f(3)
f(4)
1
+
2
=
3


f(2)
1


f(3)
2


f(4)
3

f(2)

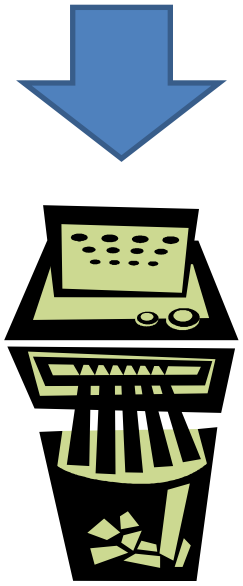
1

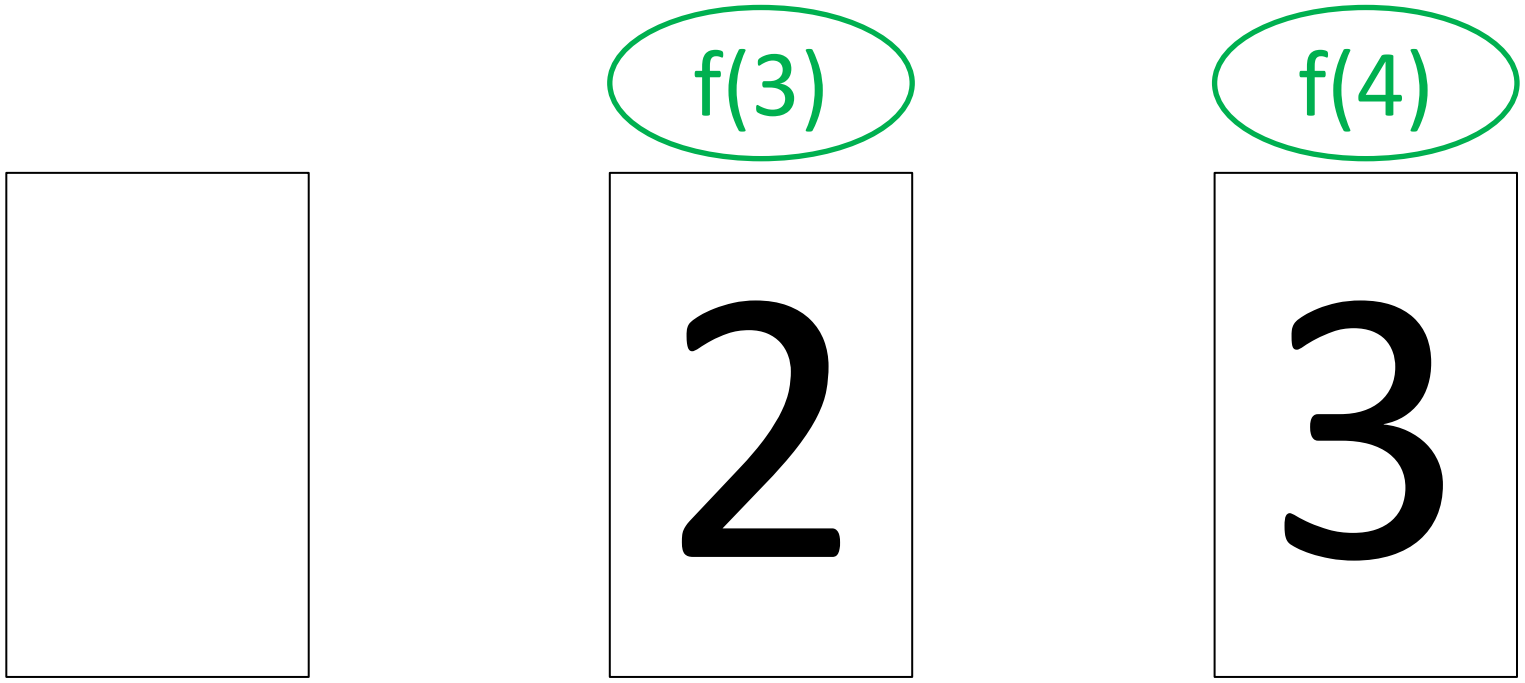
f(3)

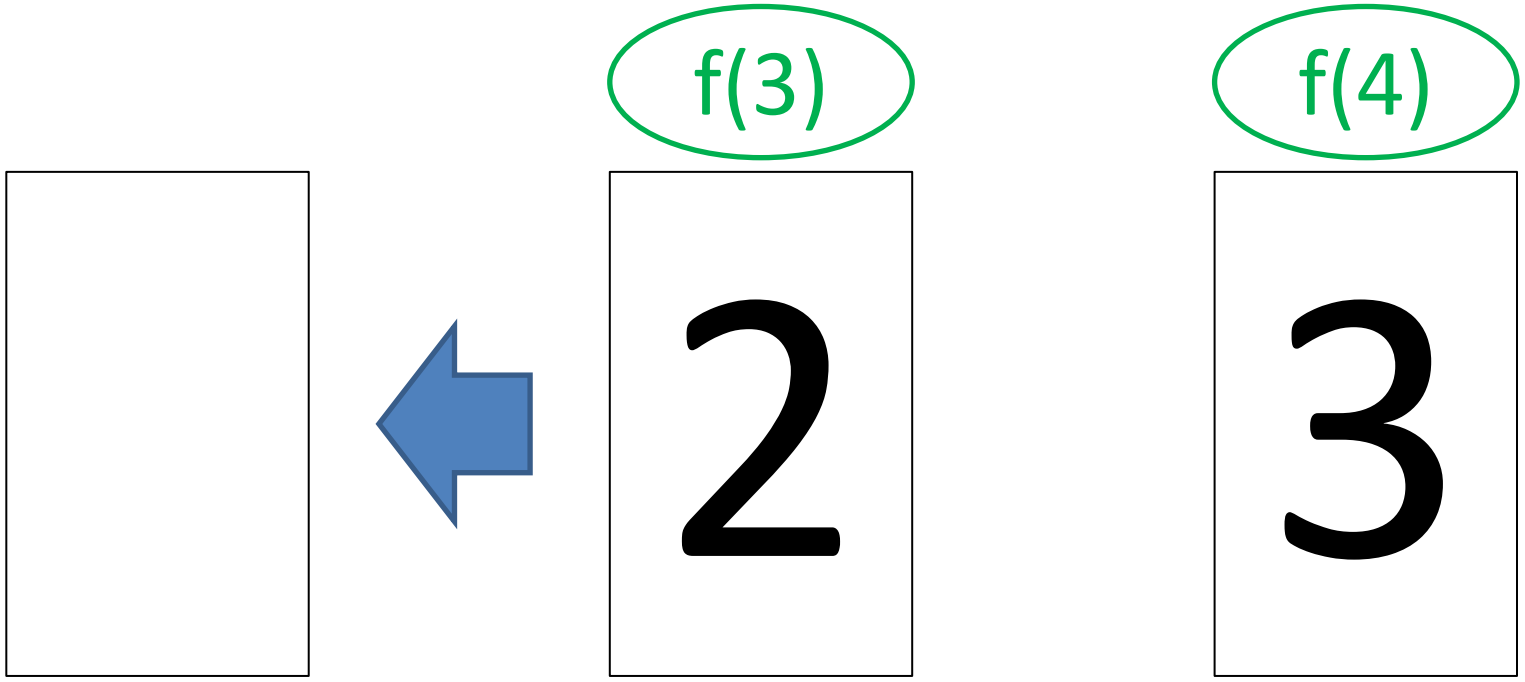
2

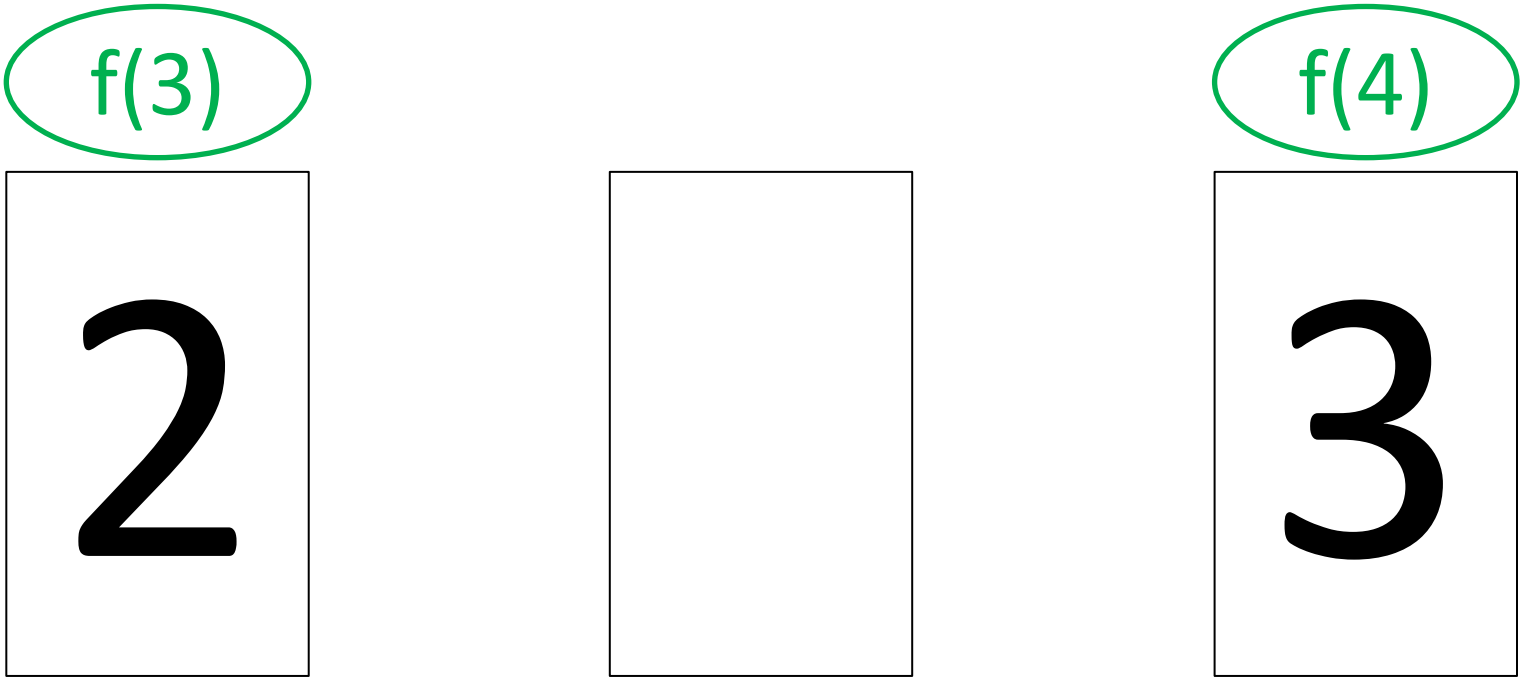
f(4)

