

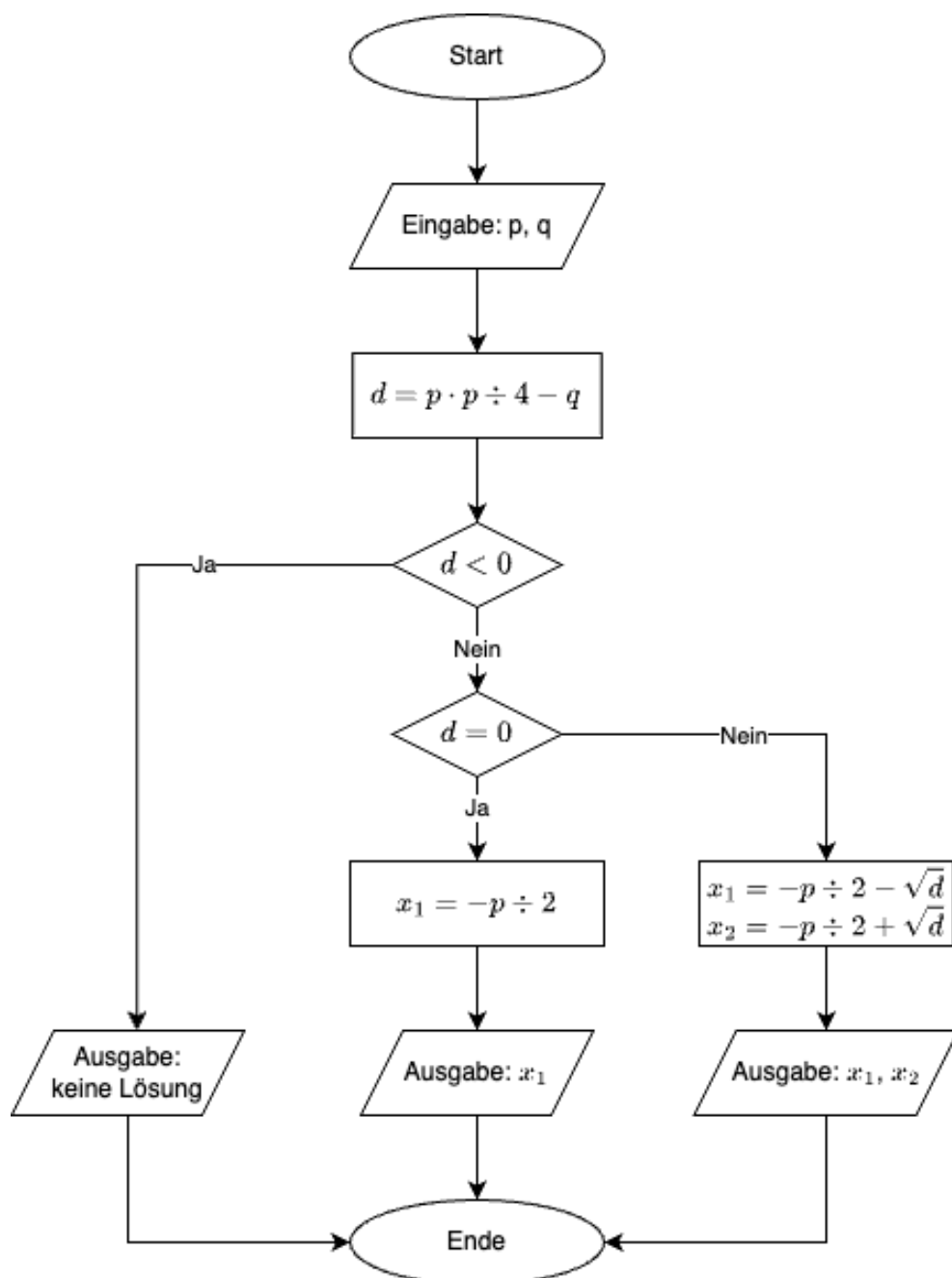
# Programmbeschreibung und Flussdiagramm

**Hinweis**

Mit diesem Programm soll Mit-hilfe der p-q-Formel die Nullstellen einer quadratischen Funktion berechnet werden.

Die Umsetzung des Programm verwendet:

- if-Abfrage
- else if - Abfrage
- Variablen
- Einsatz einer Formel



# Schritt-für-Schritt-Anleitung

In dem folgenden Teil führt „Schritt-für-Schritt“ durch das Flussdiagramm.

Am Ende dieser Anleitung steht das vollständige Programm passend zum Flussdiagramm.

