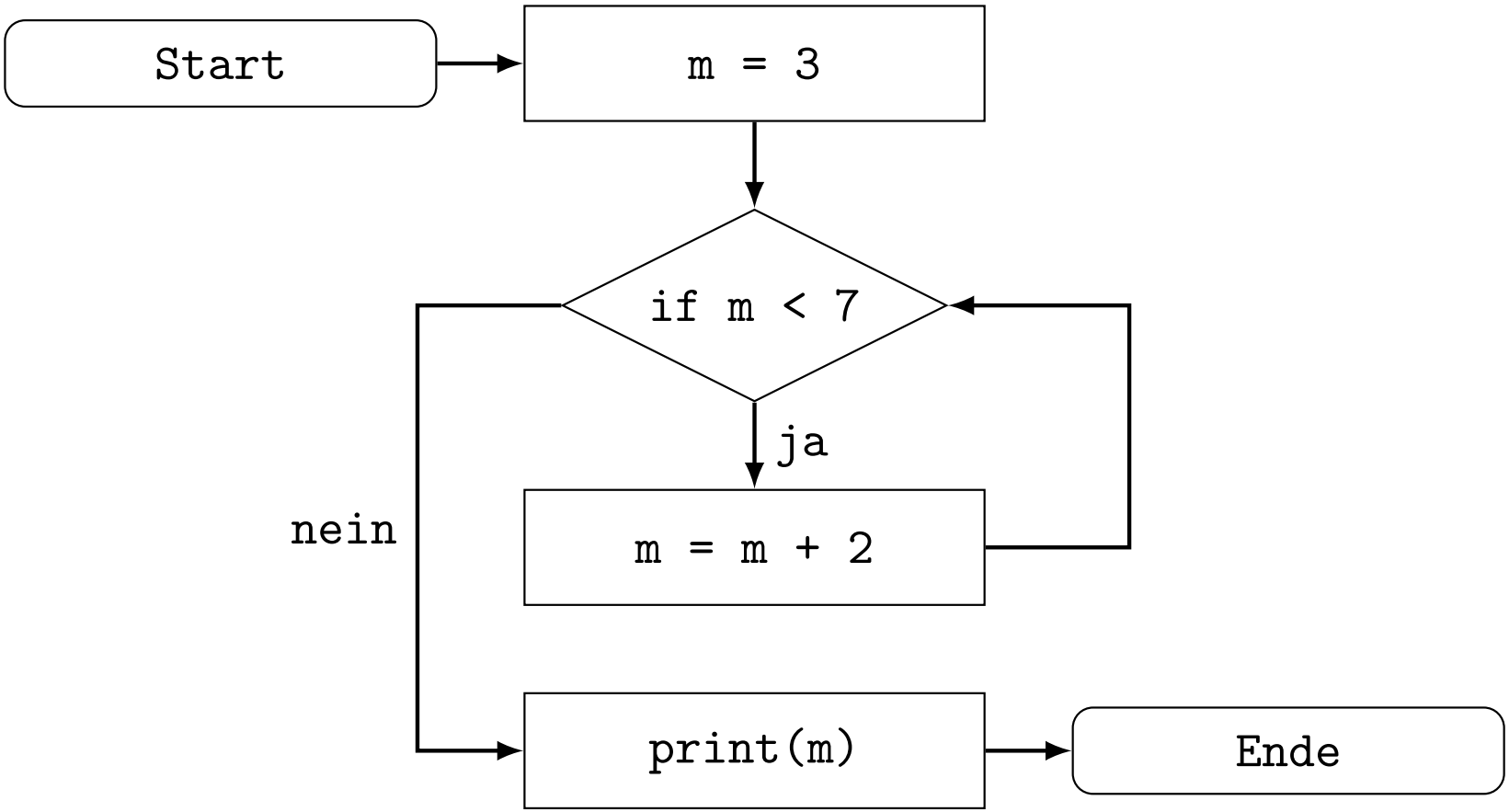
Python (Schleifen)