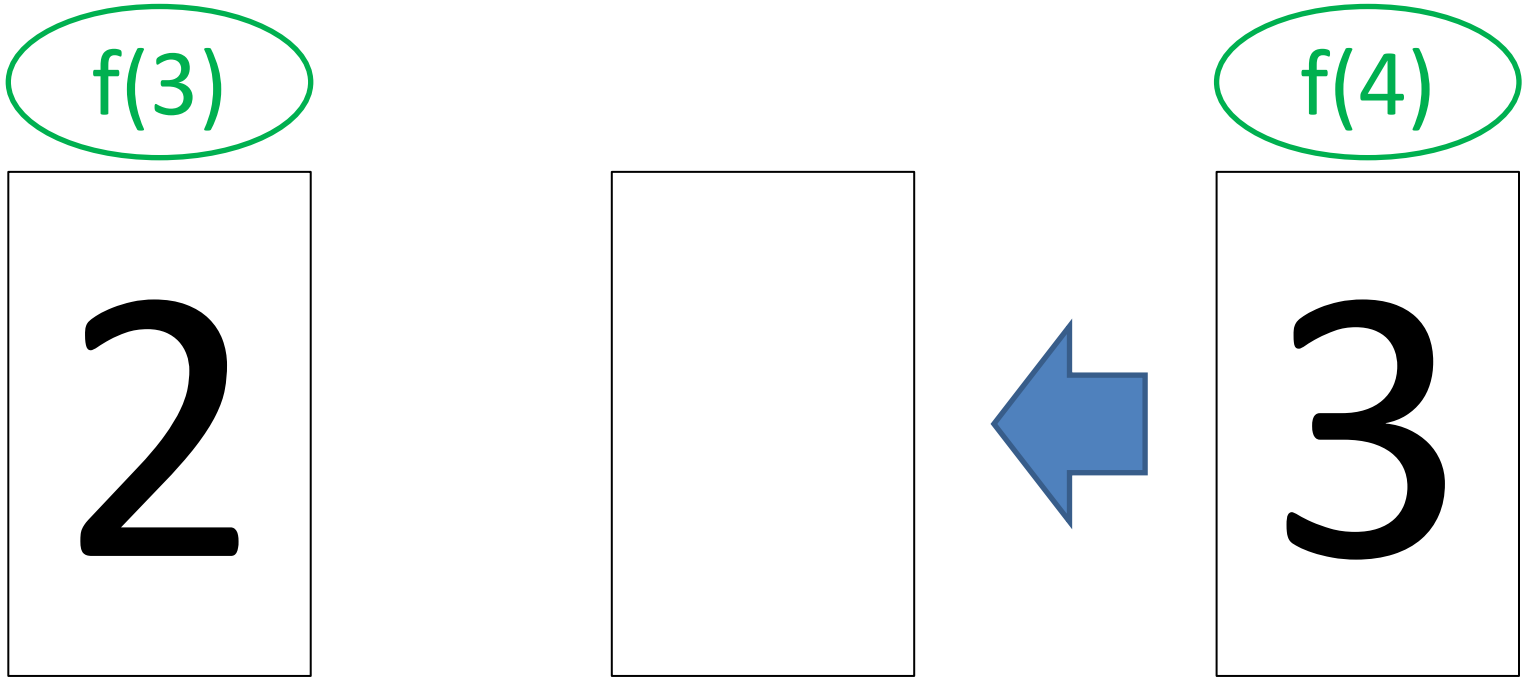
3

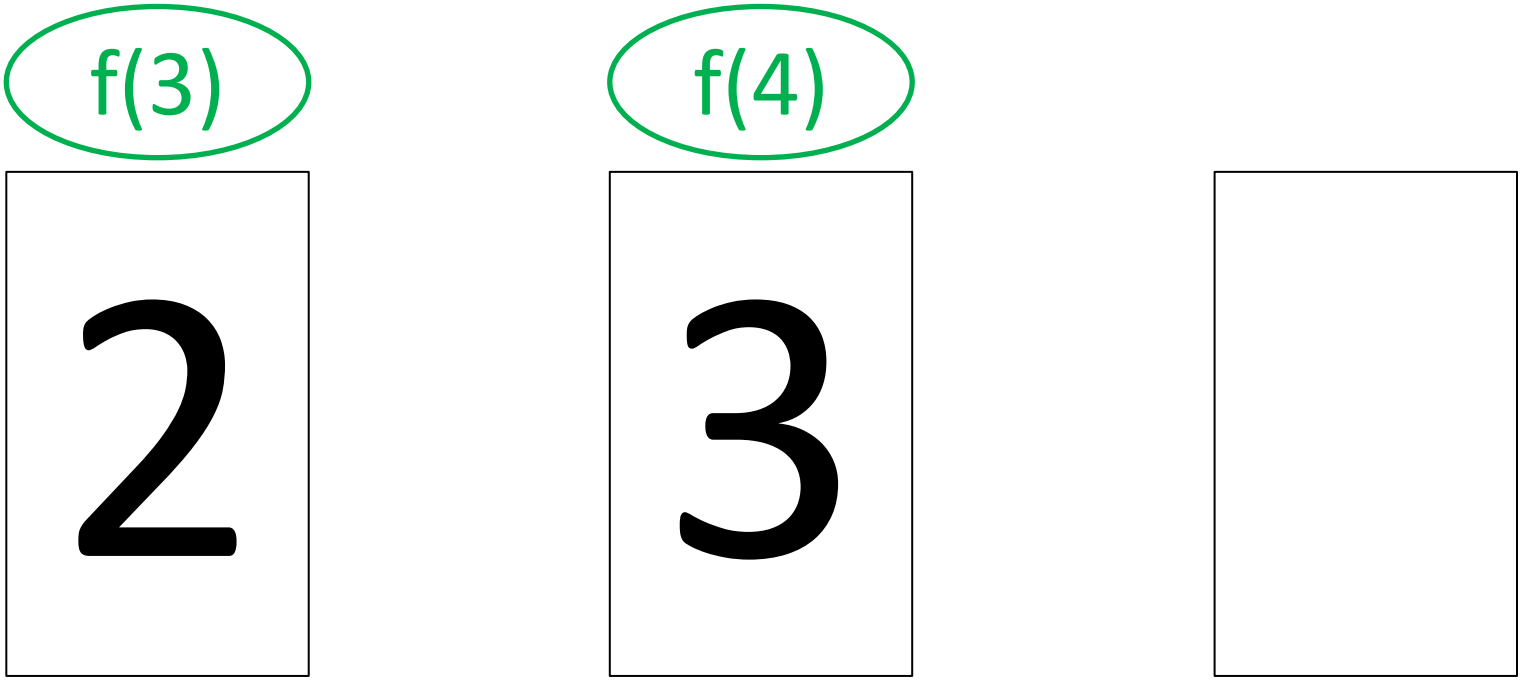


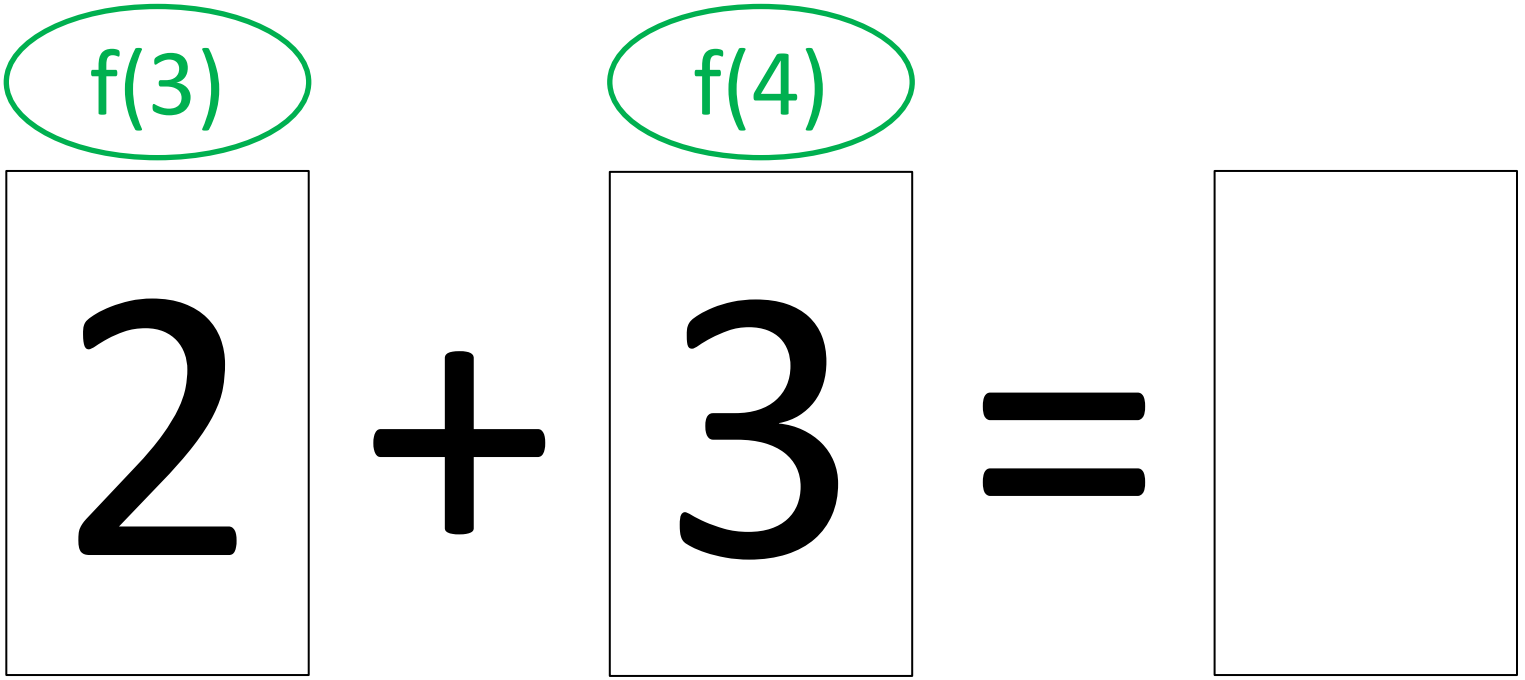

f(3)
f(4)
2
3

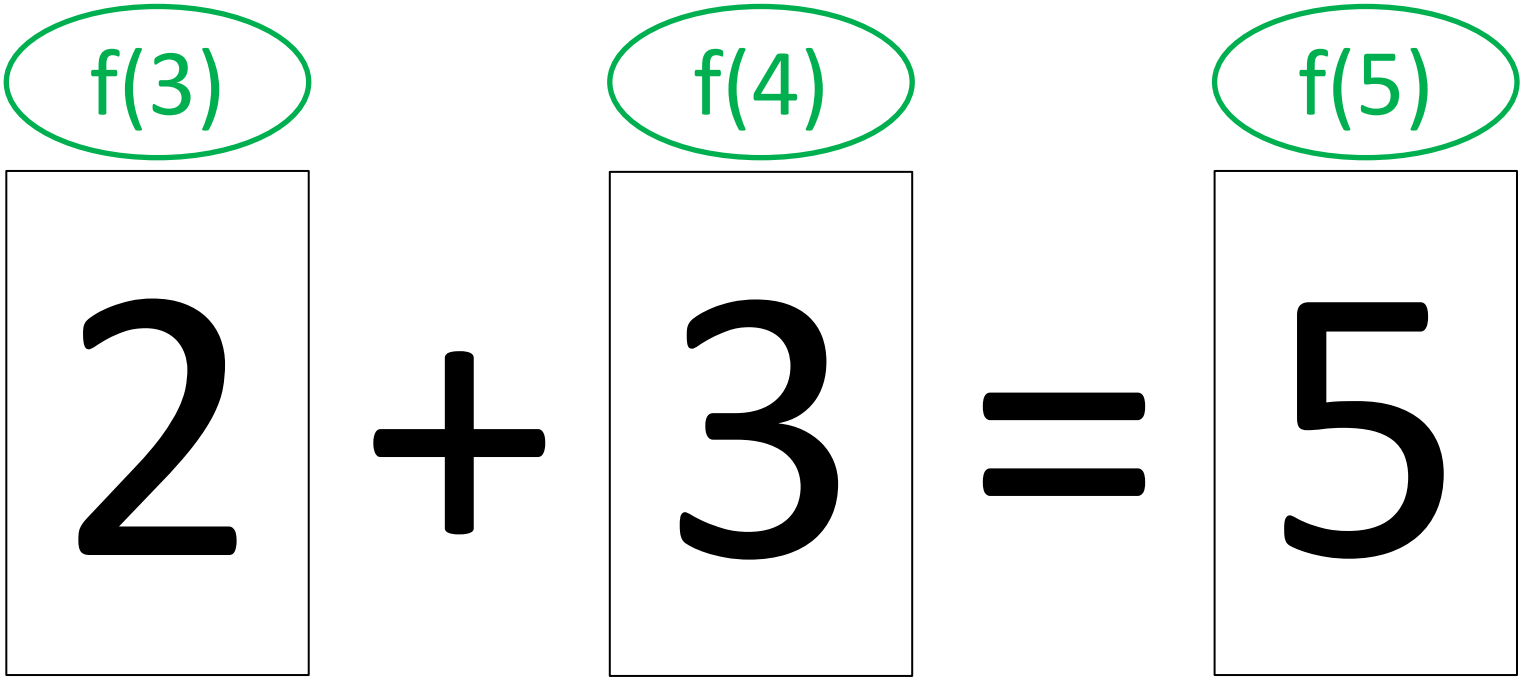

f(3)
f(4)
2
3

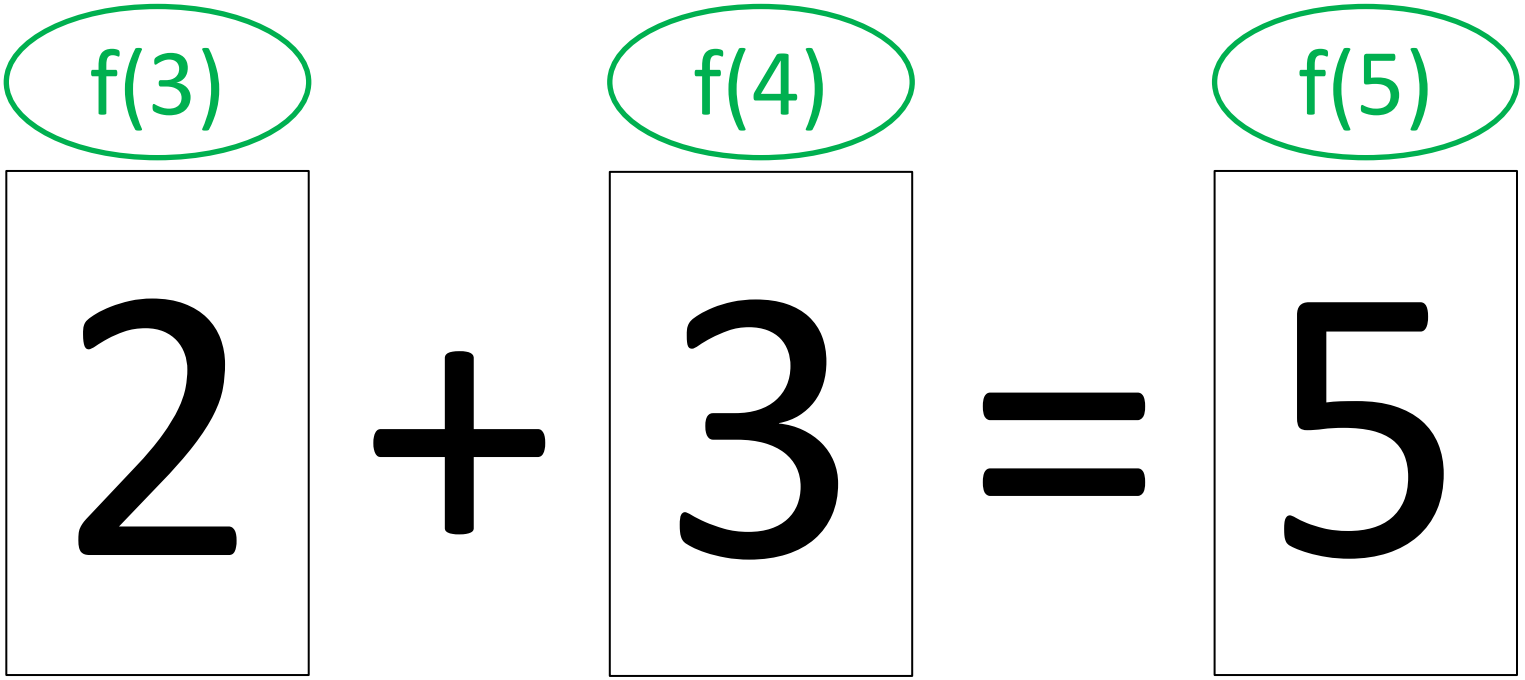

f(3)
2
f(4)
3


f(3)
2
f(4)
3


f(3)
f(4)
2
3


f(3)
f(4)
2
+
3
=


f(3)
f(4)
f(5)
2
+
3
=
5



usw.

```
fib := proc(n :: integer)
  local i, f, f1, f2;
  f1 := 0; f2 := 1; f := n;
  for i from 2 to n do
    f := f1 + f2;
    f1 := f2;
    f2 := f;
  end do;
  return f;
end proc :
```

Fibonacci („falsch“)

$f(0) = 0, \quad f(1) = 1, \quad f(n) = f(n-1) + f(n-2)$

```
fib := proc(n :: integer)
   local i, f, f1, f2;
   f1 := 0;  f2 := 1;  f := n;
   for i from 2 to n do
      f := f1 + f2;
      f1 := f2;
      f2 := f;
   end do;
   return f;
end proc :
```

```
fib := proc(n)
   local i, f, f1, f2;
   f1 := 0;  f2 := 1;
   for i to n do
      f := f1 + f2;
      f2 := f1;
      f1 := f;
   end do;
   return f;
end proc :
```

```
fib := proc(n :: integer)
  local i, f, f1, f2;
  f1 := 0;  f2 := 1;  f := n;
  for i from 2 to n do
    f := f1 + f2;
    f1 := f2;
    f2 := f;
  end do;
  return f;
end proc :
```

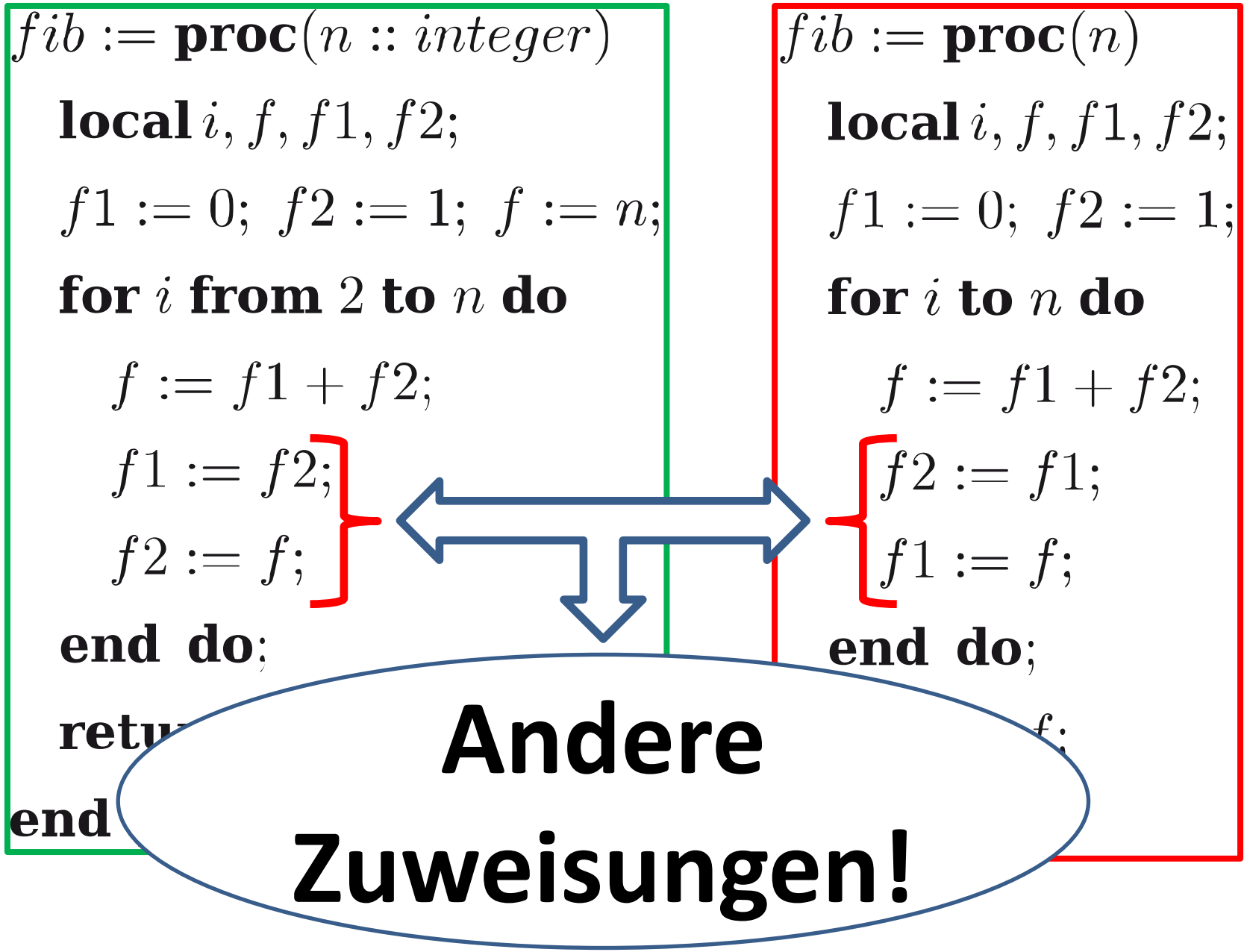
```
fib := proc(n)
  local i, f, f1, f2;
  f1 := 0;  f2 := 1;
  for i to n do
```

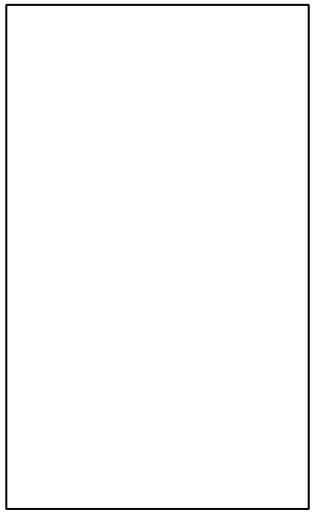
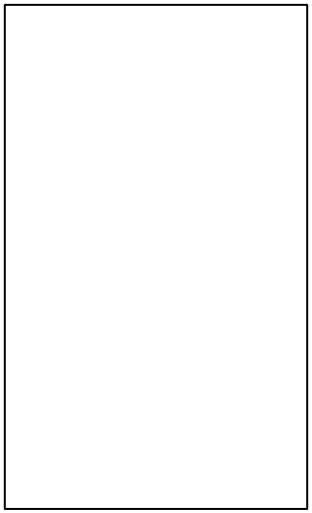
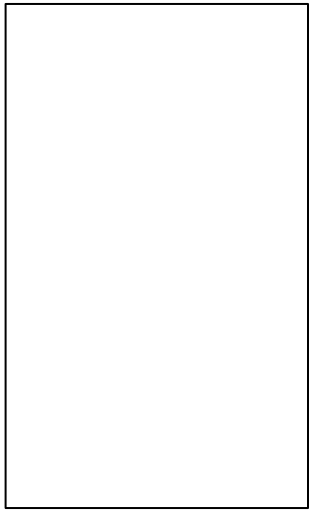
Spielt keine Rolle, bewirkt nur, dass die Funktion auch für n = 0 und n = 1 richtig arbeitet.

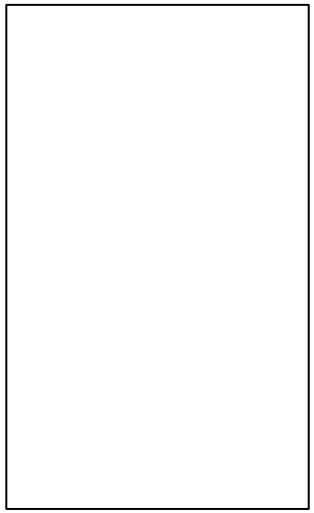
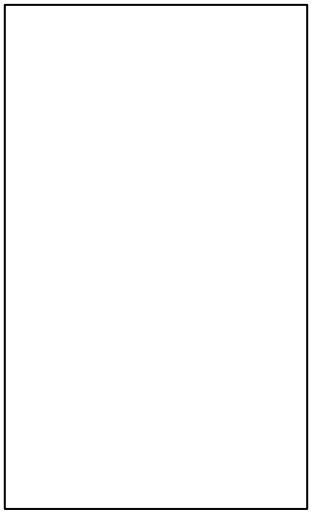
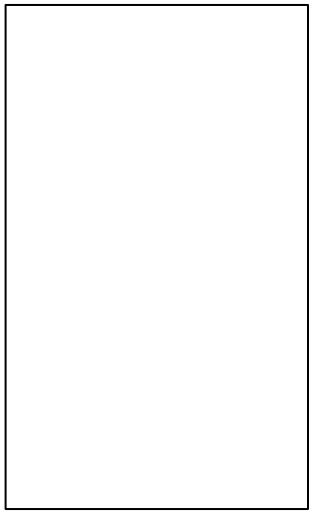
```
fib := proc(n :: integer)
  local i, f, f1, f2;
  f1 := 0; f2 := 1; f := n;
  for i from 2 to n do
    f := f1 + f2;
    f1 := f2;
    f2 := f;
  end do;
  return f;
end proc :
```

```
fib := proc(n)
  local i, f, f1, f2;
  f1 := 0; f2 := 1;
  for i to n do
    f := f1 + f2;
    f1;
```

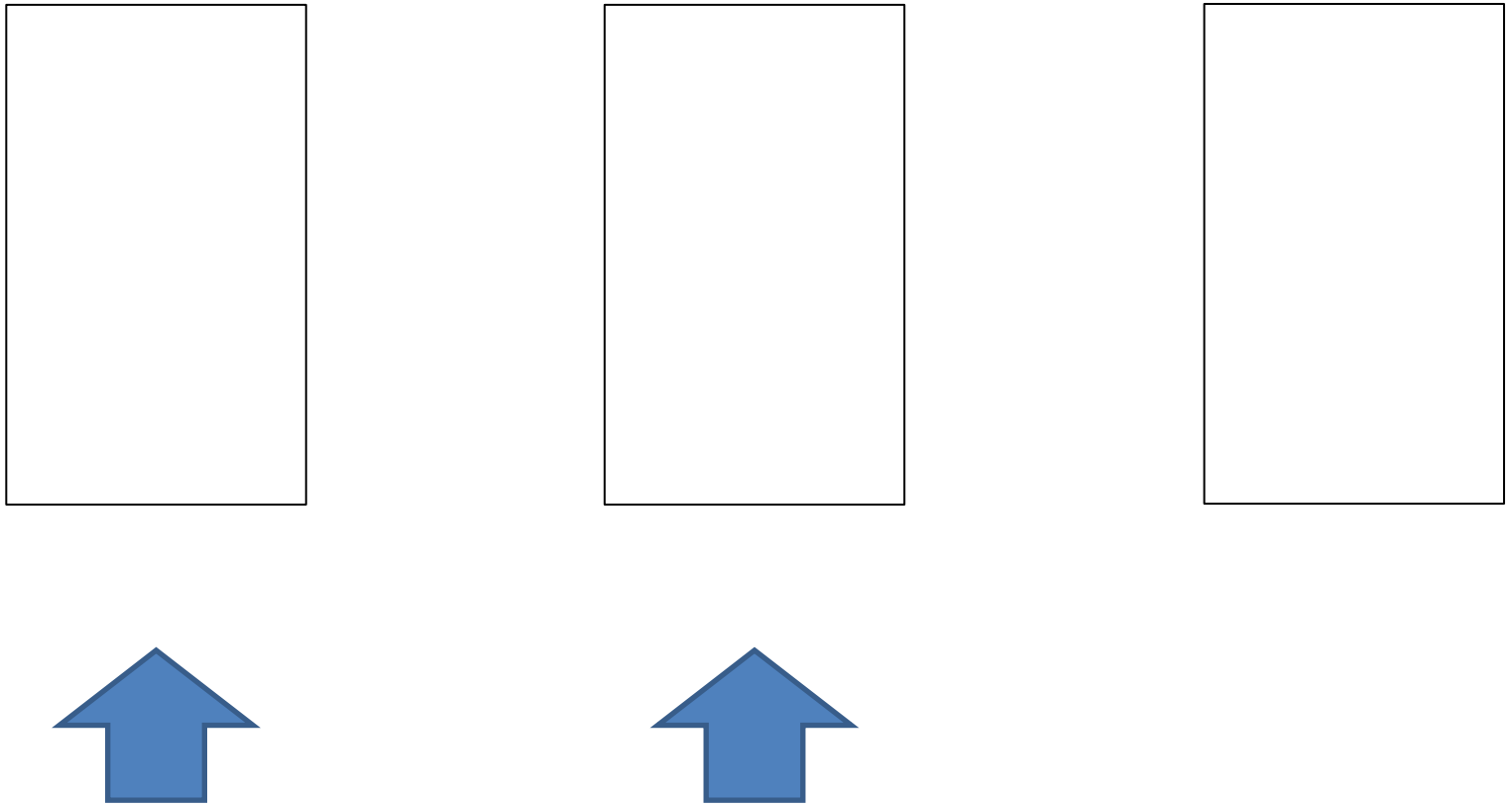
Anderer Schleifenbeginn!


fib := proc(n :: integer)
local i, f, f1, f2;
f1 := 0; f2 := 1; f := n;
for i from 2 to n do
f := f1 + f2;
f1 := f2;
f2 := f;
end do;
retu
end
fib := proc(n)
local i, f, f1, f2;
f1 := 0; f2 := 1;
for i to n do
f := f1 + f2;
f2 := f1;
f1 := f;
end do;
f;
Andere
Zuweisungen!