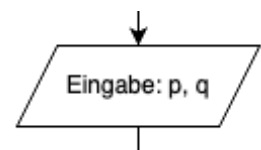
## 1. Schritt: Die Eingabe (Zeile 1 und 3)

Die Variablen p und q werden deklariert. „input“ fordert den „Nutzer“ auf, jeweils zu diesen Variablen Werte anzugeben. „int()“ wandelt die Eingaben in den Typ „integer“ um, denn die Eingabe erfolgt als „String“.

„import math“ fügt später benötigte mathematische Funktionen hinzu.

Python

```
1 import math
2 q = int(input('Gebe den Wert für q an: '))
3 p = int(input('Gebe den Wert für p an: '))
```

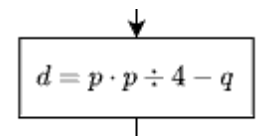


## 2. Schritt: Berechnung der Diskriminante (Zeile 4)

Die Variable d wird gleich mit der passenden Berechnung deklariert.

Python

```
1 import math
2 q = int(input('Gebe den Wert für q an: '))
3 p = int(input('Gebe den Wert für p an: '))
4 d = p * p / 4 - q
```

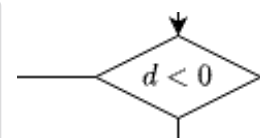


## 3. Schritt: Die erste Abzweigung (Zeile 6 bis 7)

Wenn  $d < 0$ , dann gehe zur Ausgabe „keine Lösung“.

Python

```
1 import math
2 q = int(input('Gebe den Wert für q an: '))
3 p = int(input('Gebe den Wert für p an: '))
4 d = p * p / 4 - q
5
6 if (d < 0):
7     print('Keine Lösung')
```

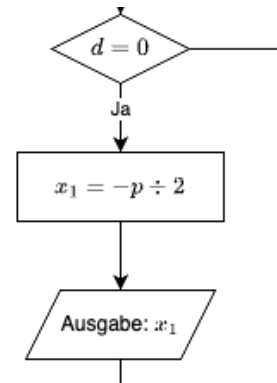


**4. Schritt:** Die zweite Abzweigung. (Zeile 8 bis 10).

Wenn  $d = 0$ , dann Berechne  $x_1$  gehe zur Ausgabe „ $x_1$ “.

Python

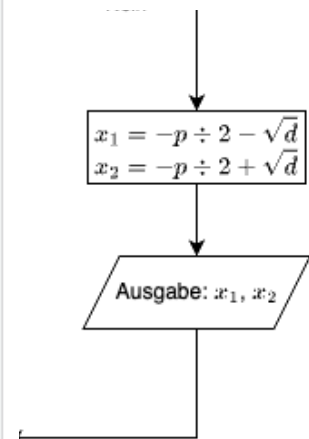
```
1 import math
2 q = int(input('Gebe den Wert für q an: '))
3 p = int(input('Gebe den Wert für p an: '))
4 d = p * p / 4 - q
5
6 if (d < 0):
7     print('Keine Lösung')
8 elif (d == 0):
9     x1 = -p/2
10    print(x1)
```

**5. Schritt:** Letzte Berechnung. (Zeile 12 bis 14).

Wenn  $d > 0$ , dann Berechne  $x_1$  und  $x_2$  gehe zur Ausgabe „ $x_1, x_2$ “.

Python

```
1 import math
2 q = int(input('Gebe den Wert für q an: '))
3 p = int(input('Gebe den Wert für p an: '))
4 d = p * p / 4 - q
5
6 if (d < 0):
7     print('Keine Lösung')
8 elif (d == 0):
9     x1 = -p/2
10    print(x1)
11 else:
12    x1 = -p/2 - math.sqrt(d)
13    x2 = -p/2 + math.sqrt(d)
14    print(x1 , " , " , x2)
```

**ENDE**

# Aufgaben

- ① Füge in dem Quellcode passend zu den Schritten Kommentare ein, die den jeweiligen Teil in seiner Funktion beschreiben.

Python

```
1 # Ein "Hashtag" ergibt einen Zeilenkommentar
2
3 """Drei Anführungszeichen am Anfang und Ende ergeben
4 einen Kommentar auf mehreren Zeilen"""
```

- ② Überlege Dir mögliche Verbesserungen für das Programm. Beispielsweise:
- Könnte der Text der Eingabe und der Ausgabe verbessert werden.
  - Sollte das Ergebnis der Ausgabe gerundet werden.
  - ...

---

---

---

---

---

---

---

---

---

---

- ③ Passe das Flussdiagramm und den Quelltext entsprechend Deiner Überlegungen aus Aufgabe 2 an.