Aufgabe 1: Beweisen Sie: Iteriertes Bilden der Quersumme einer von 6 verschiedenen vollkommenen geraden Zahl führt schließlich zur Zahl 1.
Beispiel: 28 ist eine vollkommen Zahl ungleich 6, die erste Quersumme ist $2+8=10$, die Quersumme der ersten Quersumme ist $1+0=1$.

Das Iterierte Bilden der Quersumme hängt mit der Teilbarkeit durch 9 zusammen: Eine Zahl ist durch 9 Teilbar genau dann wenn es ihre Quersumme ist. Wie entscheidet man, ob die Quersumme durch 9 teilbar ist? Man bildet wieder die Quersumme .... bis man z.B. etwas einstelliges hat.

Es gilt wegen $10 \equiv 1 \pmod 9 \Rightarrow 10^k \equiv 1^k \equiv 1 \pmod 9$: Falls $n = \sum_{k=0}^{m} n_k 10^k$ ist, dann ist $n \equiv \sum_{k=0}^{m} n_k \pmod 9$. Sei $q_1 := \sum_{k=0}^{m} n_k$, will man wissen welchen Rest mod 9 q_1 so berechnet man die Quersumme q_2 von q_1 und so weiter. n, $q_1, q_2, \dots$ haben alle den gleiche Rest mod 9. Insbesondere ist n durch 9 teilbar, falls es alle iterierten Quersummen sind, d. h. die letzte iterierte Quersumme gleich 9 ist.

In unserem Fall soll die iterierte Quersumme =1 sein, d.h. die Zahl n soll den Rest 1 mod 9 haben: Es ist zu zeigen: sei $6 < n$ eine vollkommene gerade Zahl, dann ist $n \equiv 1 \pmod 9$:

Für vollkommene gerade Zahlen n haben wir den Satz von Euler: $n = 2^{p-1} M_p$, mit einer Mersenneschen Primzahl $M_p = 2^p - 1$. Also $n = 2^{p-1}(2^p - 1)$, wegen $n > 6$ ist $p > 2$, eine ungerade Primzahl (M_p prim $\Rightarrow p$ prim). Da wir Reste mod 9 betrachten, spielt für die Exponenten nur Kongruenz mod $\varphi(9) = 3^2 - 3 = 6$ eine Rolle, wir machen eine Tabelle: für $n = (2^p - 1)2^{p-1} \equiv (2^{p \pmod 6} - 1)2^{(p-1) \pmod 6} \equiv x \pmod 9$

$p \pmod 6$	$p-1 \pmod 6$	x
1	0	$(2^1 - 1)2^0 \equiv 1 \pmod 9$
3	2	$(2^3 - 1)2^2 = 7 \cdot 4 \equiv 1 \pmod 9$
5	4	$(2^5 - 1)2^4 = 31 \cdot 16 \equiv 4 \cdot 7 \equiv 1 \pmod 9$

Jedesmal ist $n \equiv 1 \pmod 9$, n hat also auch die iterierte Quersumme 1.

Aufgabe 4:

```
/* Pari Brute-Force:
  ellgens bringt nur einen rationalen Punkt, keine Lösung in ZZ
*/

f(x)=1/6*x*(x+1)*(2*x+1);


{
   for(x=1,1000,
     t=f(x);
     if(issquare(t),
       print(" Höhe der Pyramide:",x,", Quadratseite: ",sqrtint(t), ", Anzahl der Kugeln: ", t);
     );
   );
}

/*
 Höhe der Pyramide:1, Quadratseite: 1, Anzahl der Kugeln: 1
 Höhe der Pyramide:24, Quadratseite: 70, Anzahl der Kugeln: 4900
*/
```

Aufgabe 2:

```
suche(N = 3000)=
{ local(gefunden=0,
        Summe= 0,
        aktuellePrimzahlNr= 0
  );

  while(gefunden < N,
    P = prime(aktuellePrimzahlNr++);
    P2 = 2*P+1;
    if(isprime(P2),
      Summe+=P;
      gefunden++;
      print("[ ",P,", ",P2," ]");
    );
  );

  print("\nSumme: ", Summe);
}

suche();

\\ Summe: 427659091
```

Aufgabe 3:

```
lucasLehmerTest(p) =
/*
    Input
      p: an odd prime
    Output
      1 (true) if M_p=2^p-1 is a prime,
      0 otherwise.
*/
{
  local(m,s);
  m = 2^p-1;
  s = 4;
  for( n = 2, p-1, s = (s*s - 2)%m);
  return( 0 == s);
}

suche(N=17)=
{ local(gefunden=0,
        aktuellePrimzahlNr= 1
  );

  print(gefunden++,": M_",2);\\, "= ", 2^P-1); \\ 2 ist nicht ungerade, aber Mersenne-PZ
  while(gefunden<N,

    P=prime(aktuellePrimzahlNr++);
    if( lucasLehmerTest(P),
        gefunden++;
        print(gefunden,": M_",P);\\, "= ", 2^P-1);
    );
  );
}


suche();
/*
```

```
1: M_2
2: M_3
3: M_5
4: M_7
5: M_13
6: M_17
7: M_19
8: M_31
9: M_61
10: M_89
11: M_107
12: M_127
13: M_521
14: M_607
15: M_1279
16: M_2203
17: M_2281
*/
```