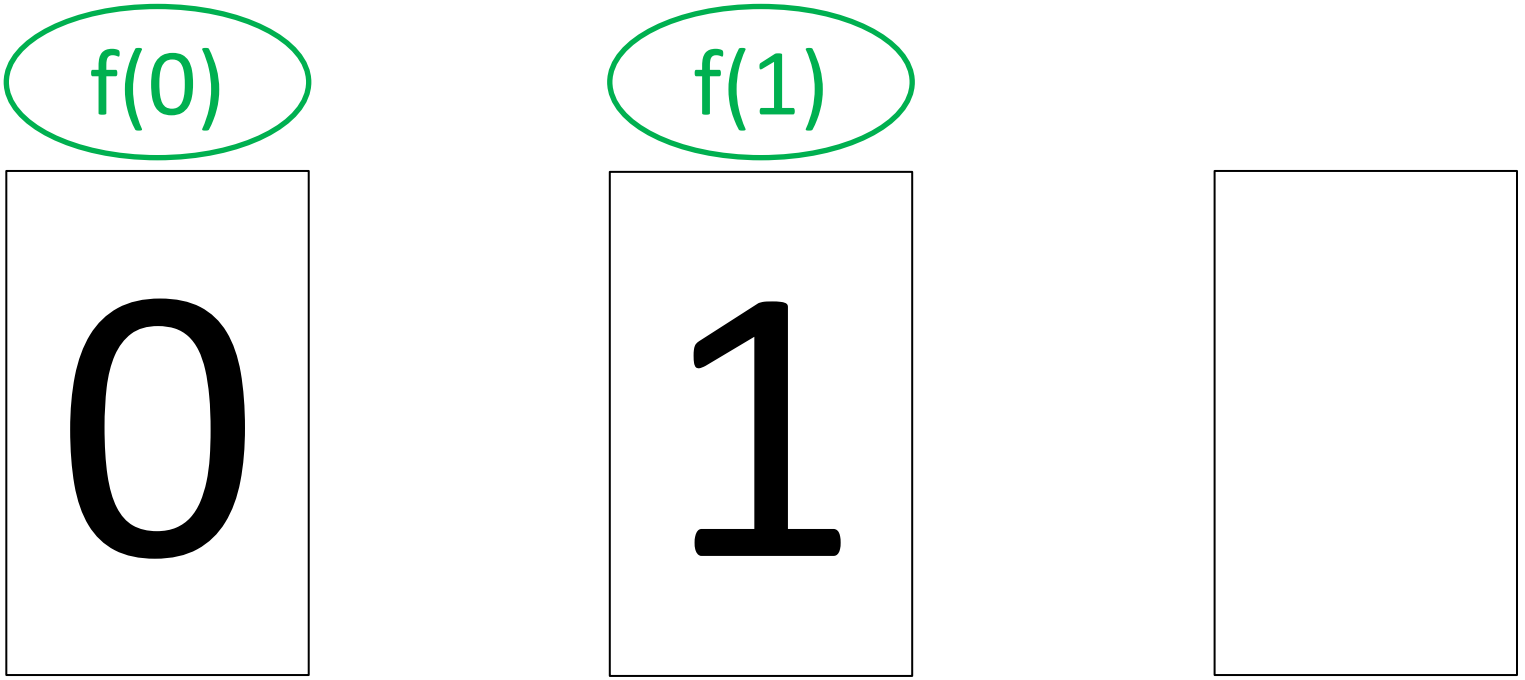




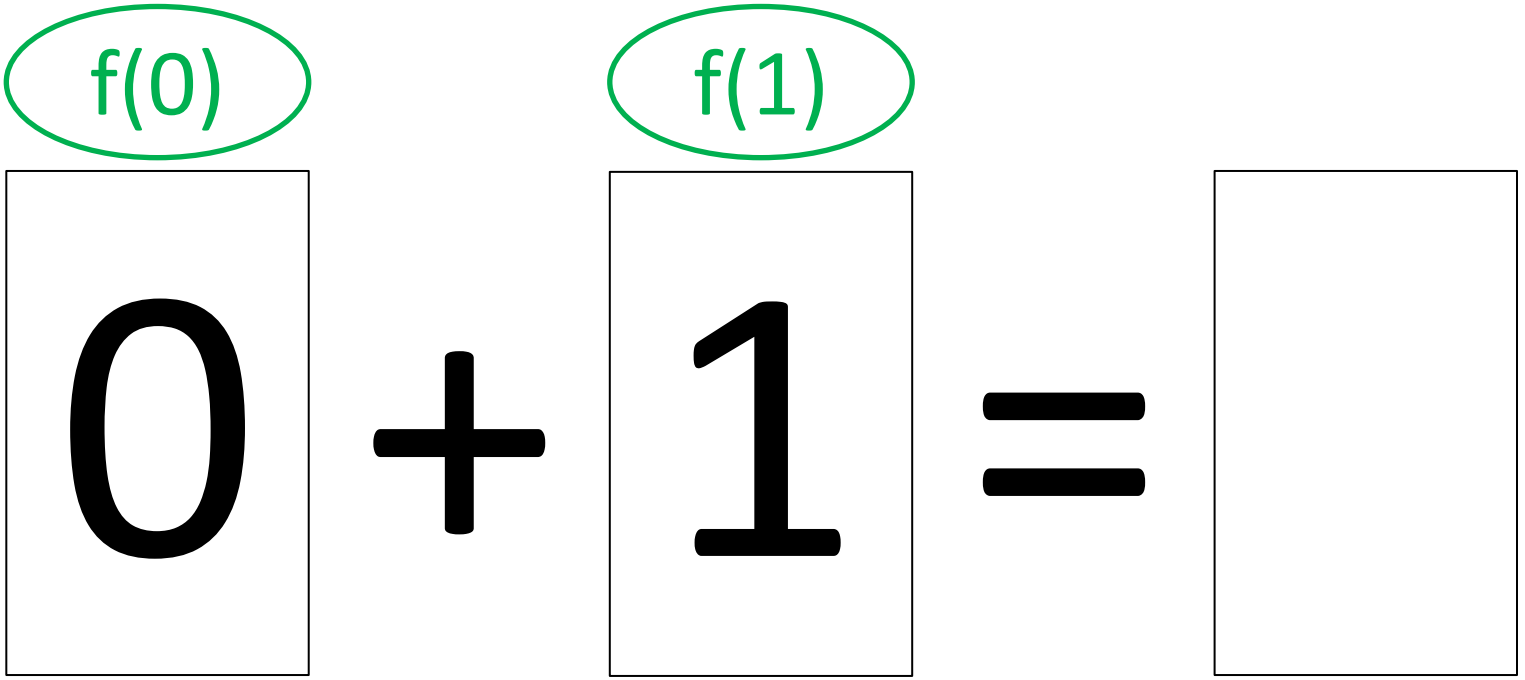




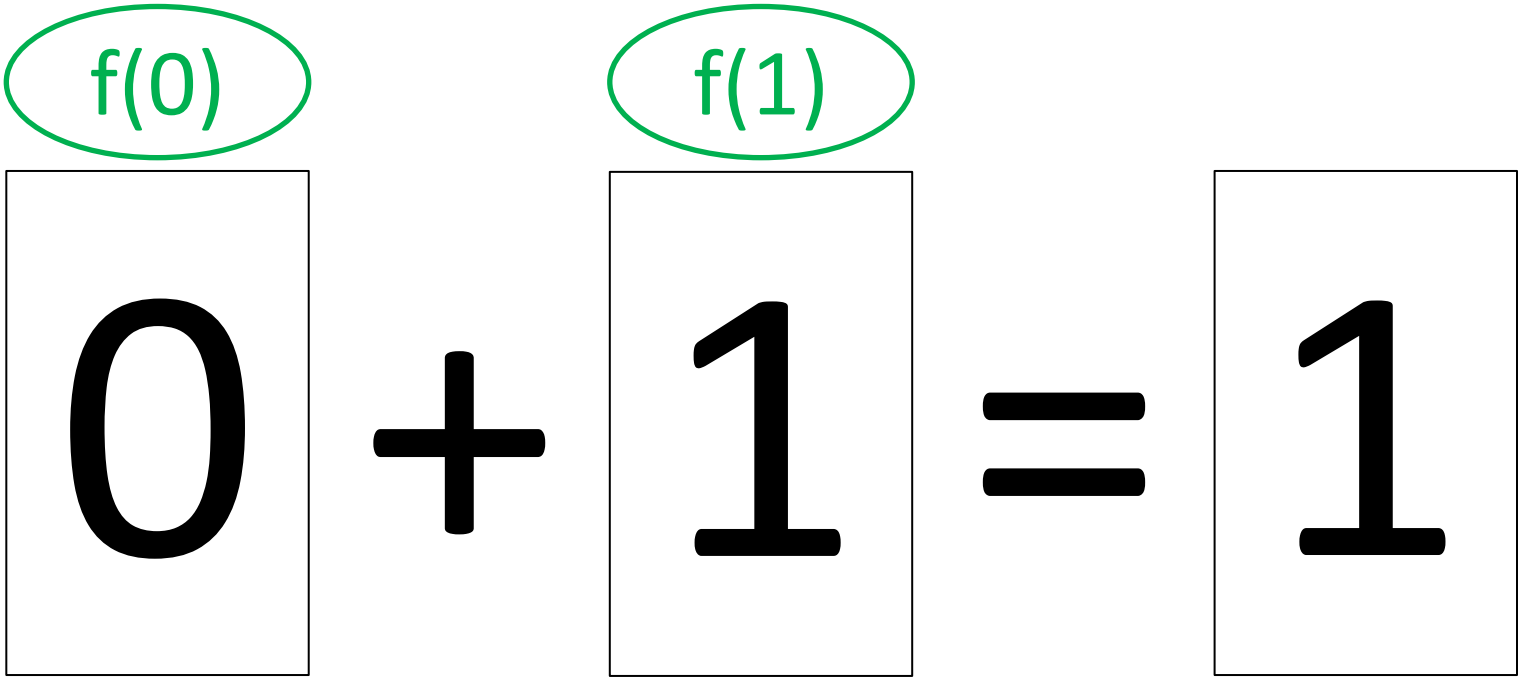
f(0) = 0 f(1) = 1

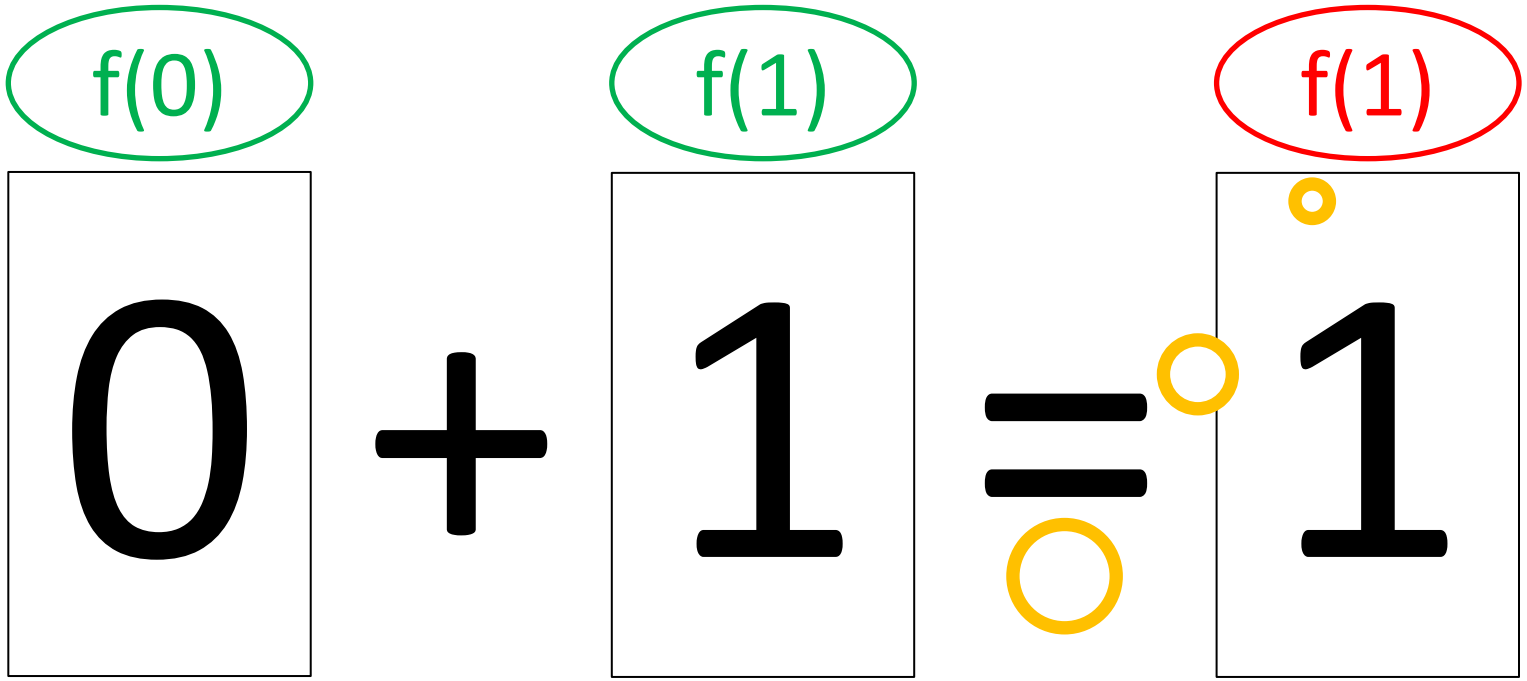


f(0) = 0 f(1) = 1

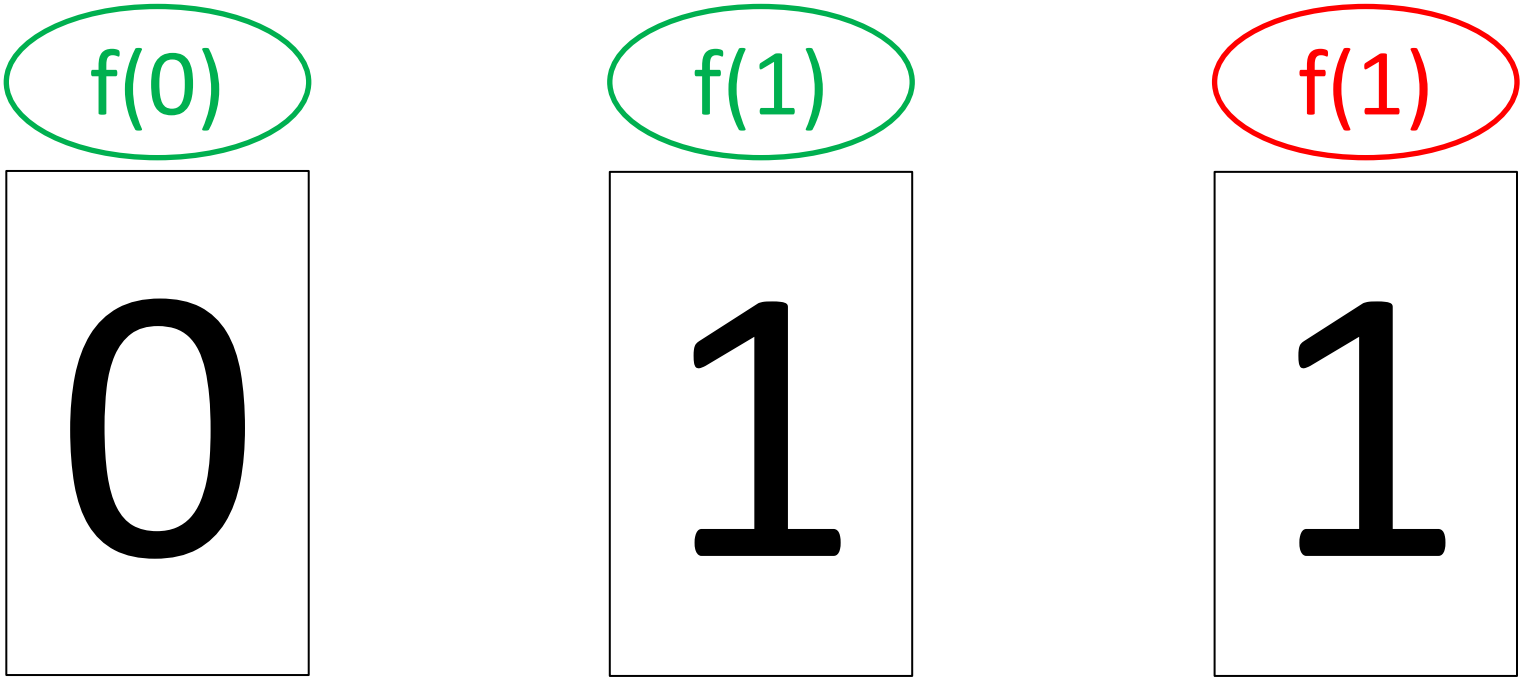

f(0)
f(1)
0
1


f(0)
f(1)
0
+
1
=


f(0)
f(1)
0
+
1
=
1



Schleife beginnt
bei n = 1!!!

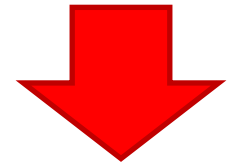

f(0)
0
f(1)
1
f(1)
1

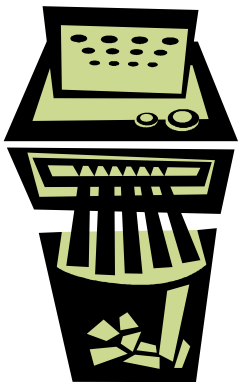
f(0)

0

f(1)

