Aufgabe 1

In [1]:
```
111111^2%100003
```
Out[1]: 83965

In [2]:
```
gcd(12345,54321)
```
Out[2]: 3

In [3]:
```
next_prime(1000)
```
Out[3]: 1009

In [128]:
```
for i in range(1,1003):
    if (i^2)%1003==19:
        print(i)
```
```
45
368
635
958
```

In [55]:
```
l=[]
for i in range(1,101):
    a=i^2%101
    if not a in l:
        l.append(a)
len(l)
```
Out[55]: 50

In [56]:
```
l=[]
for i in range(1,101):
    a=2^i%101
    if not a in l:
        l.append(a)
len(l)
```
Out[56]: 100

Teil 7 und 8 sind etwas knifflig. Was naheliegt, aber nicht klappt:

In [1]:
```
solve(u^2==u,u)
```
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-f4915ba74dc0> in <module>()
----> 1 solve(u**Integer(2)==u,u)

NameError: name 'u' is not defined
```

So klappt es: erst u als Variable deklarieren

In [2]:
```
u=var('u')
```

In [3]:
```
1 solve(u^2==2,u)
```

Out[3]:
```
[u == -sqrt(2), u == sqrt(2)]
```

In [5]:
```
1 l=solve(u^3+u==1,u)
2 l
```

Out[5]:
```
[u == -1/2*(1/18*sqrt(31)*sqrt(3) + 1/2)^(1/3)*(I*sqrt(3) + 1) + 1/6*(-I*sqrt(3)
+ 1)/(1/18*sqrt(31)*sqrt(3) + 1/2)^(1/3), u == -1/2*(1/18*sqrt(31)*sqrt(3) + 1/
2)^(1/3)*(-I*sqrt(3) + 1) + 1/6*(I*sqrt(3) + 1)/(1/18*sqrt(31)*sqrt(3) + 1/2)^(1
/3), u == (1/18*sqrt(31)*sqrt(3) + 1/2)^(1/3) - 1/3/(1/18*sqrt(31)*sqrt(3) + 1/
2)^(1/3)]
```

l ist eine Liste von Gleichungen! rhs extrahiert die rechte Seite (right hand side) der Gleichung

In [9]:
```
1 for i in range(3):
2     print(n(l[i].rhs()))
```

```
-0.341163901914010 - 1.16154139999725*I
-0.341163901914010 + 1.16154139999725*I
0.682327803828019
```

In [12]:
```
1 n(l[2].rhs(),digits=20)       # n für numerischer Wert
```

Out[12]:
```
0.68232780382801932737
```

A2 Faktorisieren

In [4]:
```
1  1111111111111111111111111111*1111111111111111111111111111111
```

Out[4]:
```
123456790123456790123456790122209876543209876543209876543 21
```

In [5]:
```
1 factor(123456790123456790123456790200209876543209876543209876543 21)
```

Out[5]:
```
11111111111111111111111111117 * 111111111111111111111111111213
```

In [92]:
```
1 factor(123456790123456790123456790123456790124331987654320987654320987654320 9
```

Das erste faktorisieren dauert bei mir nur ein paar Sekunden, das zweite eine Stunde. Daher die Werte hier so:

In [7]:
```
1 a=111111111111111111111111111111111111111189
2 b=111111111111111111111111111111111111111199
```

In [8]:
```
1 a*b
```

Out[8]:
```
123456790123456790123456790123456790124331987654320987654320987654320987654320 9
611
```

A3 Buchstaben zu Zahlen 0..25 und Vigenerecode

Hier wird zuerst das Ver- und Entschlüsseln mit bekanntem k gezeigt.

In [45]:
```
s='techfakstudierende'
k='lambda'
l=list(s)
l1=[ord(a)-97 for a in l]
kk=[ord(a)-97 for a in k]
l1
```

Out[45]: [19, 4, 2, 7, 5, 0, 10, 18, 19, 20, 3, 8, 4, 17, 4, 13, 3, 4]

In [46]:
```
lk=[]
for i in range(len(l1)):
    lk.append((l1[i]+kk[i%len(k)])%26)
lk
```

Out[46]: [4, 4, 14, 8, 8, 0, 21, 18, 5, 21, 6, 8, 15, 17, 16, 14, 6, 4]

In [47]:
```
lc=[chr(i+97) for i in lk]
text=''
for s in lc:
    text+=s
text
```

Out[47]: 'eeoiiavsfvgiprqoge'

In [48]:
```
w='TNFFOLTGQOWEDYEUHMP'
lw=list(w)
l2=[ord(a)-65 for a in lw]
l2
```

Out[48]: [19, 13, 5, 5, 14, 11, 19, 6, 16, 14, 22, 4, 3, 24, 4, 20, 7, 12, 15]

In [49]:
```
ll=[]
for i in range(len(l2)):
    ll.append((l2[i]-kk[i%len(k)])%26)
ll
```

Out[49]: [8, 13, 19, 4, 11, 11, 8, 6, 4, 13, 19, 4, 18, 24, 18, 19, 4, 12, 4]

In [50]:
```
lc=[chr(i+97) for i in ll]
text=''
for s in lc:
    text+=s
text
```

Out[50]: 'intelligentesysteme'

Den Schlüssel findet man, indem man die Liste l1 des Klartexts von der Liste lk des vershlüsselten Worts abzieht:

In [51]:
```
l3=[]
for i in range(len(l1)):
    l3.append((lk[i]-l1[i])%26)
l3
```

Out[51]: [11, 0, 12, 1, 3, 0, 11, 0, 12, 1, 3, 0, 11, 0, 12, 1, 3, 0]

In [52]:
```
lc=[chr(i+97) for i in l3]
text=''
for s in lc:
    text+=s
text
```

Out[52]: 'lambdalambdalambda'

In [ ]:
```
1
```