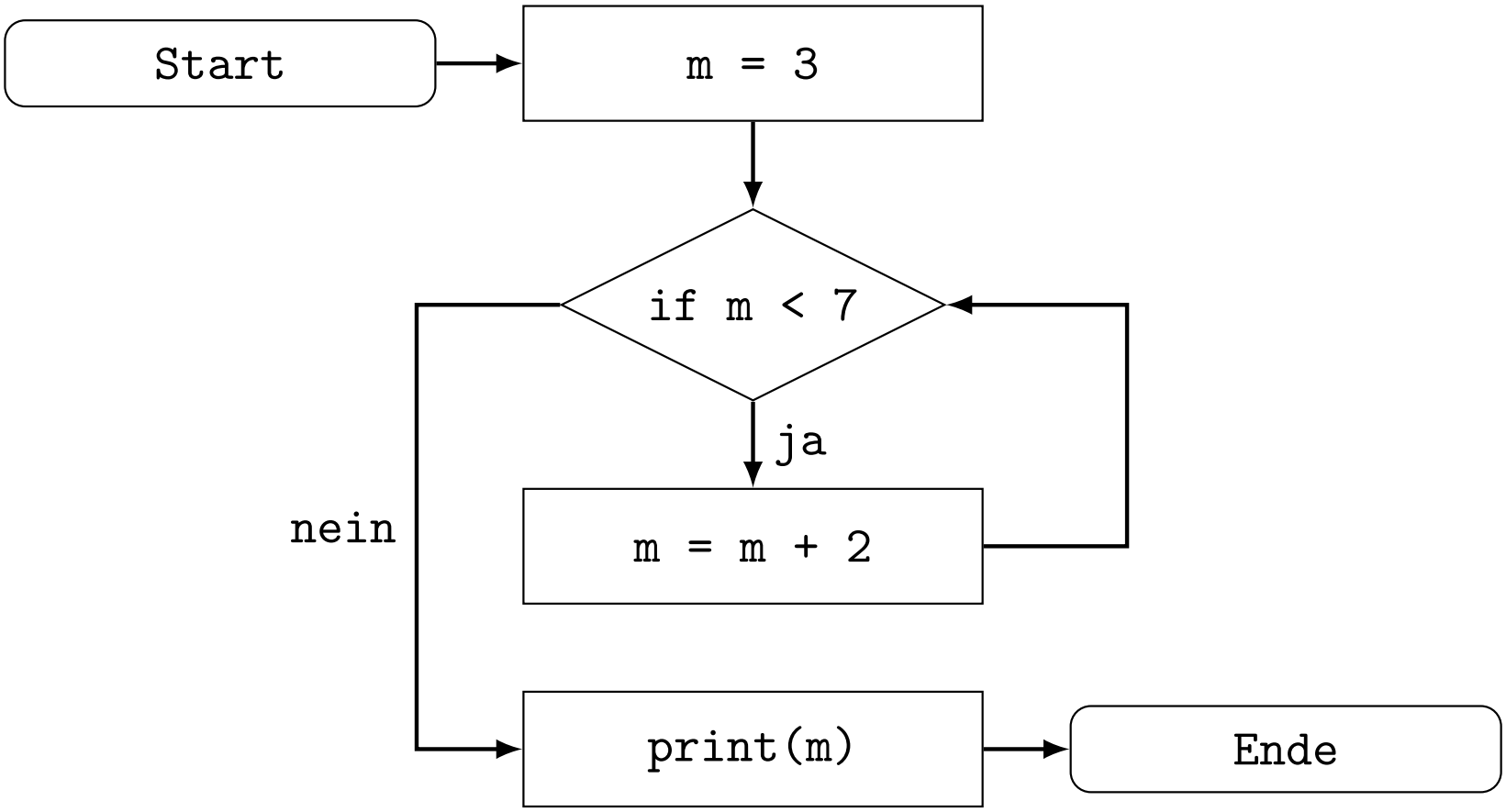
Übungen

Aufgabe 1

Welche Ausgabe macht das durch das Flussdiagramm gegebene Programm?