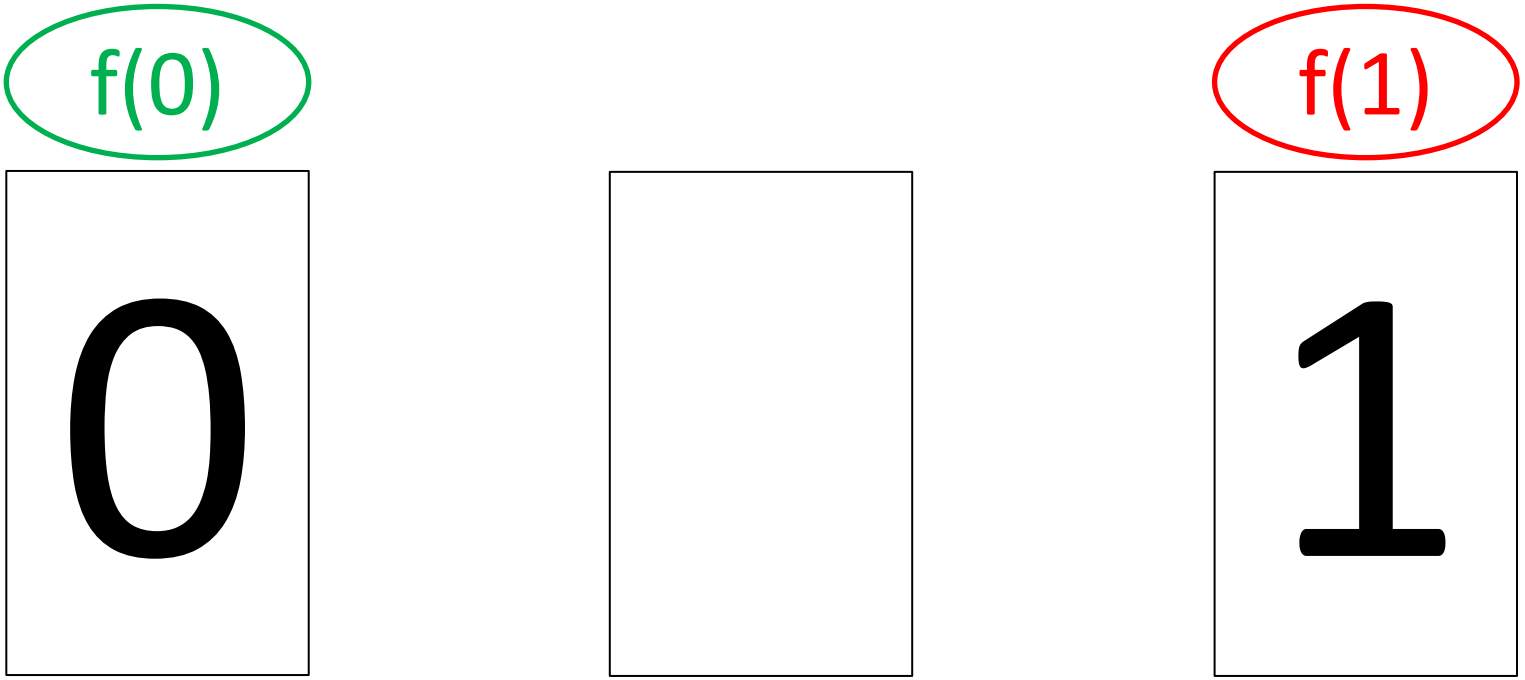
1

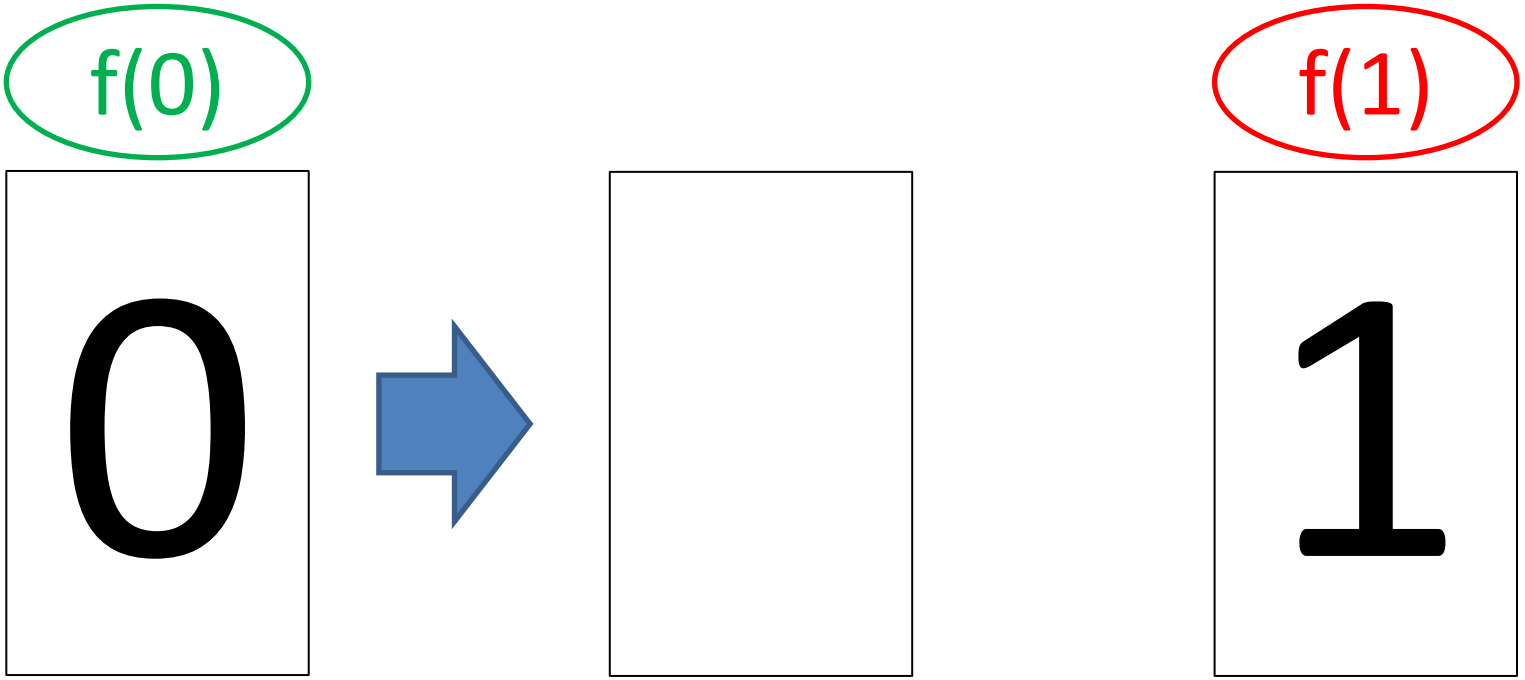


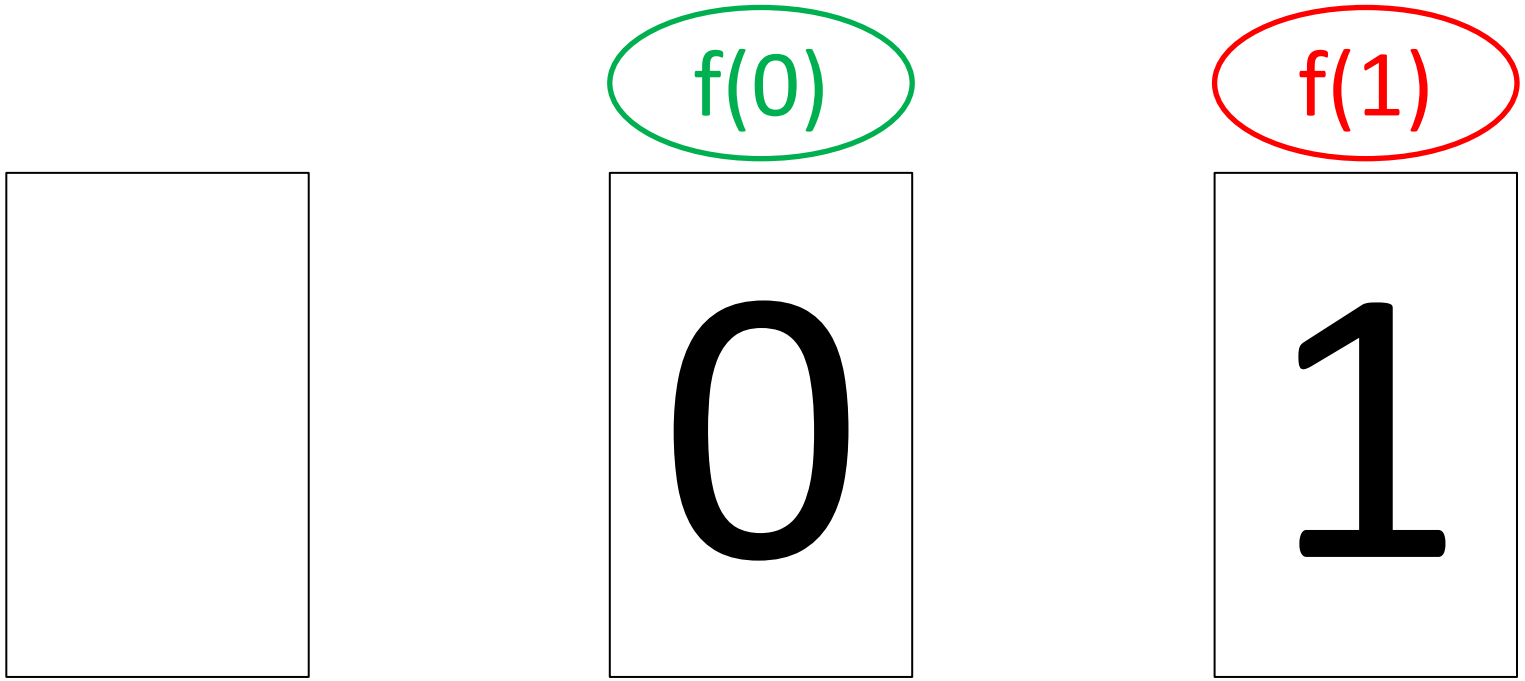


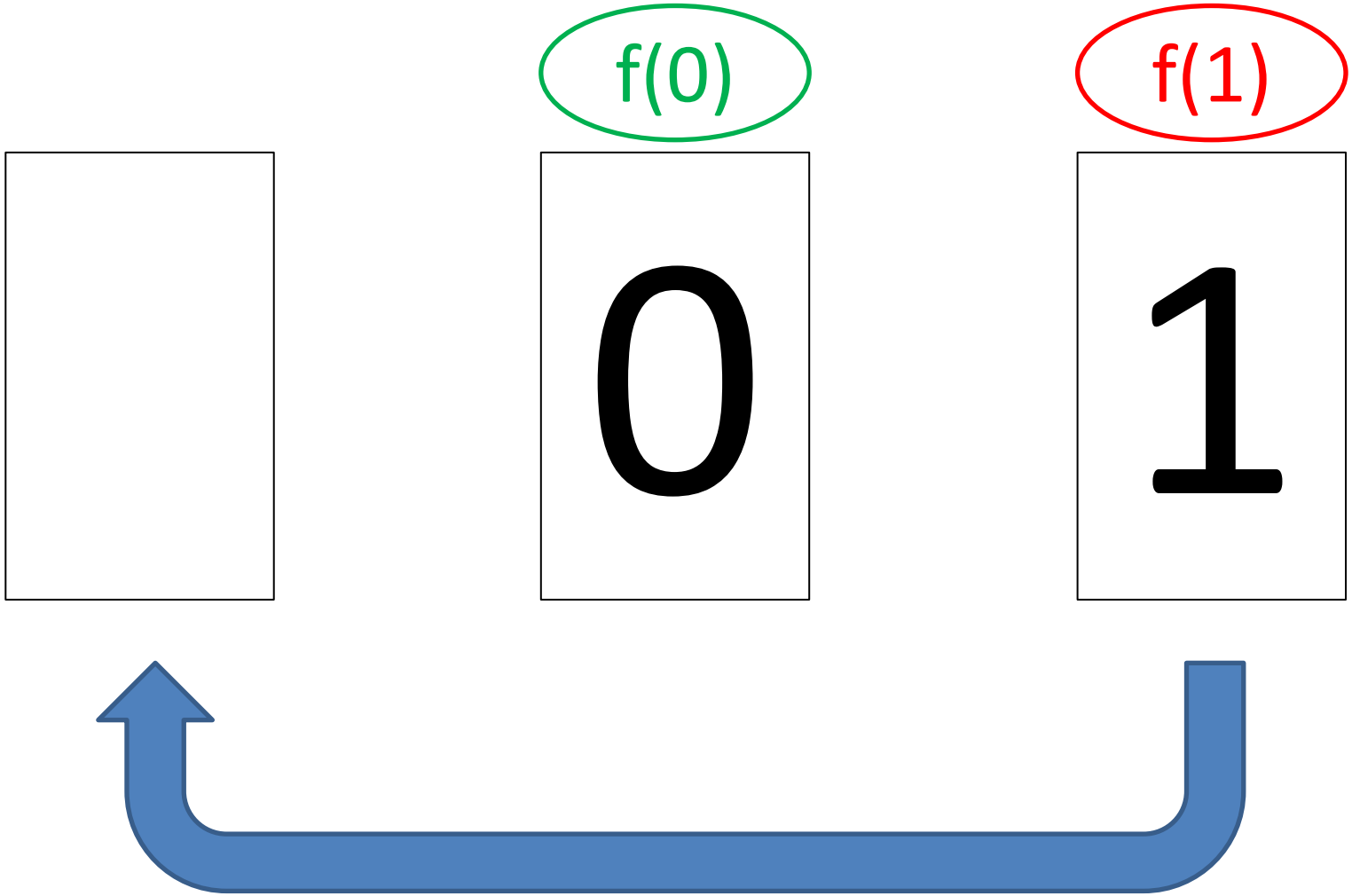
f(1)

