

Eine natürliche Zahl n wird vollkommene Zahl (auch perfekte Zahl) genannt, wenn sie gleich der Summe aller ihrer Teiler außer sich selbst ist.

Die erste vollkommene Zahl ist 28:

$$28 = 1 + 2 + 4 + 7 + 14$$

Dann folgt:

$$496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248$$

Vollkommenen Zahlen werden mit dem Term ...

$$2^{k-1} (2^k - 1)$$

... berechnet. Der Term ergibt immer dann eine vollkommene Zahl, wenn der Wert von $(2^k - 1)$ eine Primzahl ergibt.

So sieht es aus:



Für diese Anleitung brauchst du nur den Arduino.

```
// sorgt dafür, dass das Programm nur einmal ausgeführt wird
bool Stopp = true;

void setup()
{
  // Seriellen Monitor starten
  Serial.begin(9600);

  // kurze Pause für die korrekte Darstellung im Seriellen Monitor
  delay(500);
  Serial.println("Die ersten 4 vollkommenen Zahlen:");
  Serial.println("-----");
}
```

```

bool Primzahl(float Zahl)
{
    // zum Testen nach int umwandeln
    int PruefZahl = round(Zahl);

    // nur bis zur Hälfte der Eingabe prüfen
    for (int i = 2 ; i <= PruefZahl / 2 ; i++)
    {
        // Teilbarkeit mit modulo testen
        // wenn kein Rest → Eingabe ist teilbar → false zurückgeben
        if (int(PruefZahl) % i == 0) return false;
    }

    // keine Teiler gefunden -> Primzahl = true zurückgeben
    return true;
}

void loop()
{
    // solange Stopp = true wird loop ausgeführt
    while (Stopp)
    {
        /*
            eine vollkommene Zahl ist die Summe ihrer Teiler
            (ausgenommen 1 und die Zahl selbst)
            die ersten 4 vollkommenen Zahlen
        */
        for (int i = 2; i < 8; i++)
        {
            /*
                Formel zur Berechnung:
                 $2^{(i-1)} * (2^i - 1)$ 
                wobei der Wert des Terms  $(2^i) - 1$  eine Primzahl sein muss
            */

            // Wert des Terms  $\text{pow}(2, i) - 1$  auf Primzahl untersuchen
            bool Suche = Primzahl(pow(2, i) - 1);

            // der Wert des Terms ist eine Primzahl (Suche = true)
            // -> vollkommene Zahl gefunden
            if (Suche)
            {
                // Ergebnis ist eine Zahl vom Typ float -> muss gerundet werden
                Serial.println(round(pow(2, (i - 1)) * (pow(2, i) - 1)));
            }
        }
        Serial.println("-----");
        Stopp = false;
    }
}

```