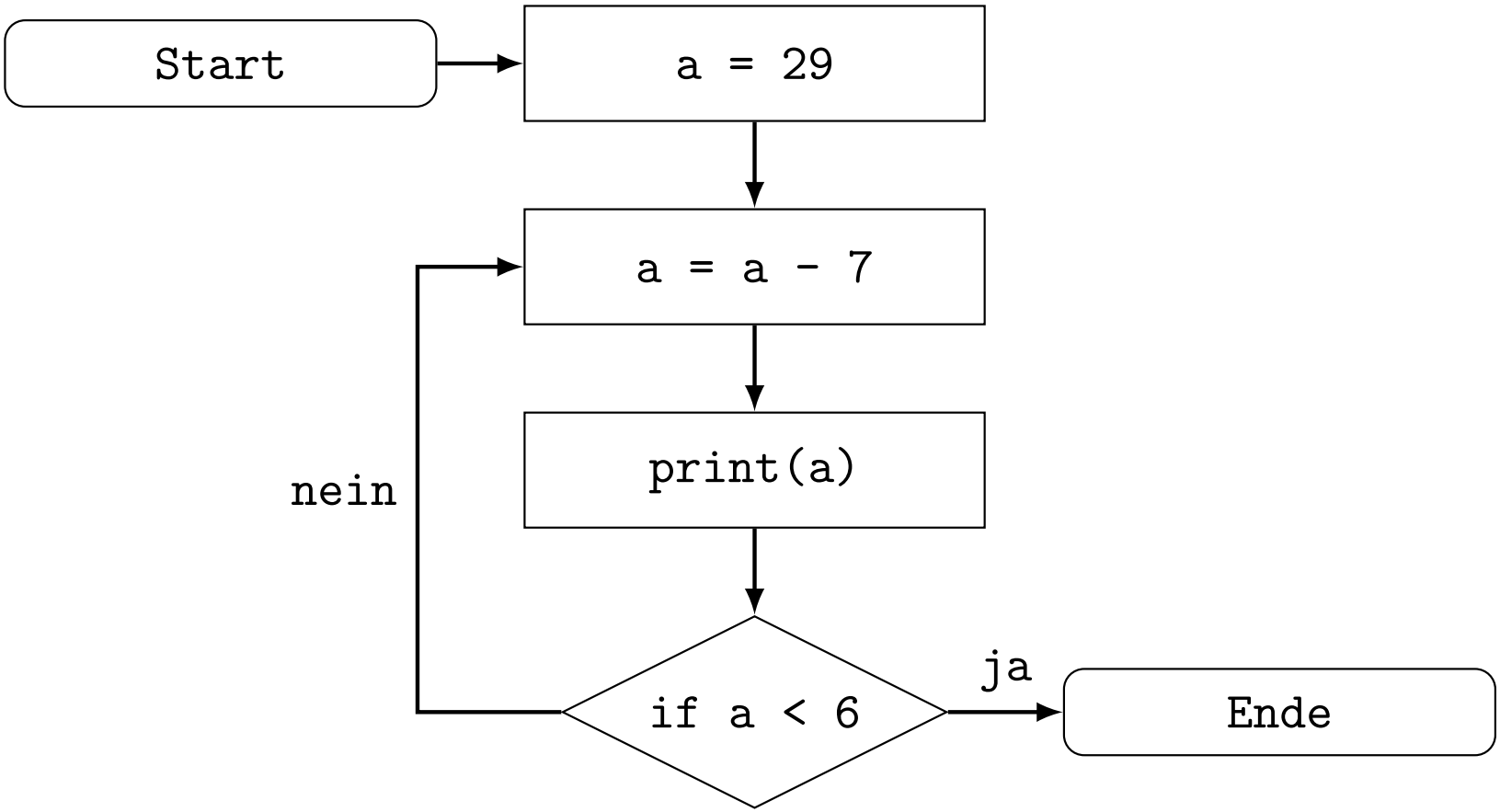
Aufgabe 1



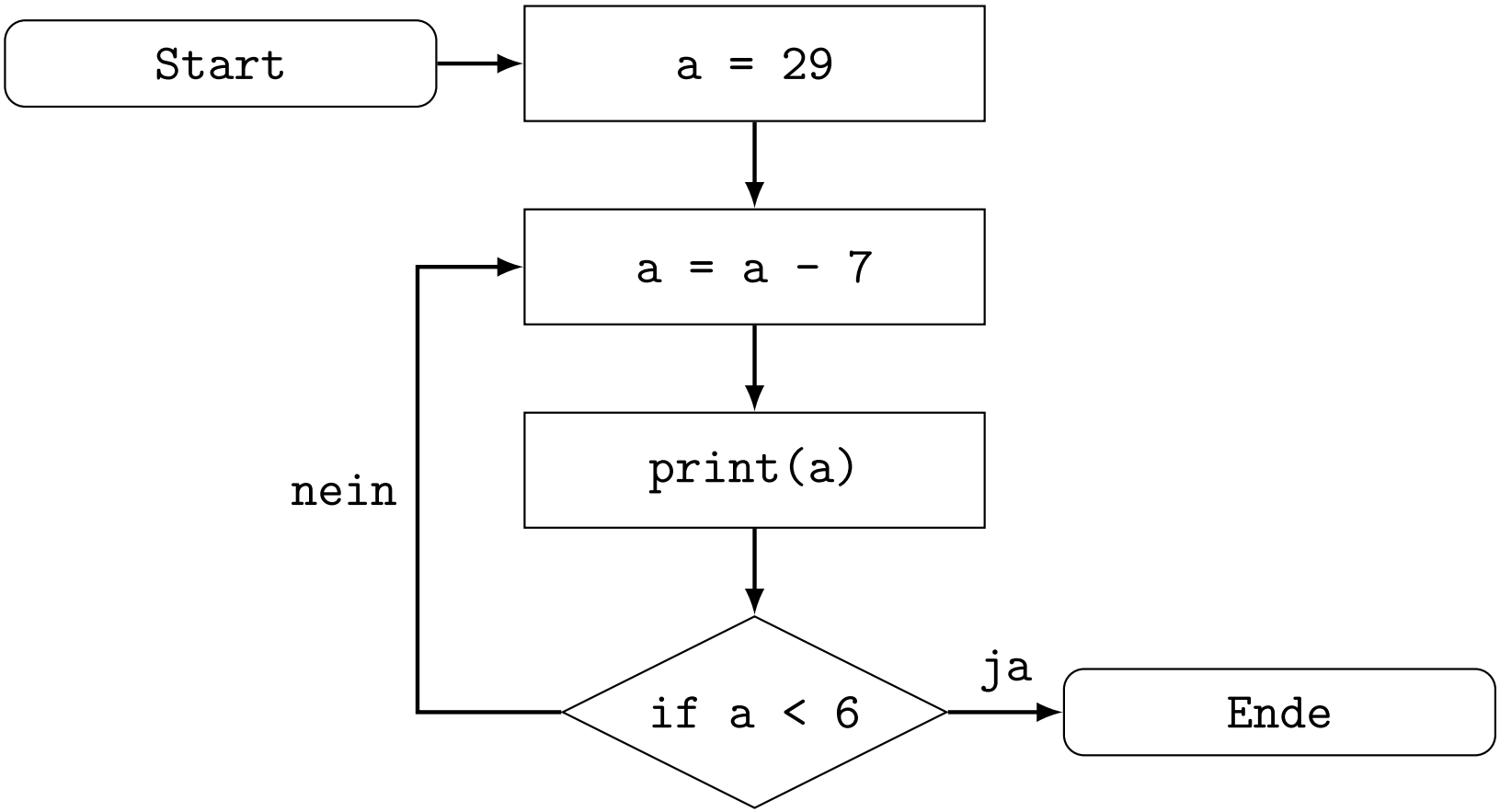
Ausgabe: 9

Aufgabe 2

Welche Ausgabe macht das durch das Flussdiagramm gegebene Programm?