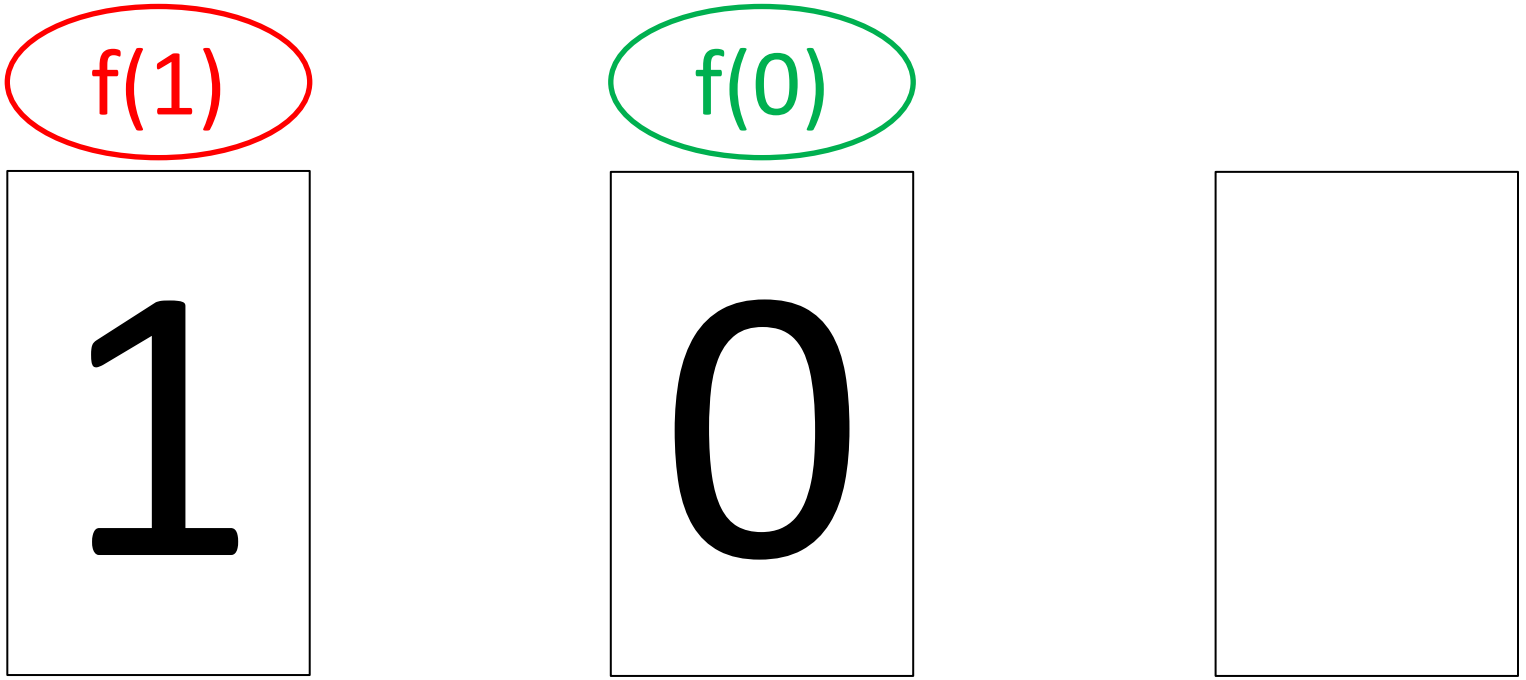
1

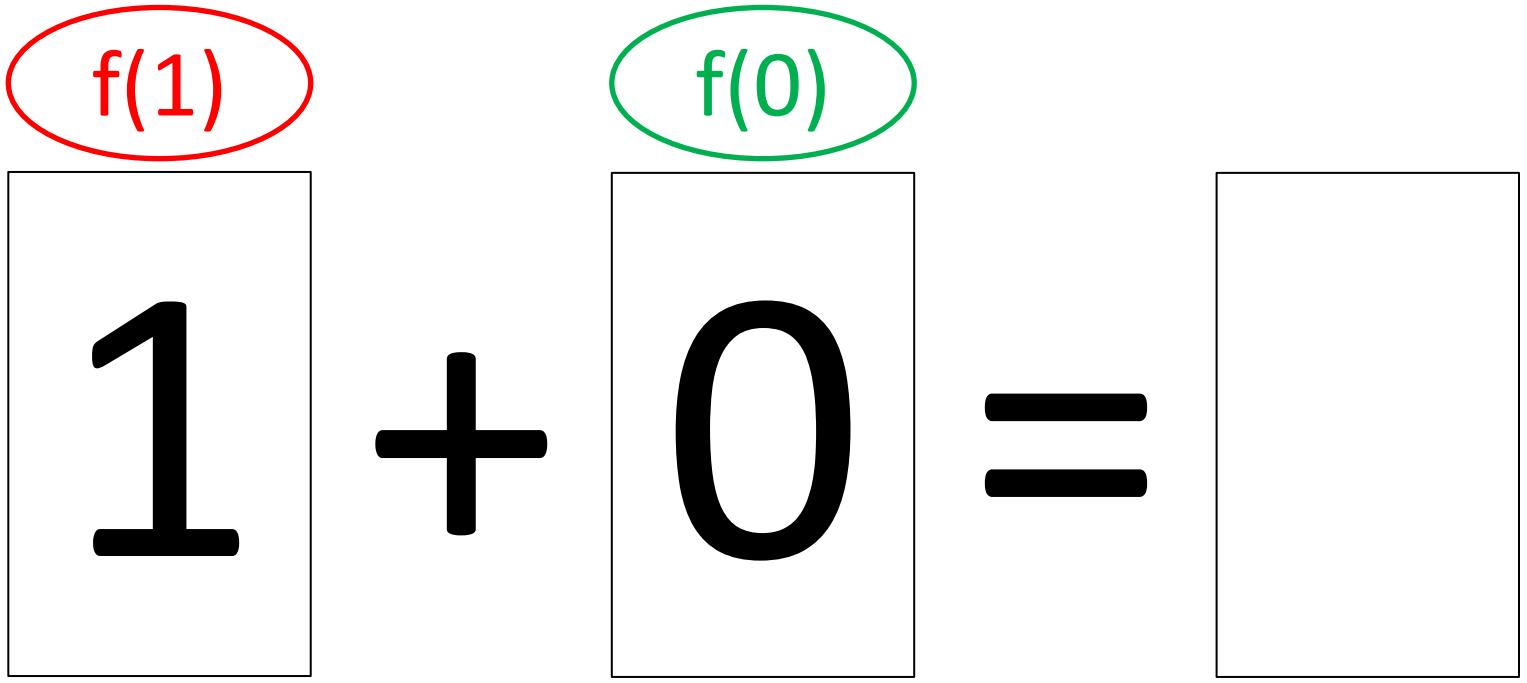

f(0)
0
f(1)
1

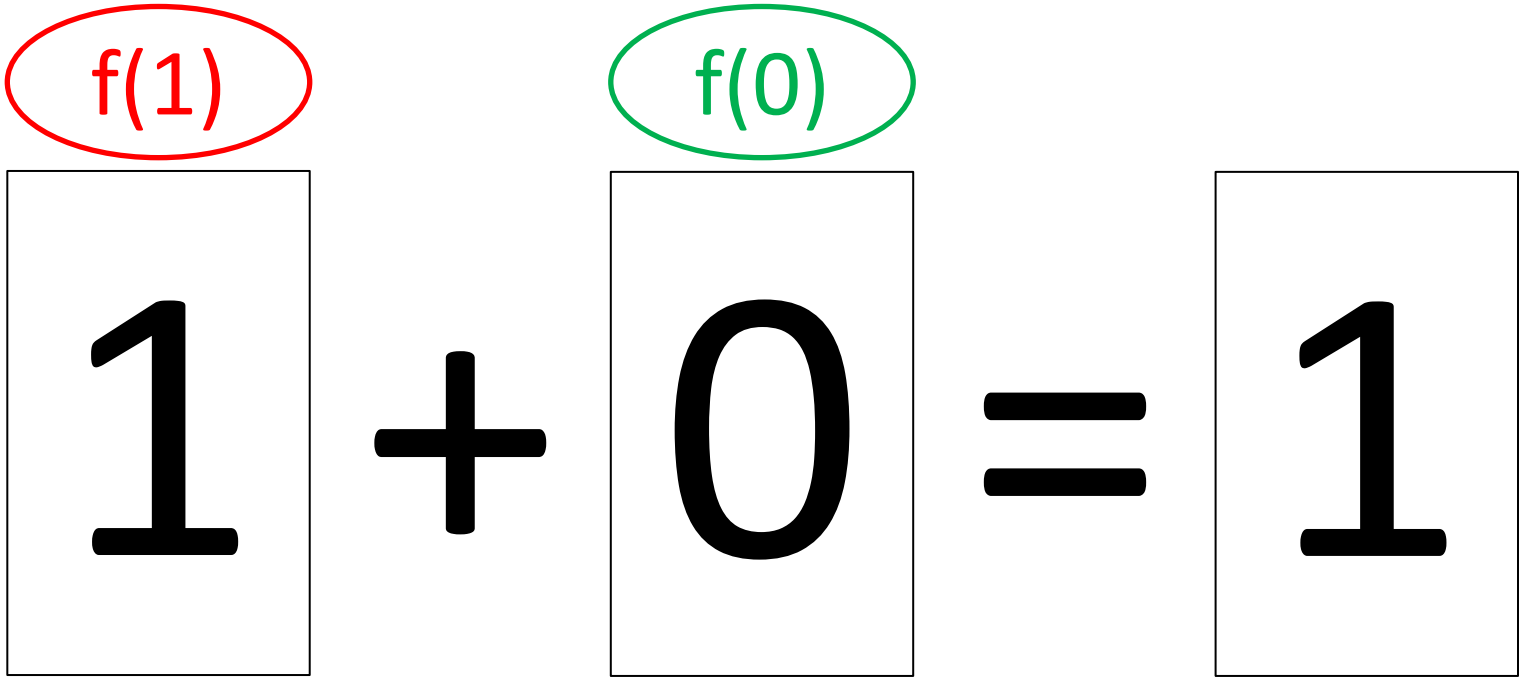

f(0)
0
f(1)
1

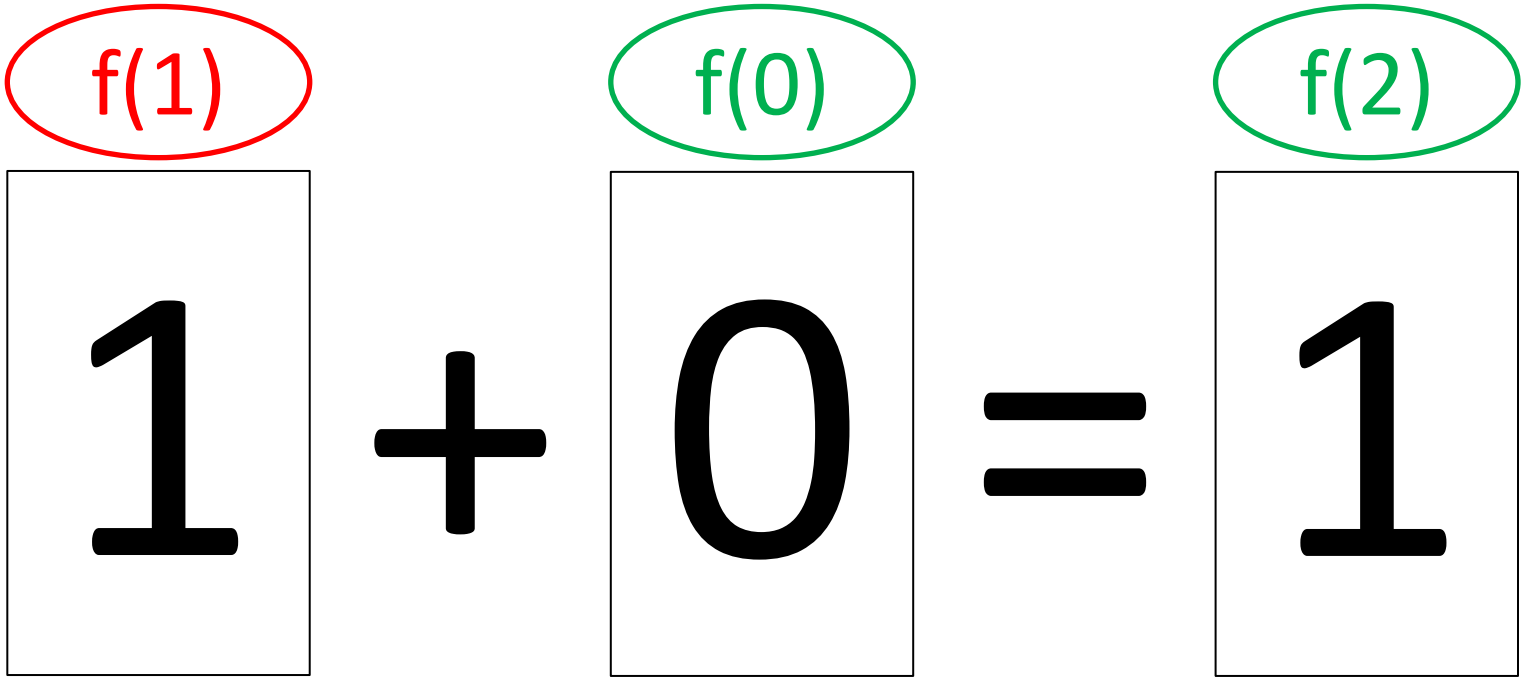

f(0)
f(1)
0
1

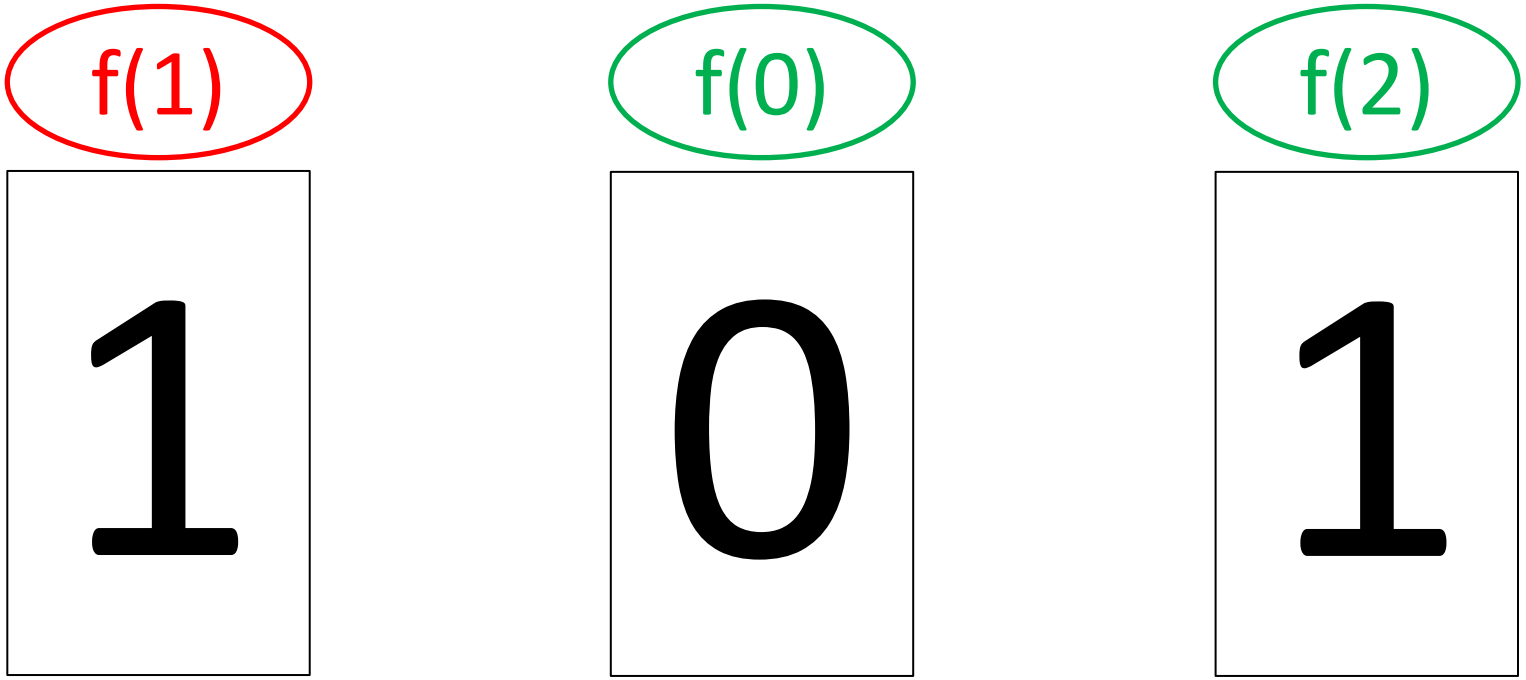

f(0)
f(1)
0
1


f(1)
f(0)
1
0


f(1)
f(0)
1
+
0
=


f(1)
f(0)
1
+
0
=
1


f(1)
f(0)
f(2)
1
+
0
=
1


f(1)
1
f(0)
0
f(2)
1

f(1)

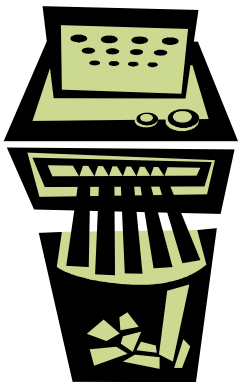
1

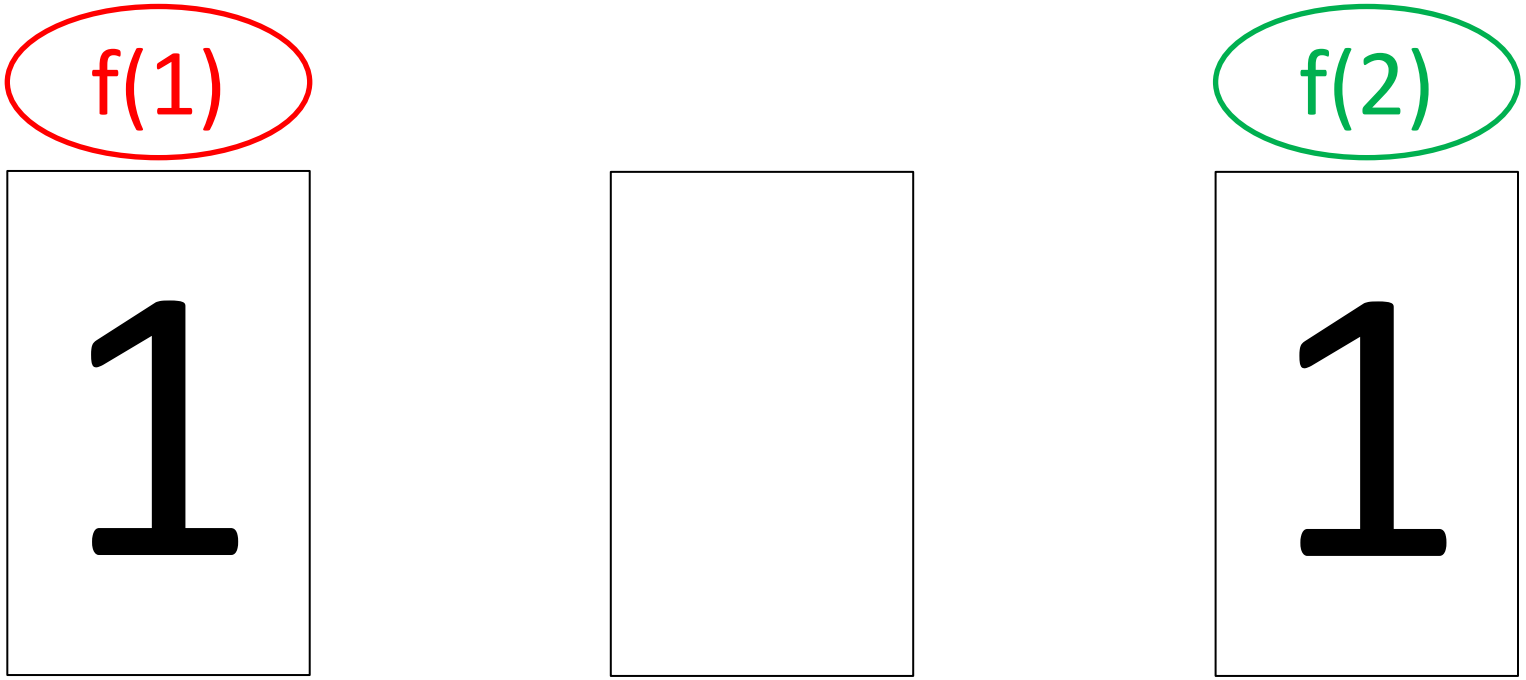
f(0)

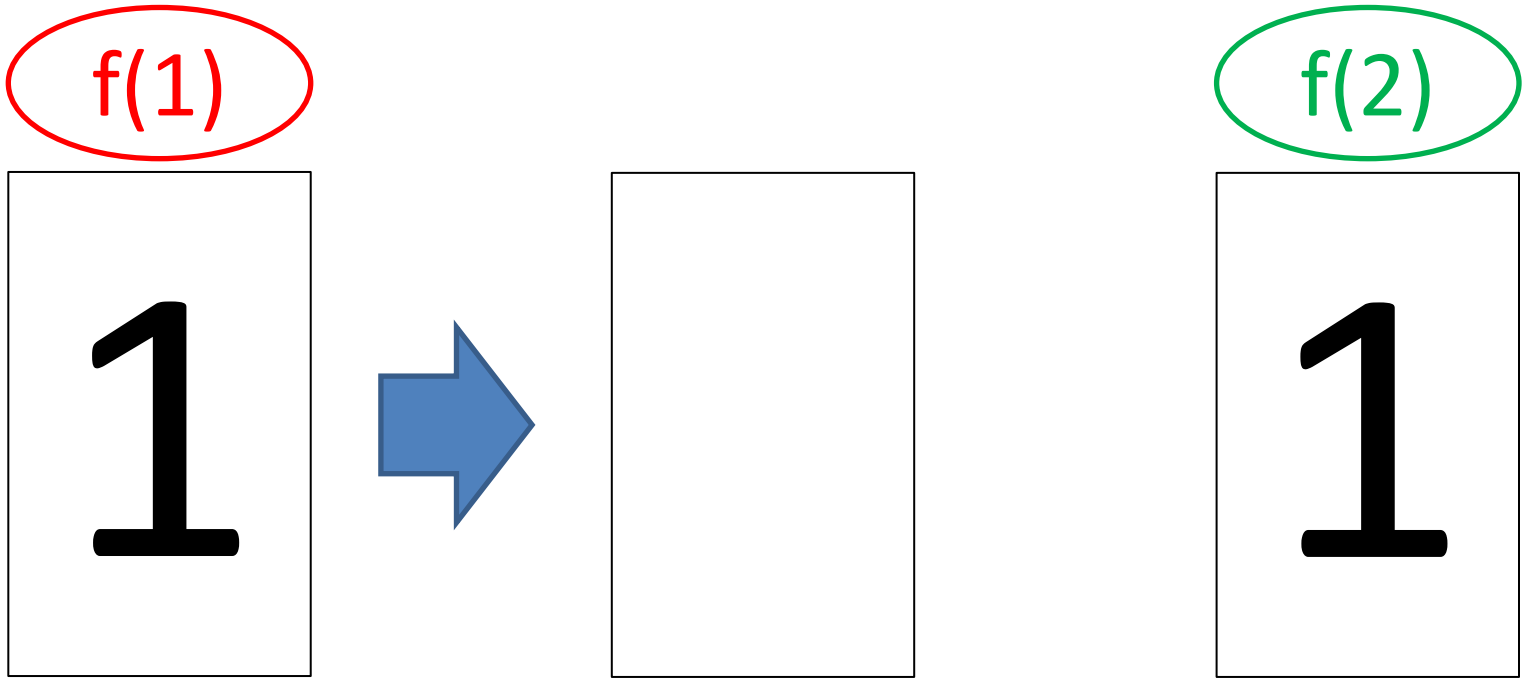
0

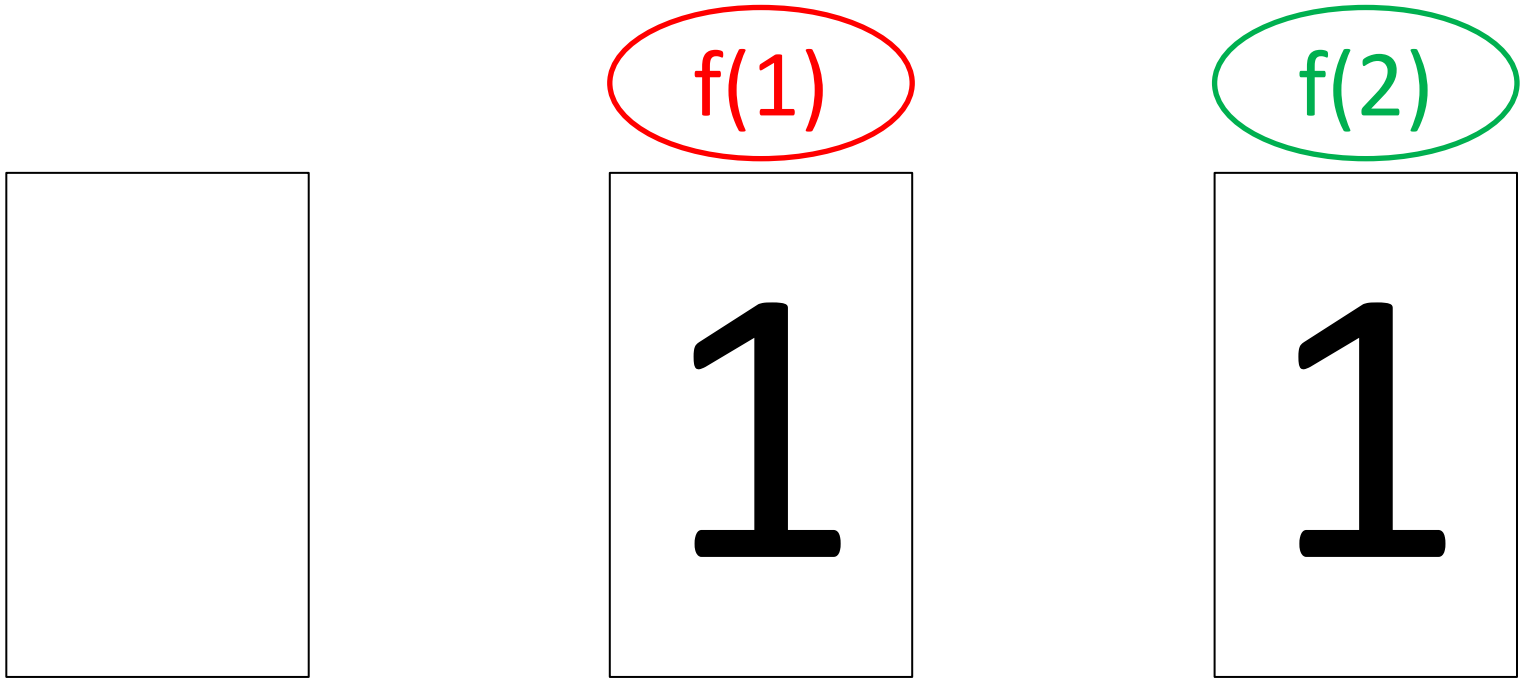
f(2)

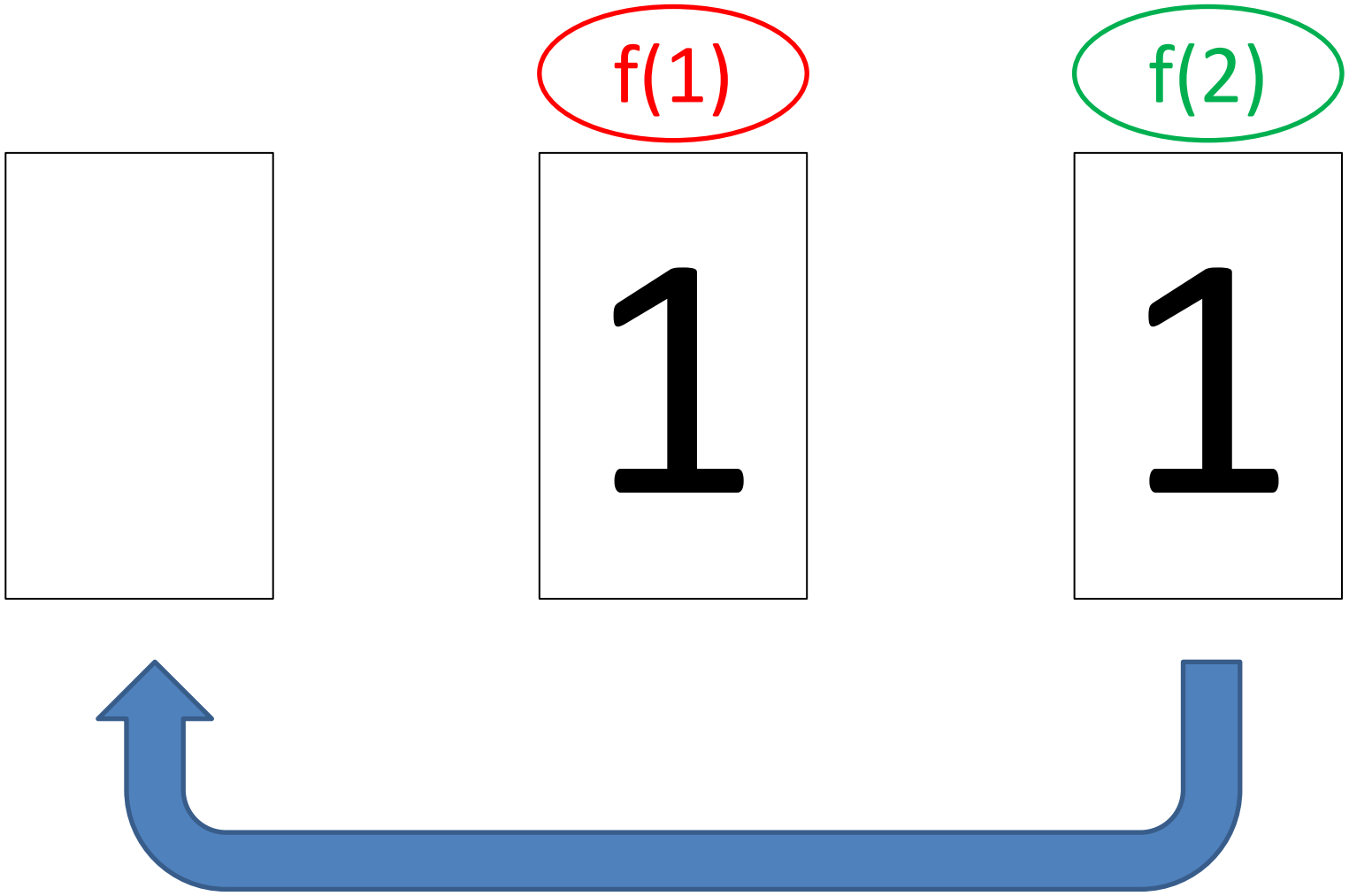
1

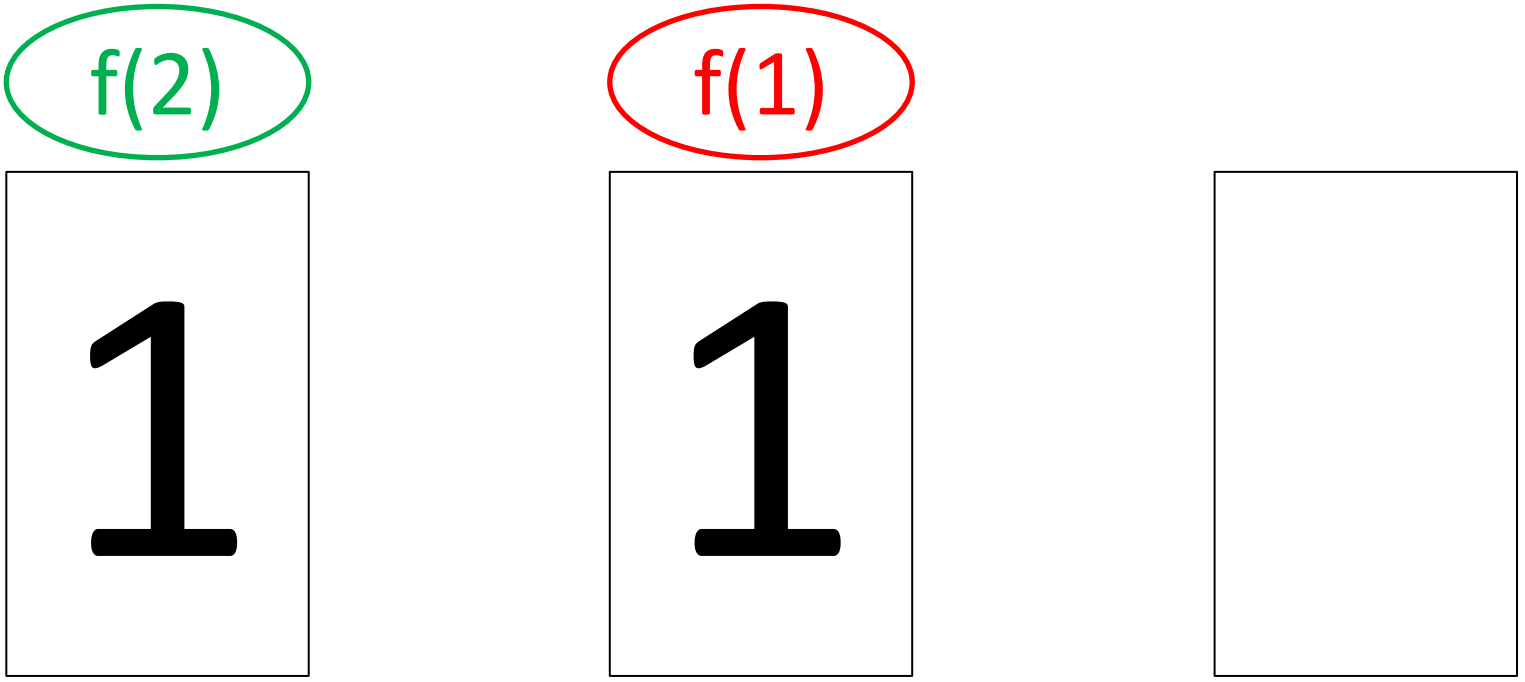


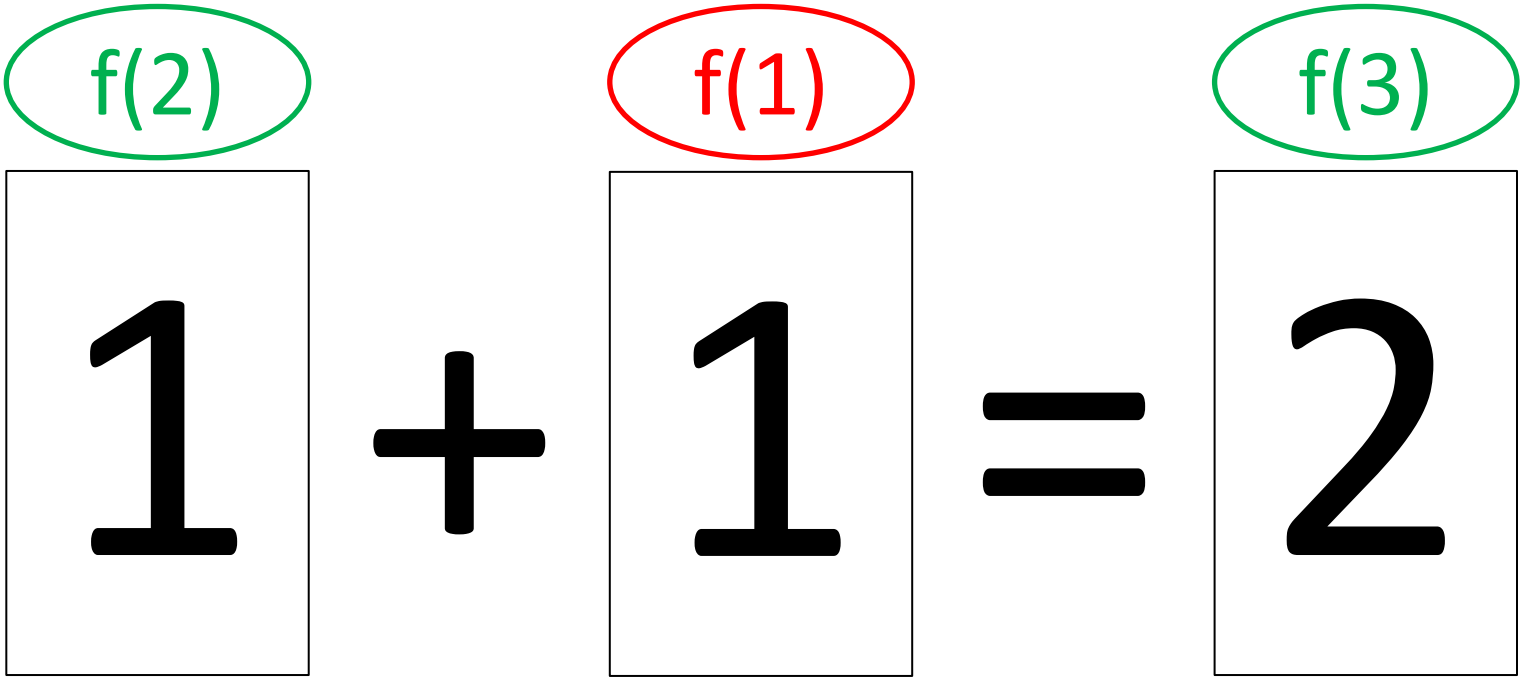

f(1)
1
f(2)
1

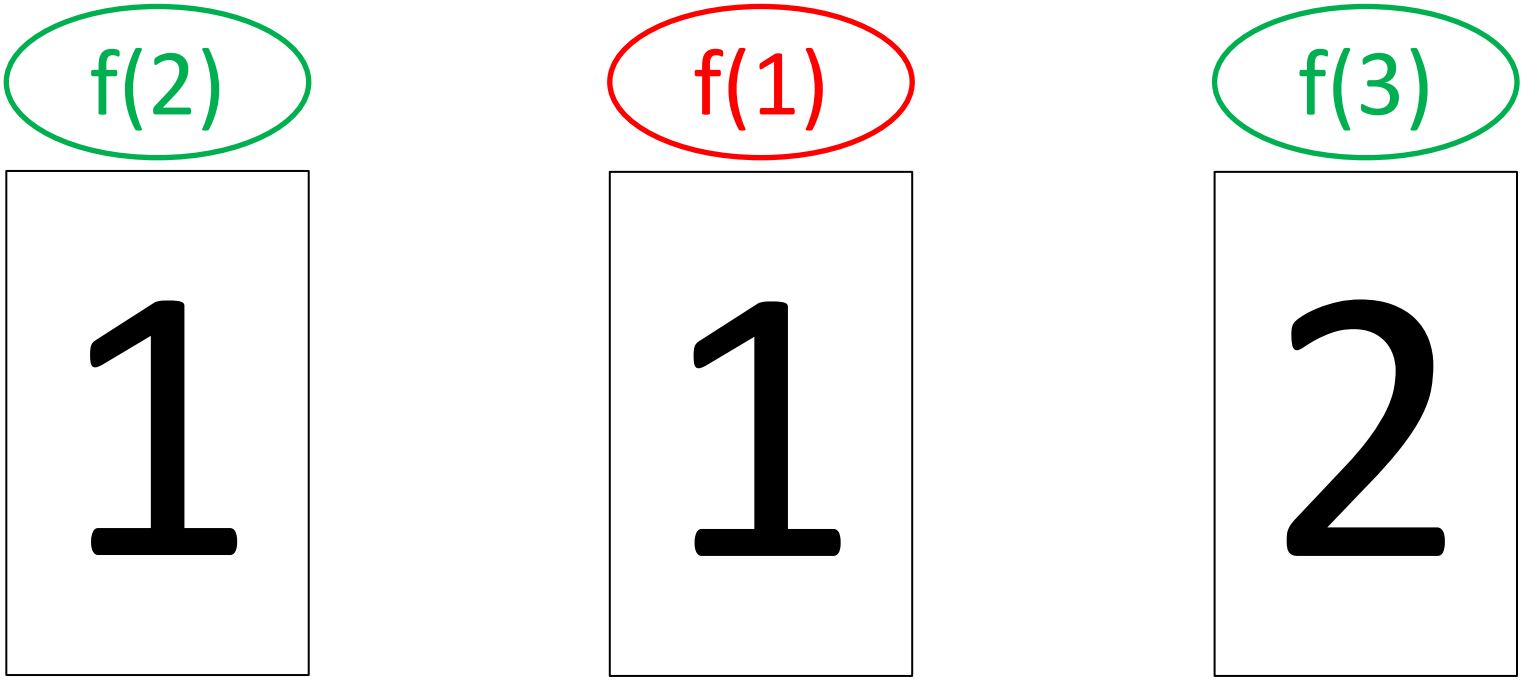

f(1)
1
f(2)
1


f(1)
f(2)
1
1


f(1)
f(2)
1
1


f(2)
1
f(1)
1


f(2)
f(1)
f(3)
1
+
1
=
2


f(2)
1
f(1)
1
f(3)
2

f(2)

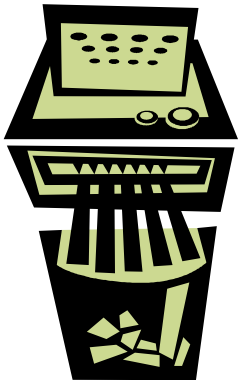
1

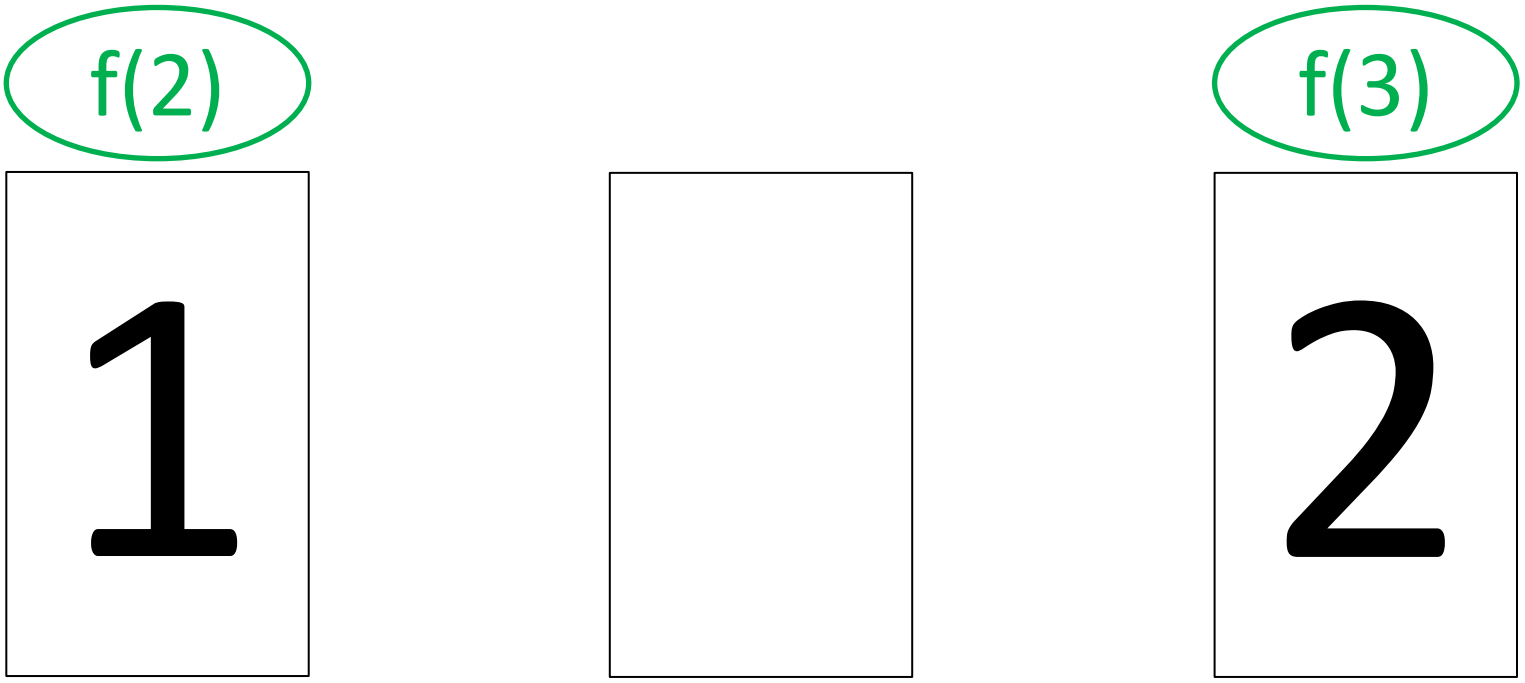
f(1)

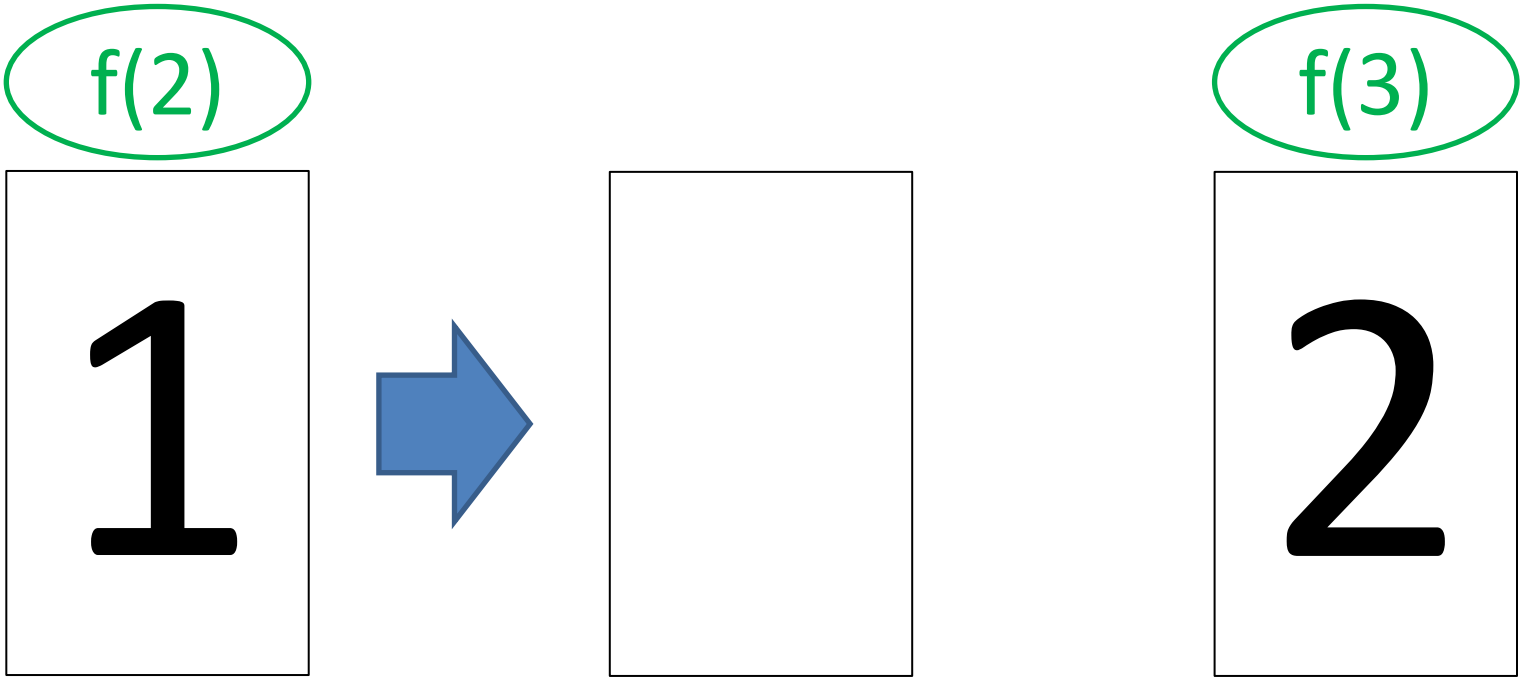
1

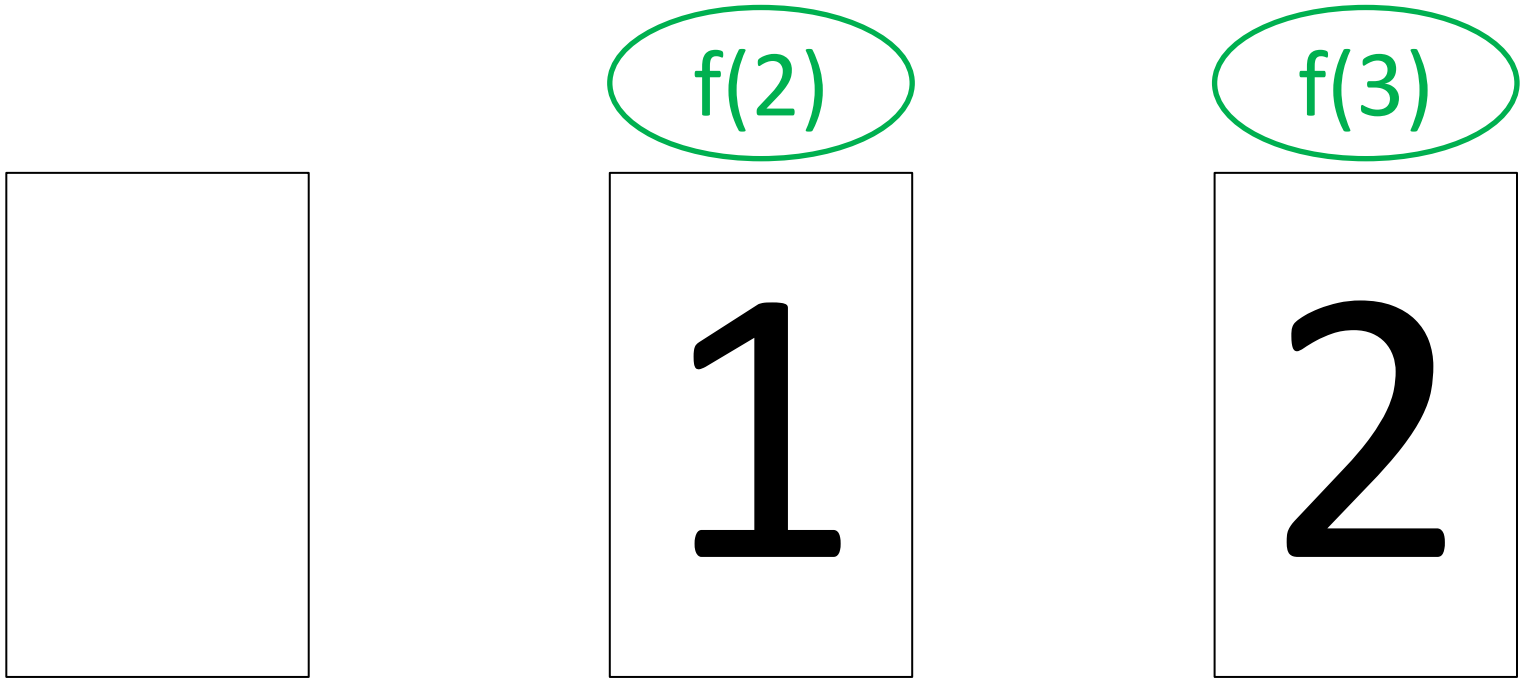
f(3)