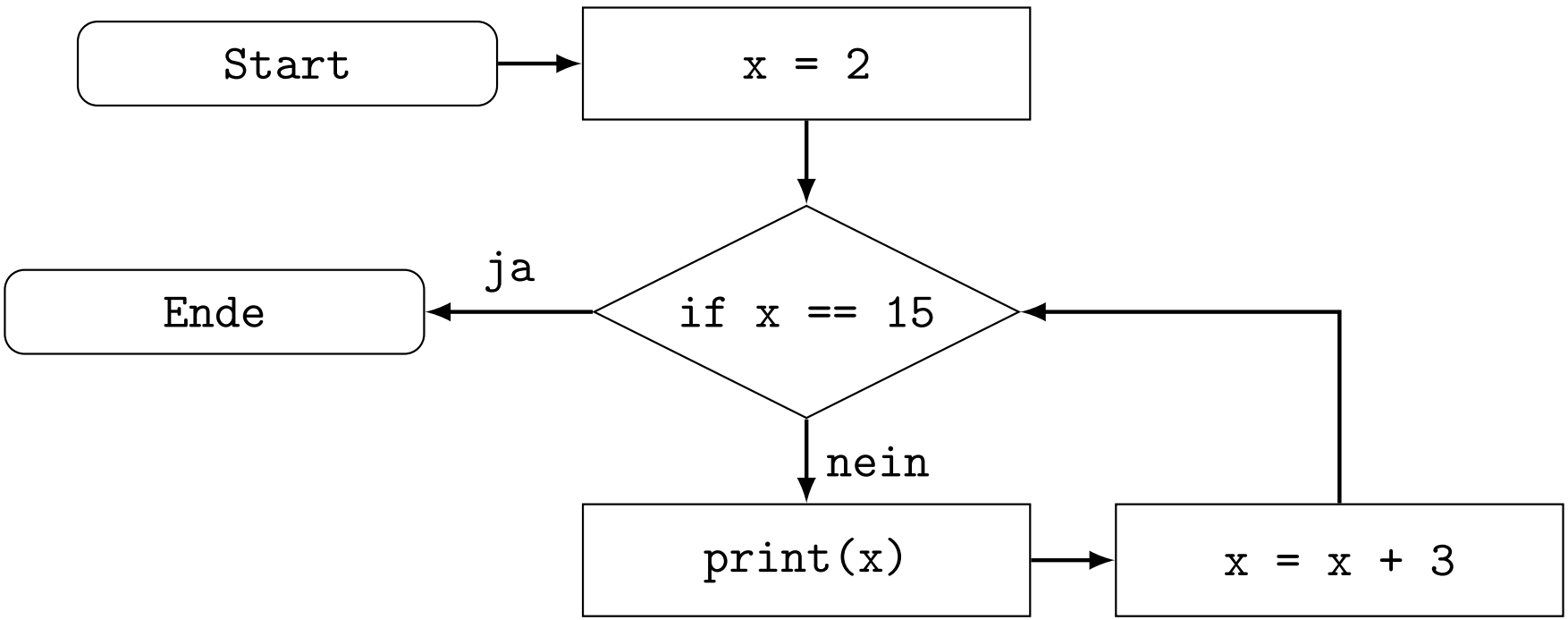
Aufgabe 2



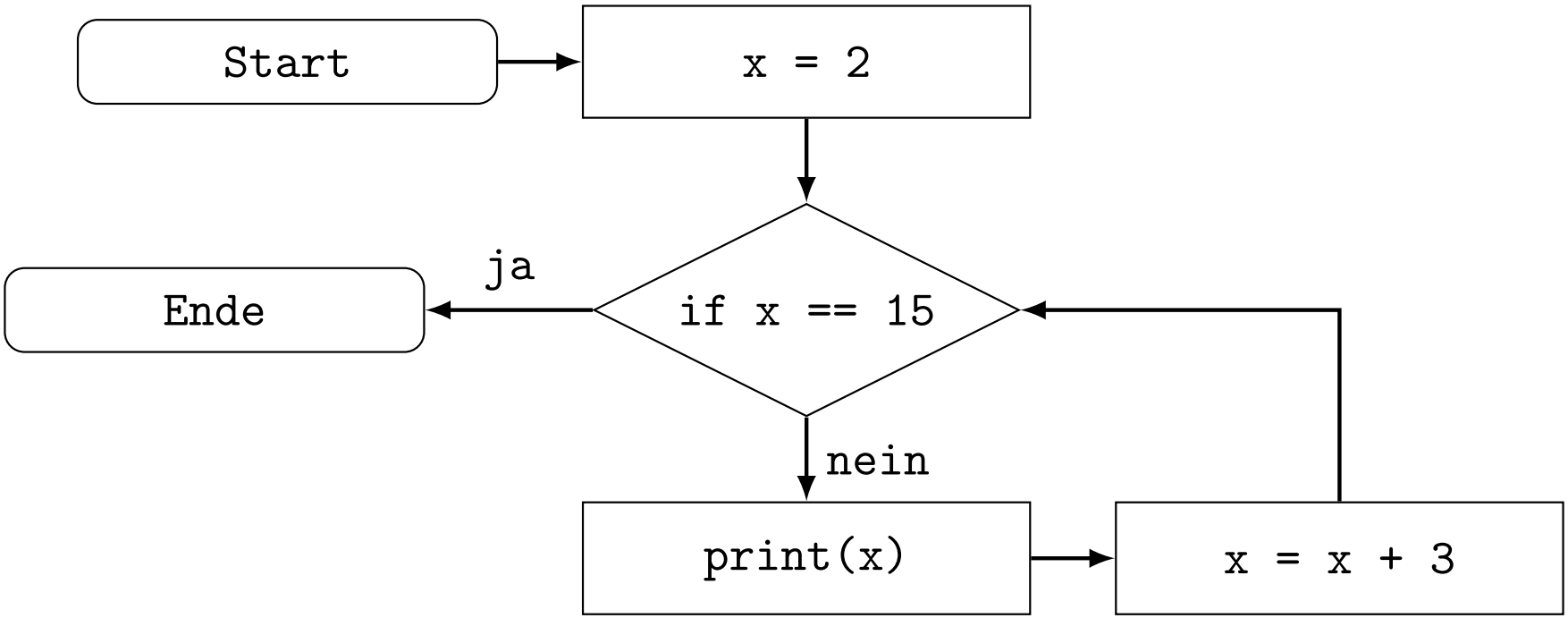
Ausgabe: 22, 15, 8, 1

Aufgabe 3

Welche Ausgabe macht das durch das Flussdiagramm gegebene Programm? Wird das Programm beendet? Begründen Sie Ihre Antwort.



Aufgabe 3



Ausgabe: 2, 5, 8, 11, 14, 17, ...

Da die Variable `x` den Wert 15 überspringt, erreicht sie niemals den Wert 15.

Aufgabe 4

Welche Ausgabe erzeugt das Programm?

```
for i in range(2, 5):
    print(10*i)
```

Aufgabe 4

```
for i in range(2, 5):
    print(10*i)
```

20
30
40

Aufgabe 5

Welche Ausgabe erzeugt das Programm?

```
for i in range(3, 16, 3):
    print(i)
```

Aufgabe 5

```
for i in range(3, 16, 3):
    print(i)
```

3
6
9
12
15

Aufgabe 6

Welche Ausgabe erzeugt das Programm?

```
for i in range(12, 2, -2):
    print(i)
```

Aufgabe 6

```
for i in range(12, 2, -2):
    print(i)
```

```
12
10
8
6
4
```

Aufgabe 7

Welche Ausgabe erzeugt das Programm?

```
farben = ["rot", "gruen", "blau"]
for farbe in farben:
    print(farbe)
```

Aufgabe 7

```
farben = ["rot", "gruen", "blau"]
for farbe in farben:
    print(farbe)
```

```
rot
gruen
blau
```

Aufgabe 8

Welche Ausgabe erzeugt das Programm?

```
zahlen = [2, 5, 8, 11]
for zahl in zahlen:
    print(zahl * 2)
```

Aufgabe 8

```
zahlen = [2, 5, 8, 11]
for zahl in zahlen:
    print(zahl * 2)
```

```
4
10
16
22
```

Aufgabe 9

Welche Ausgabe erzeugt das Programm?

```
x = 1
while x < 6:
    print(x)
    x += 1
```

Aufgabe 9

```
x = 1
while x < 6:
    print(x)
    x += 1
```

1
2
3
4
5

Aufgabe 10

Welche Ausgabe erzeugt das Programm?

```
x = 0
while x < 10:
    x += 3
    print(x)
```

Aufgabe 10

```
x = 0
while x < 10:
    x += 3
    print(x)
```

```
3
6
9
12
```

Aufgabe 11

Welche Ausgabe erzeugt das Programm?

```
x = 15
while x >= 3:
    print(x)
    x -= 3
```

Aufgabe 11

```
x = 15
while x >= 3:
    print(x)
    x -= 3
```

```
15
12
9
6
3
```

Aufgabe 12

Welche Ausgabe erzeugt das Programm?

```
x = 2
while x <= 50:
    print(x)
    x *= 2
```

Aufgabe 12

```
x = 2
while x <= 50:
    print(x)
    x *= 2
```

2
4
8
16
32

Aufgabe 13

Welche Ausgabe erzeugt das Programm

```
for i in range(1, 8):
    if i == 5:
        break
    print(i)
```

Aufgabe 13

```
for i in range(1, 8):
    if i == 5:
        break
    print(i)
```

1
2
3
4

Aufgabe 14

Welche Ausgabe erzeugt das Programm? Welche Elemente werden nicht mehr verarbeitet?

```
farben = ["rot", "gruen", "blau", "gelb", "violett"]
for farbe in farben:
    if farbe == "blau":
        break
    print(farbe)
```

Aufgabe 14

```
farben = ["rot", "gruen", "blau", "gelb", "violett"]
for farbe in farben:
    if farbe == "blau":
        break
    print(farbe)
```

```
rot
gruen
```

Die Schleife wird beendet, sobald farbe den Wert `"blau"` annimmt. Die Elemente `"blau"`, `"gelb"` und `"violett"` werden nicht ausgegeben.

Aufgabe 15

Welche Ausgabe erzeugt das Programm? Welche Wirkung hat `continue`?

```
for i in range(1, 8):
    if i % 2 == 0:
        continue
    print(i)
```

Aufgabe 15

```
for i in range(1, 8):
    if i % 2 == 0:
        continue
    print(i)
```

1
3
5
7

Bei geraden Zahlen ist `i % 2 == 0` wahr. Die Anweisung continue beendet den aktuellen Schleifendurchlauf vorzeitig. Deshalb wird `print(i)` für gerade Zahlen nicht ausgeführt. Die Schleife wird mit dem nächsten Wert fortgesetzt.