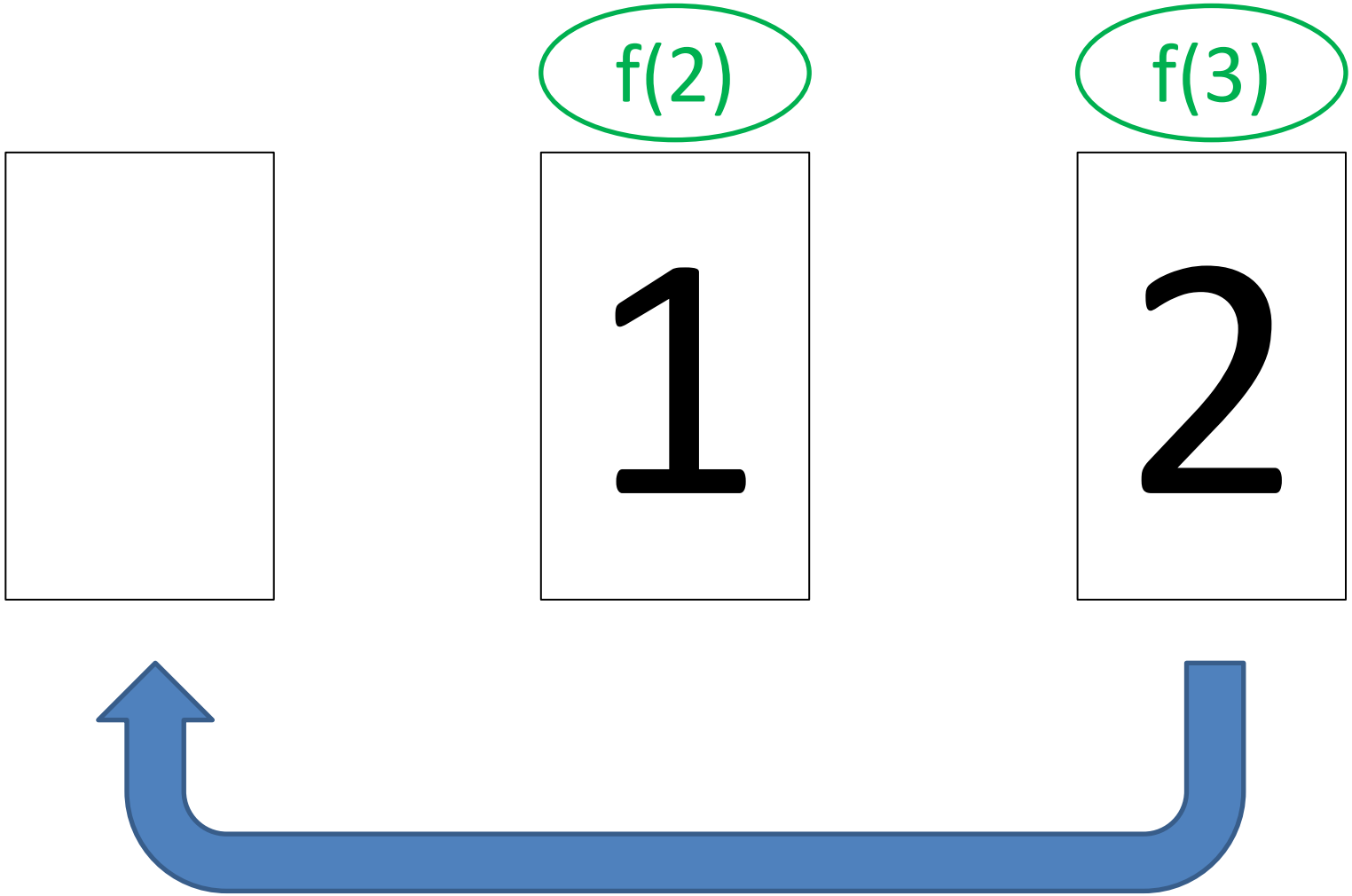
2

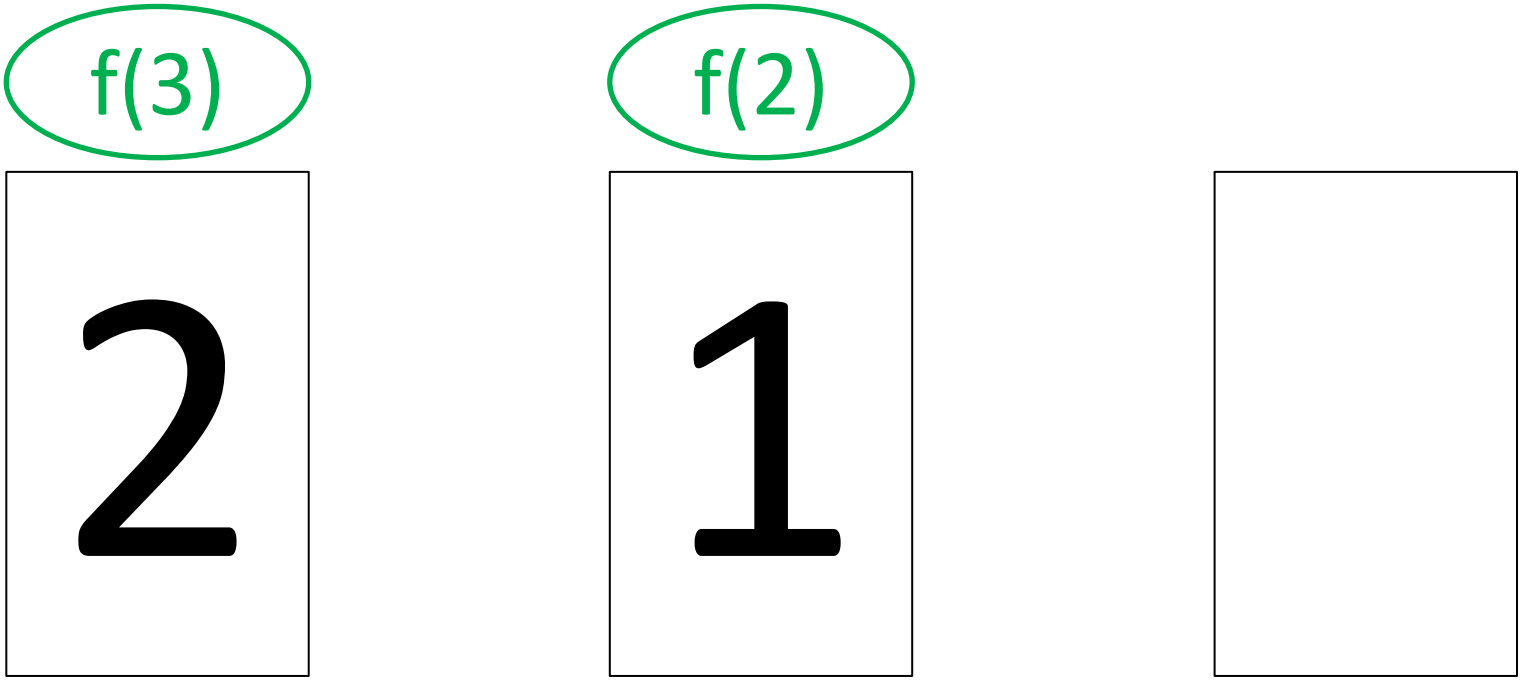


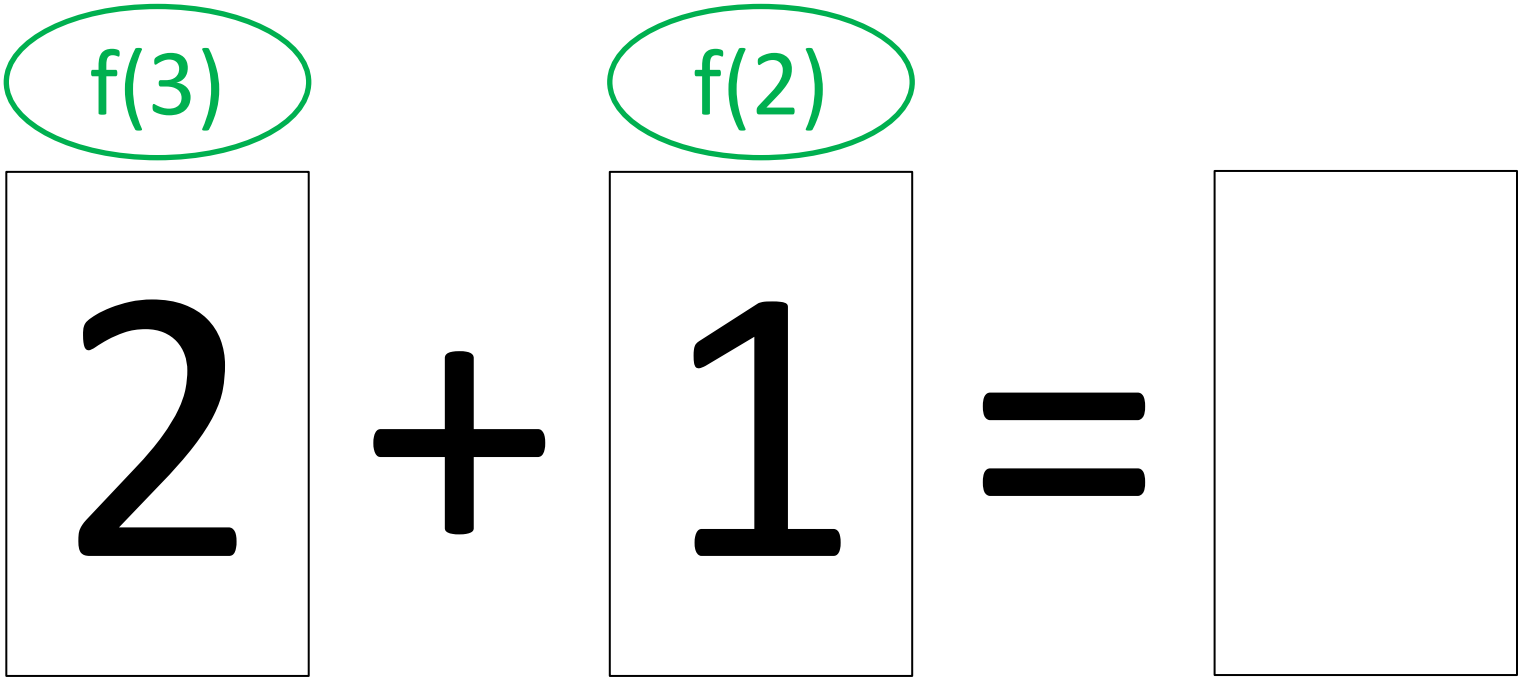

f(2)
1
f(3)
2

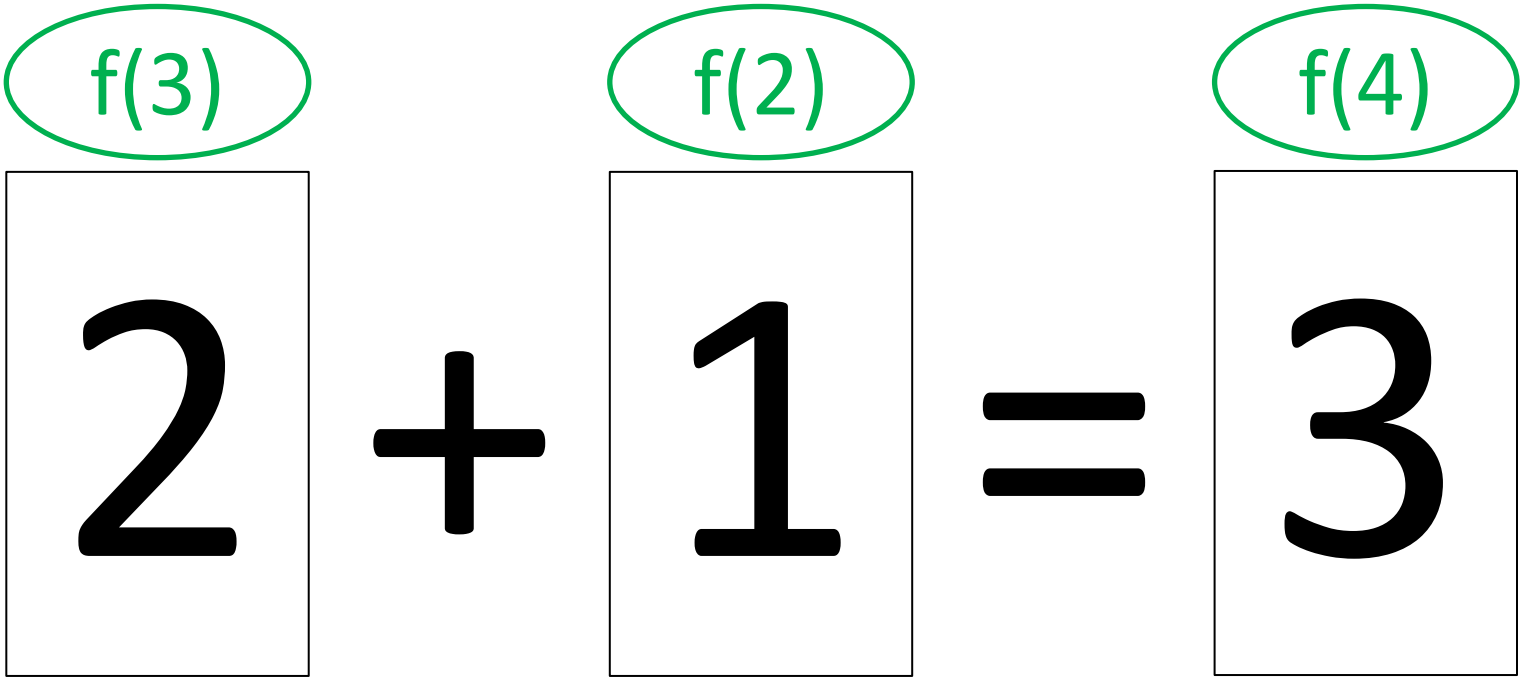

f(2)
1
f(3)
2

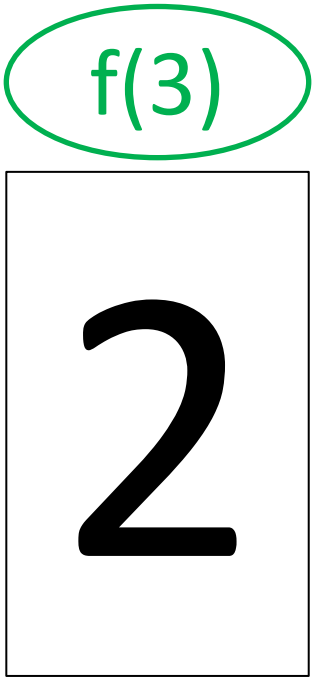

f(2)
f(3)
1
2

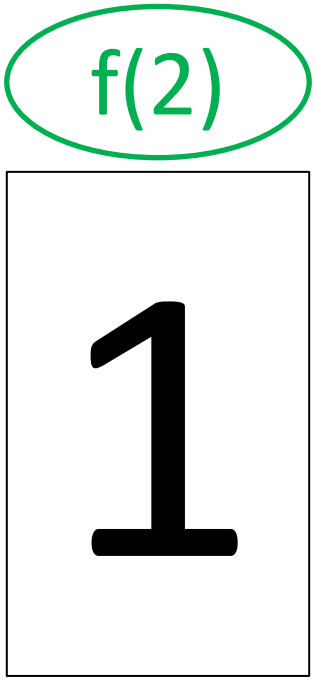

f(2)
f(3)
1
2

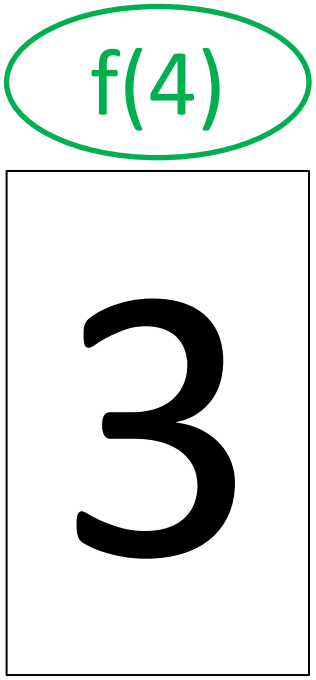

f(3)
f(2)
2
1


f(3)
f(2)
2
+
1
=


f(3)
f(2)
f(4)
2
+
1
=
3


f(3)
2


f(2)
1


f(4)
3

f(3)

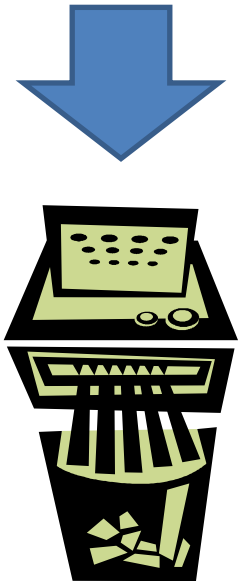
2

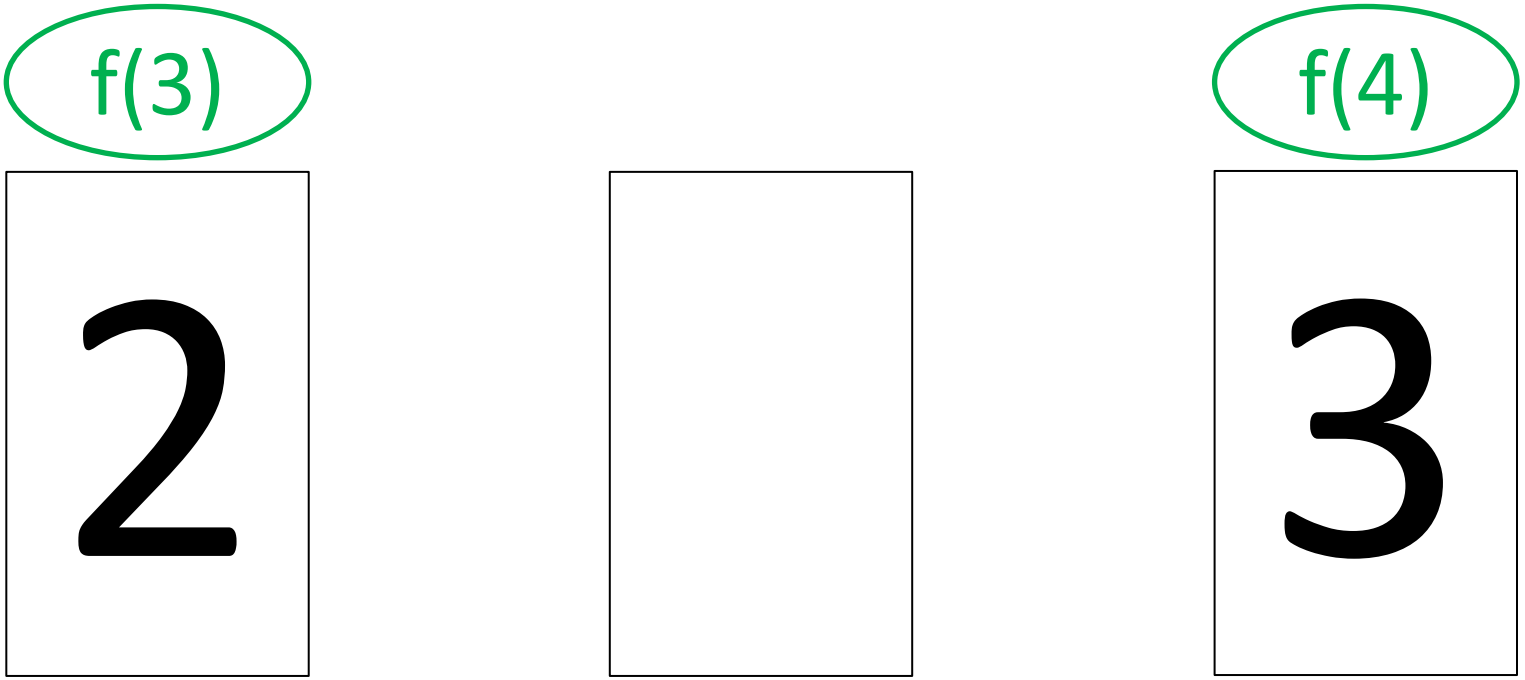
f(2)