Aufgabe 16

Welche Zahlen werden ausgegeben? Welche Zahlen werden übersprungen?

```
zahlen = [3, 12, 7, 20, 5]
for zahl in zahlen:
    if zahl < 10:
        continue
    print(zahl)
```

Aufgabe 16

```
zahlen = [3, 12, 7, 20, 5]
for zahl in zahlen:
    if zahl < 10:
        continue
    print(zahl)
```

```
12
20
```

Alle Zahlen kleiner als 10 werden übersprungen. Bei den übrigen Zahlen wird `print(zahl)` ausgeführt.

Aufgabe 17

Welche Ausgabe erzeugt das Programm?

```
x = 1
while x < 8:
    if x == 5:
        break
    print(x)
    x += 1
```

Aufgabe 17

```
x = 1
while x < 8:
    if x == 5:
        break
    print(x)
    x += 1
```

1
2
3
4

Aufgabe 18

Welche Ausgabe erzeugt das Programm? Weshalb wird die Schleife beendet?

```
x = 3
while x <= 15:
    print(x)
    if x % 2 == 0:
        break
    x += 3
```

Aufgabe 18

```
x = 3
while x <= 15:
    print(x)
    if x % 2 == 0:
        break
    x += 3
```

3
6

Beim Wert 3 ist die Bedingung `x % 2 == 0` falsch. Die Variable wird auf 6 erhöht. Dieser Wert wird ausgegeben, und die Bedingung ist nun wahr. Die Schleife wird durch `break` beendet.

Aufgabe 19

Welche Ausgabe erzeugt das Programm? Welche Werte werden übersprungen?

```
x = 0
while x < 7:
    x += 1
    if x % 2 == 0:
        continue
    print(x)
```

Aufgabe 19

```
x = 0
while x < 7:
    x += 1
    if x % 2 == 0:
        continue
    print(x)
```

1
3
5
7

Bei geraden Werten von `x` wird der aktuelle Schleifendurchlauf durch `continue` abgebrochen. Die Ausgabe wird übersprungen. Da `x` bereits vor der Prüfung erhöht wurde, kann die Schleife anschliessend mit dem nächsten Wert fortgesetzt werden.

Aufgabe 20

Welche Ausgabe erzeugt das Programm? Welche Werte werden nicht ausgegeben?

```
x = 0
while x < 10:
    x += 1
    if x % 3 == 0:
        continue
    print(x)
```

Aufgabe 20

```
x = 0
while x < 10:
    x += 1
    if x % 3 == 0:
        continue
    print(x)
```

```
1
2
4
5
7
8
10
```

Die Werte 3, 6 und 9 sind durch 3 teilbar. Bei diesen Werten wird die Ausgabe durch `continue` übersprungen.

Aufgabe 21

Endet die Schleife? Begründe deine Antwort anhand der Schleifenbedingung und der Veränderung von `x`.

```
x = 5
while x < 10:
    print(x)
    x -= 1
```

Aufgabe 21

```
x = 5
while x < 10:
    print(x)
    x -= 1
```

Die Schleife endet nicht. Die Variable `x` wird bei jedem Durchlauf um `1` verringert. Da `x` anfangs `5` beträgt und immer kleiner wird, bleibt die Bedingung `x < 10` dauerhaft wahr.

Aufgabe 22

Endet die Schleife? Begründe deine Antwort. Weshalb wird die Schleife nicht unendlich oft ausgeführt?

```
z = 8
while z < 9:
    z += 3
    print(z)
```

Aufgabe 22

```
z = 8
while z < 9:
    z += 3
    print(z)
```

11

Die Schleife endet nach einem Durchlauf. Zu Beginn ist `z = 8`, sodass die Bedingung `z < 9` wahr ist. Im Schleifenkörper wird `z` auf `11` erhöht und ausgegeben. Bei der nächsten Prüfung ist `z < 9` falsch, weshalb die Schleife beendet wird.

Aufgabe 23

Endet die Schleife? Erkläre, welche Rolle die Anweisung `continue` dabei spielt.

```
x = 0
while x < 5:
    if x == 3:
        continue
    print(x)
    x += 1
```

Aufgabe 23

```
x = 0
while x < 5:
    if x == 3:
        continue
    print(x)
    x += 1
```

Die Schleife endet nicht. Sobald `x` den Wert 3 erreicht, wird der aktuelle Durchlauf durch continue abgebrochen. Die Anweisungen `print(x)` und `x += 1` werden nicht mehr ausgeführt. Deshalb bleibt `x` dauerhaft auf 3 und die Schleifenbedingung `x < 5` bleibt wahr.

Die Zahlen `0`, `1` und `2` werden ausgegeben.

Aufgabe 24

Endet die Schleife? Welche Werte werden ausgegeben? Weshalb wird die Schleife beendet, obwohl `x < 10` weiterhin wahr wäre?

```
x = 1
while x < 10:
    x += 2
    if x == 7:
        break
    print(x)
```

Aufgabe 24

```
x = 1
while x < 10:
    x += 2
    if x == 7:
        break
    print(x)
```

3
5

Die Schleife endet durch break. Die Variable `x` nimmt nacheinander die Werte 3, 5 und 7 an. Sobald `x == 7` gilt, wird die Schleife beendet, bevor `print(x)` ausgeführt wird.

Aufgabe 25

Erkläre den Unterschied zwischen einer kopfgesteuerten und einer fussgesteuerten Schleife.

Aufgabe 25

Bei einer kopfgesteuerten Schleife wird die Bedingung vor jedem Schleifendurchlauf geprüft. Bei einer fussgesteuerten Schleife erfolgt die Prüfung erst nach dem Schleifenkörper, sodass dieser mindestens einmal ausgeführt wird.

Python besitzt keine eigene fussgesteuerte Schleife.

Aufgabe 26

Erkläre den grundlegenden Unterschied zwischen einer `for`-Schleife und einer `while`-Schleife in Python.

Aufgabe 26

Eine `for`-Schleife durchläuft die Elemente eines Iterables, beispielsweise einer Liste oder eines durch `range()` erzeugten Zahlenbereichs. Eine `while`-Schleife wird ausgeführt, solange ihre Bedingung wahr ist.

Aufgabe 27

Welche Wirkung hat die Anweisung `break` innerhalb einer Schleife?

Aufgabe 27

`break` beendet die innerste umgebende Schleife sofort. Die Ausführung wird mit der ersten Anweisung nach dieser Schleife fortgesetzt.

Aufgabe 28

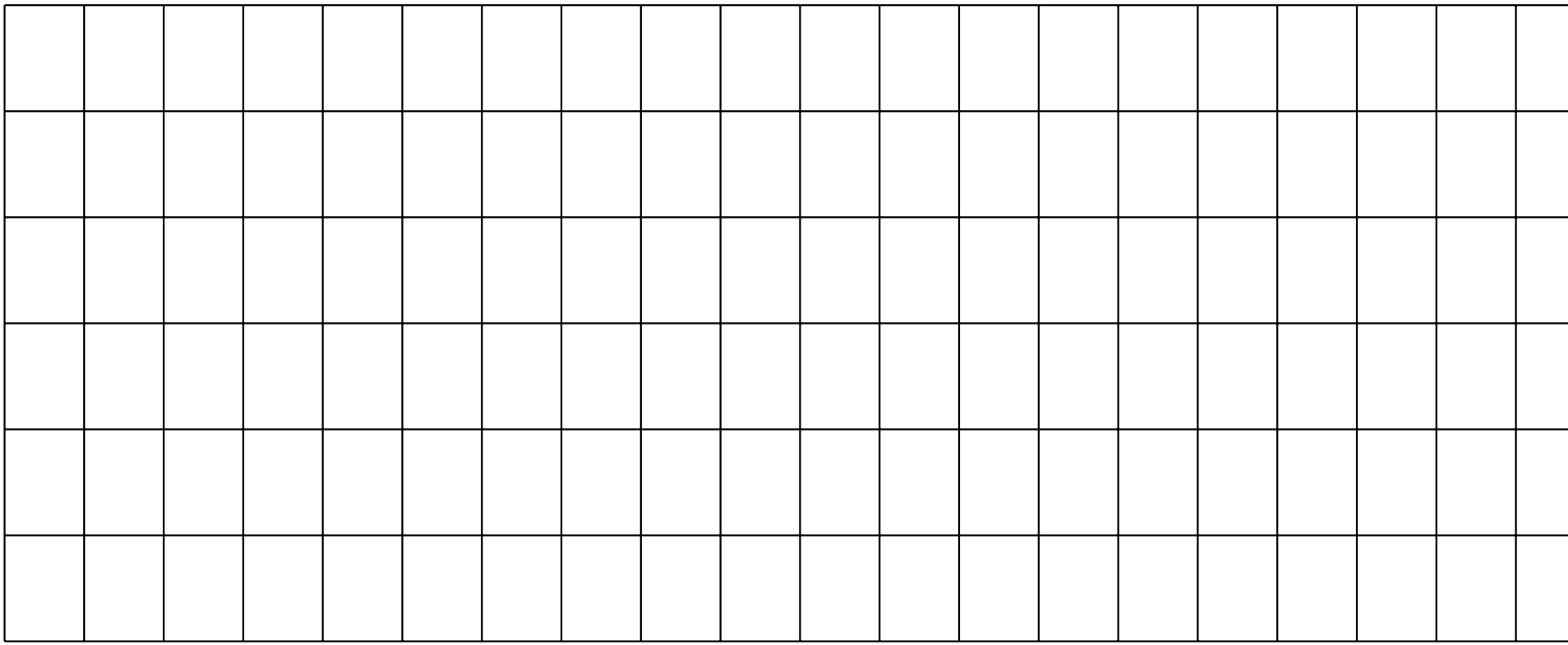
Unter welcher Voraussetzung kann eine `while`-Schleife zu einer Endlosschleife werden?

Aufgabe 28

Eine `while`-Schleife wird zur Endlosschleife, wenn ihre Bedingung dauerhaft wahr bleibt und kein anderer Kontrollfluss wie beispielsweise `break` die Schleife beendet.