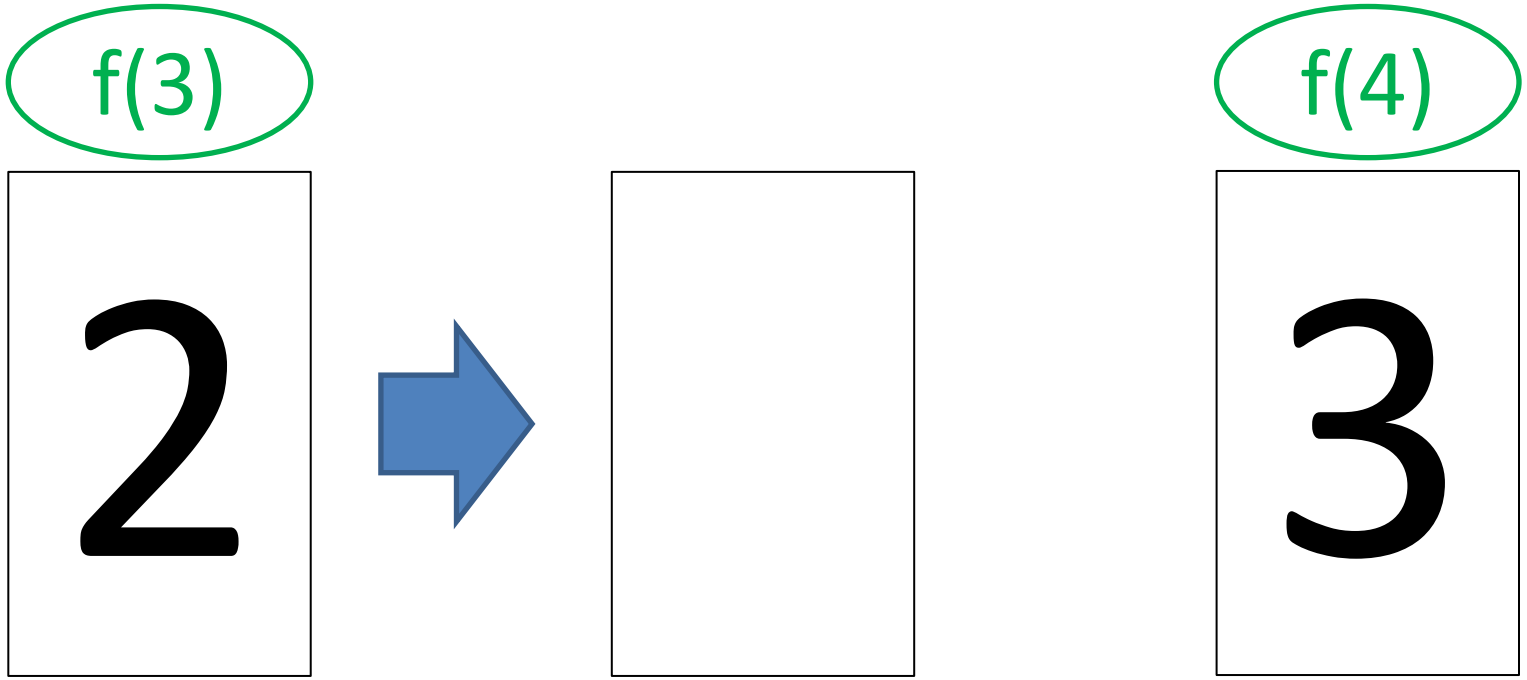
1

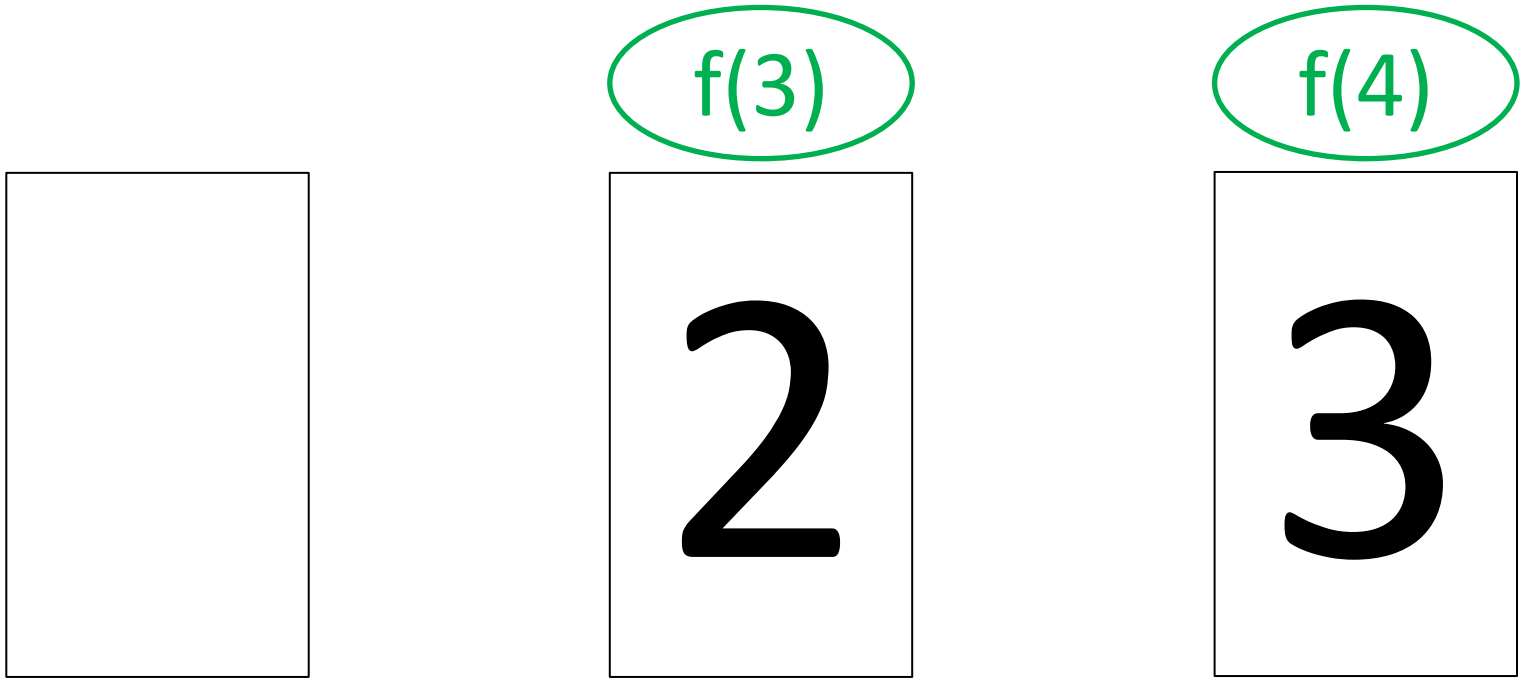
f(4)

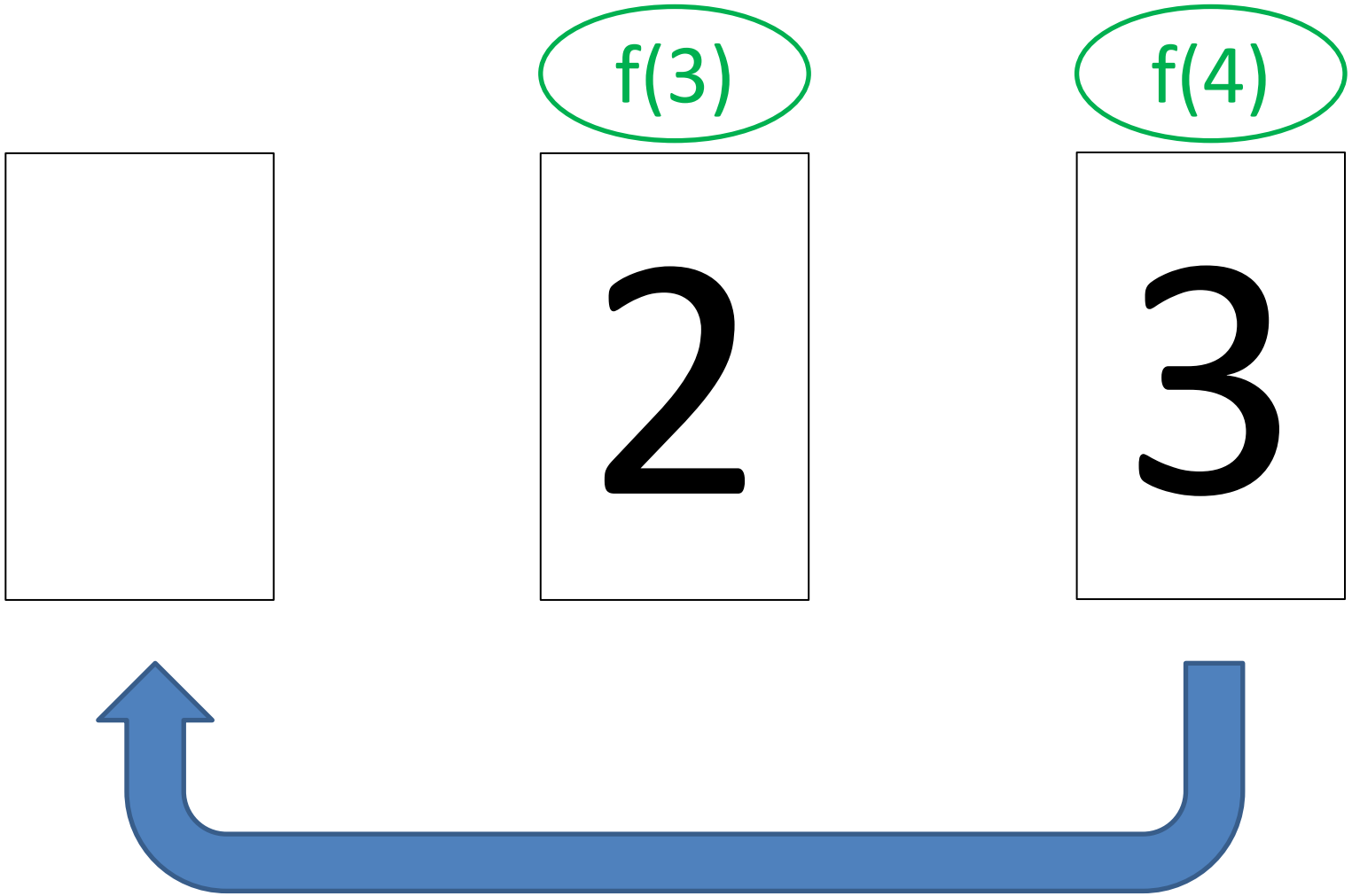
3

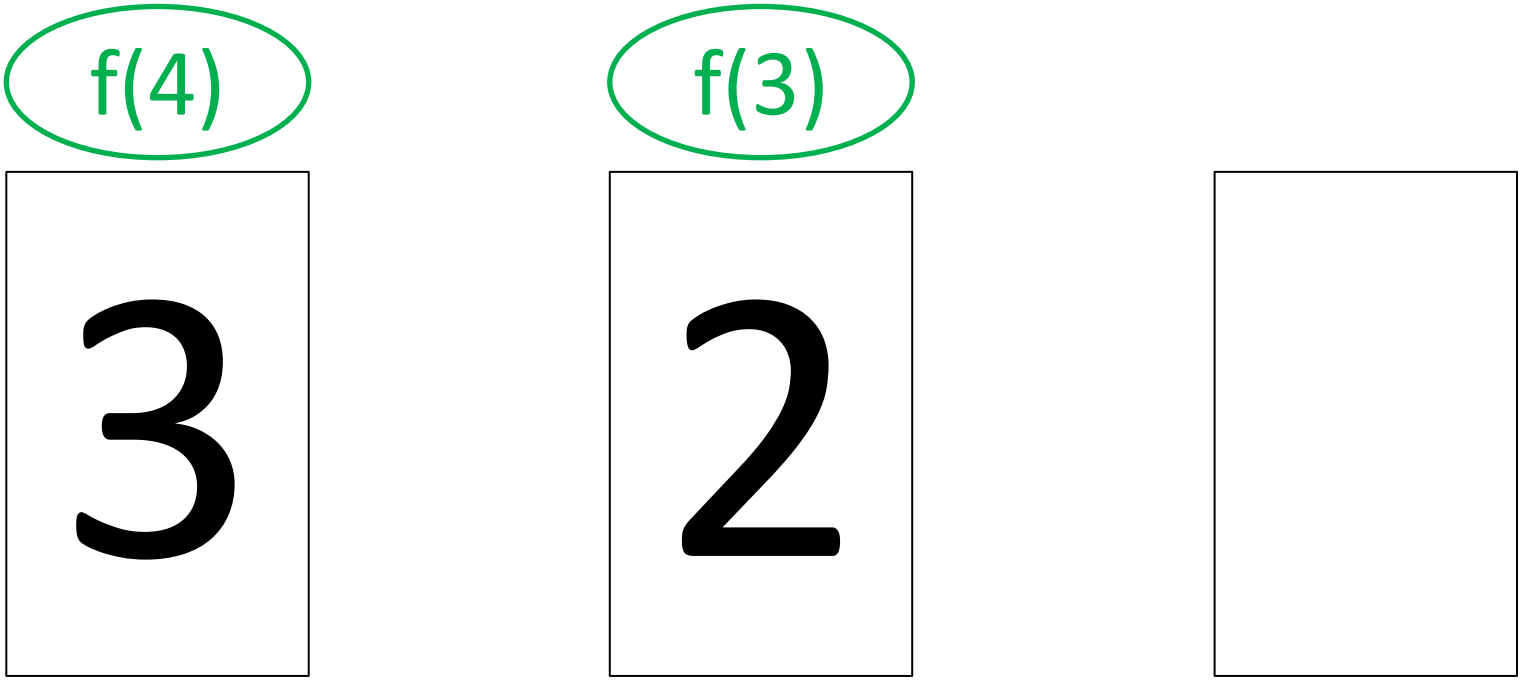


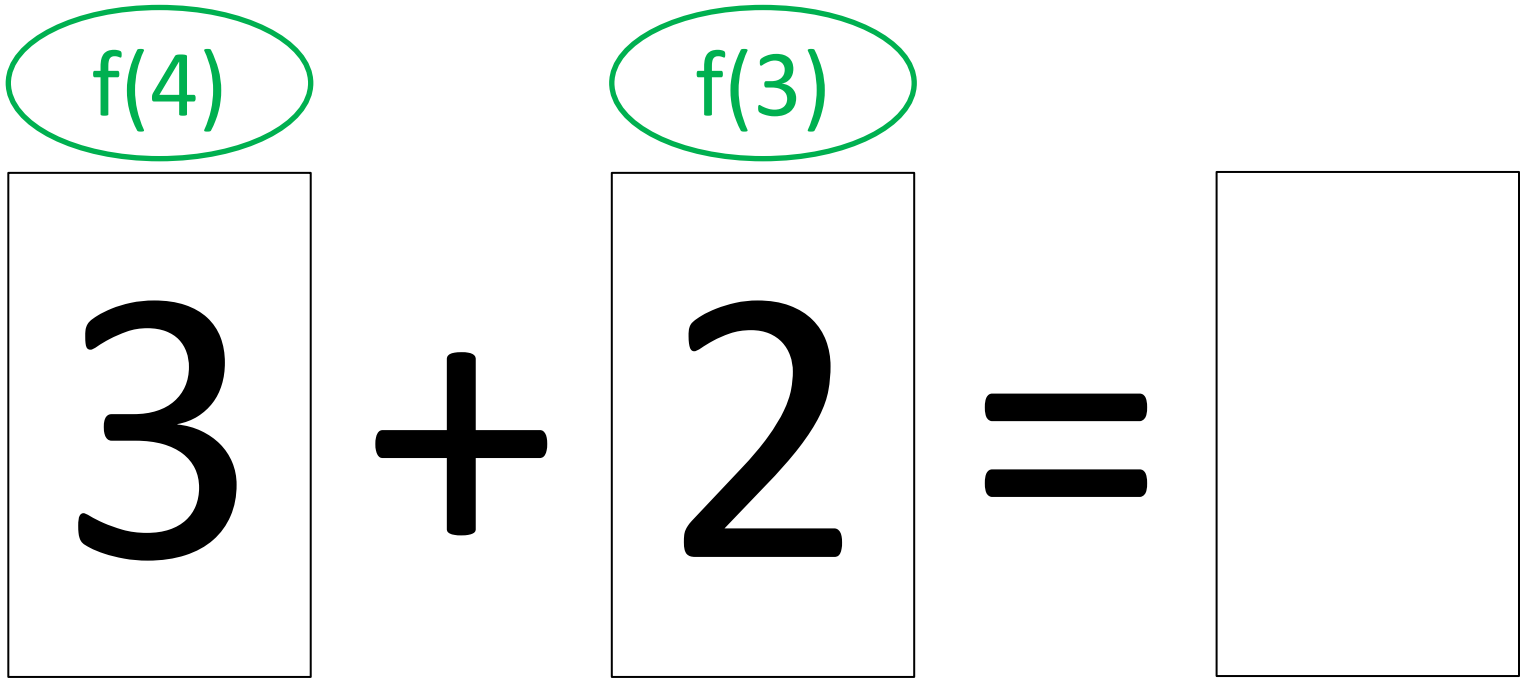

f(3)
2
f(4)
3

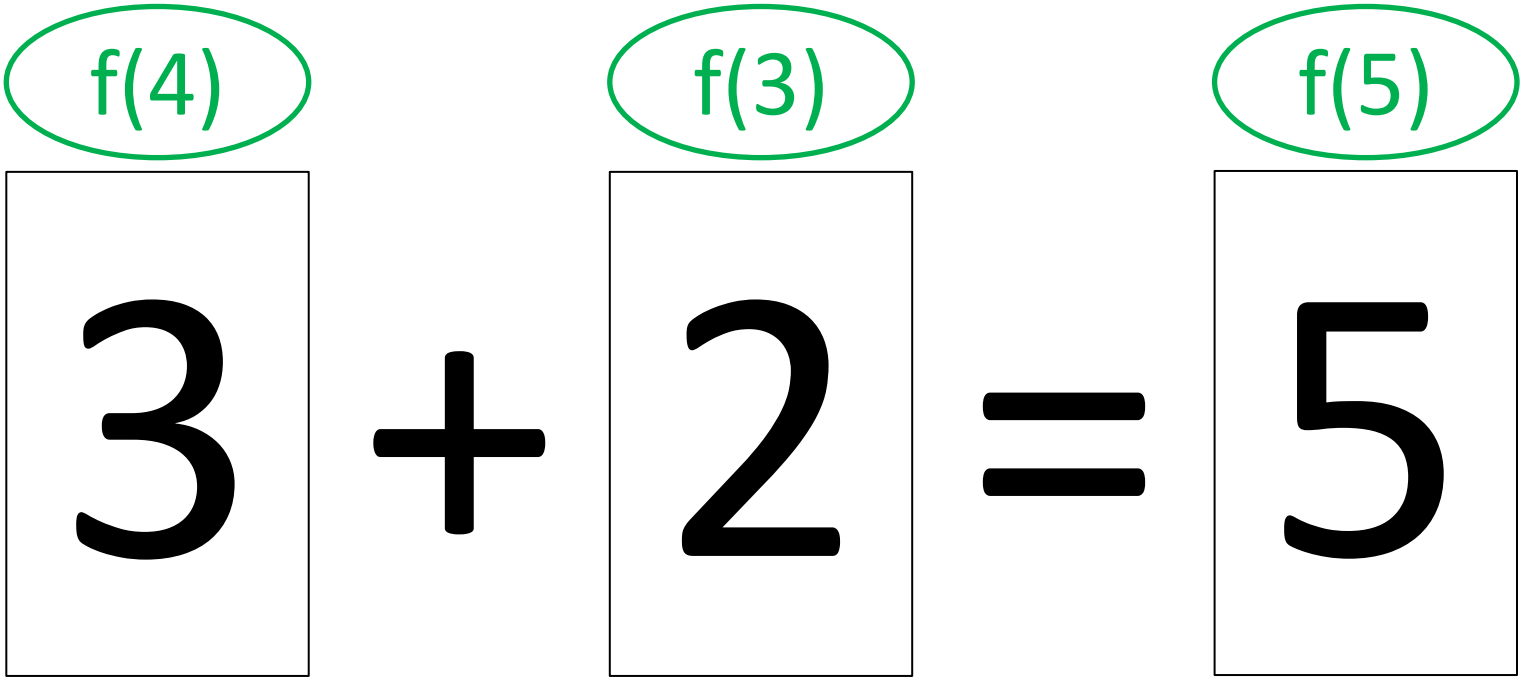

f(3)
2
f(4)
3

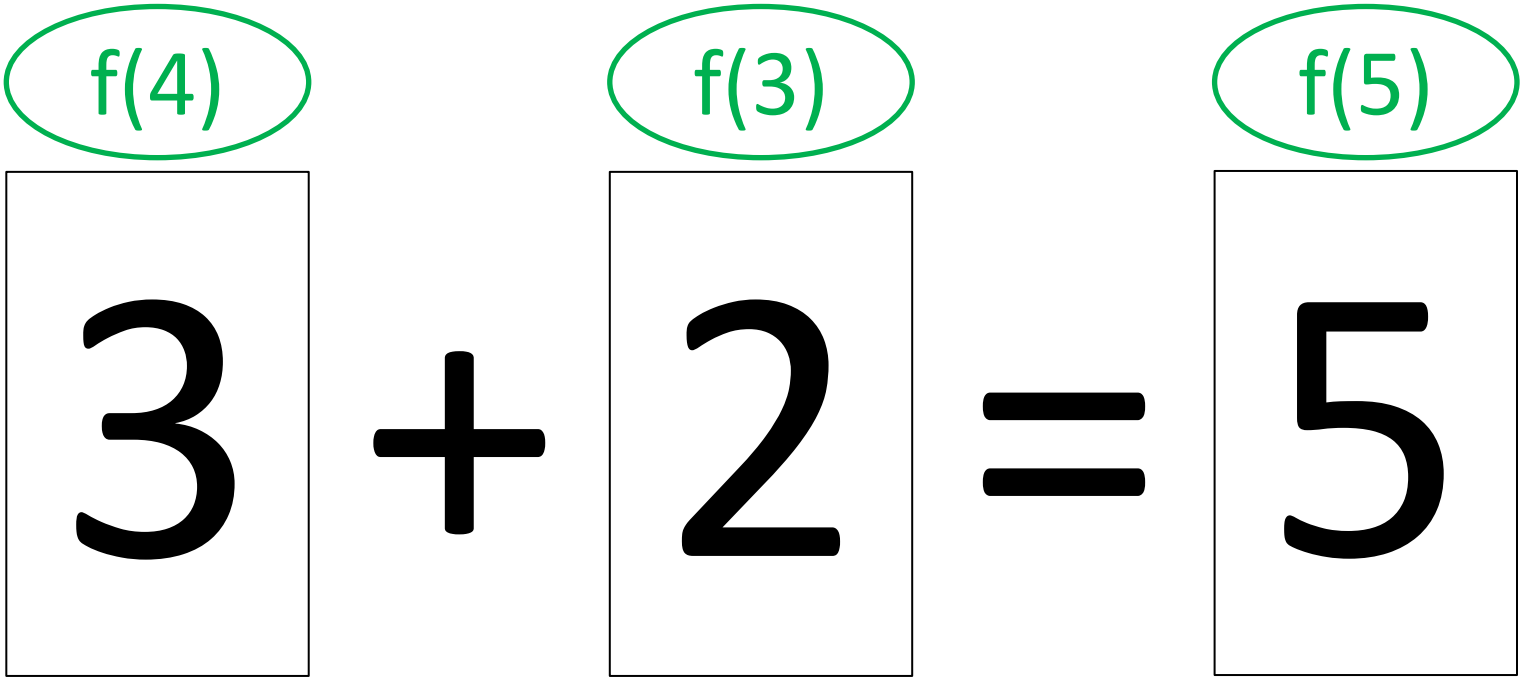

f(3)
f(4)
2
3


f(3)
f(4)
2
3


f(4)
3
f(3)
2


f(4)
f(3)
3
+
2
=


f(4)
f(3)
f(5)
3
+
2
=
5



usw.