Aufgabe 29

Schreibe in das folgende Raster ein Python-Programm, das in einer Schleife die Zahlen 3, 4, . . . , 70, 81 addiert und das Resultat in der Shell ausgibt. Einrückungstiefe: 2 Leerzeichen



Aufgabe 29

```
summe = 0

for i in range(1, 82)
    summe = summe + 1

print(summe)
```

Aufgabe 30

Schreibe ein Python-Programm, das mit der Zahl 1 beginnt, diese jeweils verdoppelt und jeden Wert ausgibt, solange der Wert kleiner als 20 ist.

Verwende eine `while`-Schleife. Einrückungstiefe: 2 Leerzeichen.

Aufgabe 30

```
x = 1
while x < 20:
    print(x)
    x *= 2
```

Ausgabe:

1
2
4
8
16

Aufgabe 31

Schreibe ein Python-Programm, das die Zahlen von 1 bis 8 ausgibt, wobei die Zahl 5 übersprungen wird.

Verwende eine `while`-Schleife und `continue`. Einrückungstiefe: 2 Leerzeichen.

Aufgabe 31

```
x = 0
while x < 8 :
    x += 1
    if x == 5 :
        continue
    print(x)
```

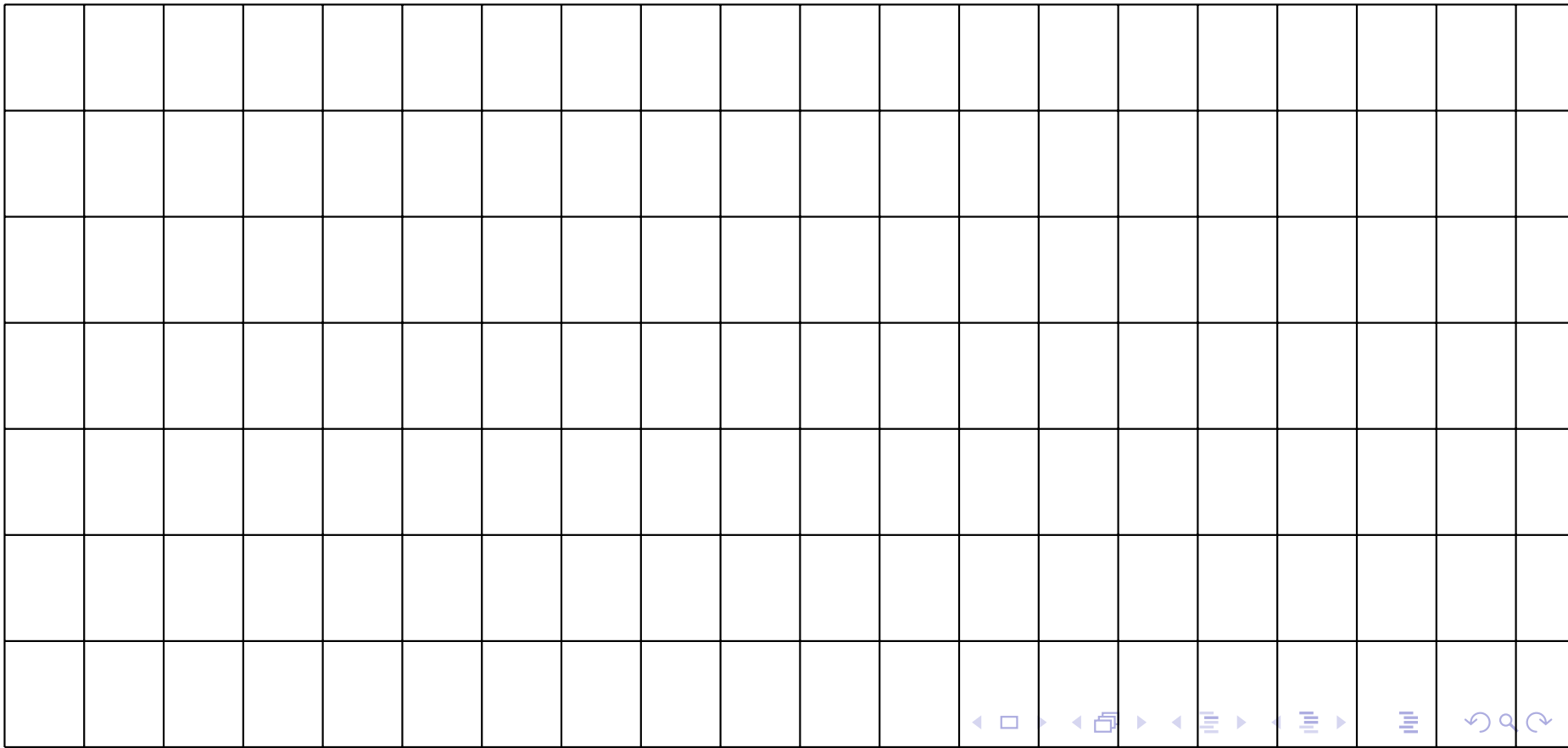
Ausgabe:

1
2
3
4
6
7

Aufgabe 32

Schreibe ein Python-Programm, das mit der Zahl 1 beginnt, jeden Wert ausgibt und die Zahl anschliessend verdoppelt.

Sobald der neue Wert mindestens 8 beträgt, soll die Schleife beendet werden. Verwende `while True` und `break`.
Einrückungstiefe: 2 Leerzeichen.



Aufgabe 32

```
x = 1
while True:
    print(x)
    x *= 2
    if x >= 8:
        break
```

Ausgabe:

1
2
4