

lsg03

Aufgabe 1

reset ()

Set([s for s in Set([1..6]).subsets() if 1 in s and 4 in s])

```
{1, 2, 4, 6}, {1, 2, 3, 4}, {1, 3, 4, 5, 6}, {1, 4, 5, 6}, {1, 2, 4}, {1, 4, 5}, {1, 3, 4}, {1, 2, 3, 4, 6}, {1, 2, 3, 4, 5, 6}, {1, 2, 4, 5, 6}, {1, 3, 4, 5}, {1, 3, 4, 6}, {1, 2, 3, 4, 5}, {1, 4}, {1, 2, 4, 5}, {1, 4, 6}}
```

Aufgabe 2

reset()

pi.n(digits=100)

```
3.141592653589793238462643383279502884197169399375105820974944592307\
816406286208998628034825342117068
```

(e^2).n(digits=100)

```
7.389056098930650227230427460575007813180315570551847324087127822522\
573796079057763384312485079121795
```

log(3, base=2).n(digits=100)

```
1.584962500721156181453738943947816508759814407692481060455752654541\
098227794358562522280474918088242
```

(5^(1/3)).n(digits=100)

```
1.709975946676696989353108872543860109868055110543054924382861707444\
295920504173216257187010020189002
```

Aufgabe 3

reset()

var('a,b,c,d,f,g')

M1 = Set([1,2,3,Set([c,4]),a,b])

M2 = Set([6,7,8,a,c,d])

M3 = Set([1,2,a,b,f,g])

M2 = M2.union(Set([Set([2,3])])); M2

```
{{2, 3}, c, a, 6, 7, 8, d}
```

```
M1.union(M2).union(M3)
```

```
{1, f, 3, c, a, 6, 7, 8, 2, b, {2, 3}, g, d, {c, 4}}
```

```
M1.intersection(M2).intersection(M3)
```

```
{a}
```

```
M1.difference(M2)
```

```
{1, 2, 3, b, {c, 4}}
```

```
Set([n for n in M1.subsets() if (2 in n) and (b in n)])
```

```
{{2, 3, b}, {1, 2, 3, b, a}, {2, b, {c, 4}}, {1, 2, 3, b, {c, 4}},  
{2, 3, b, a, {c, 4}}, {2, b}, {a, 2, b, {c, 4}}, {1, 2, b, {c, 4}},  
{1, 2, b}, {1, 2, b, a}, {2, 3, b, a}, {2, 3, b, {c, 4}}, {1, 2, 3,  
a, b, {c, 4}}, {a, 1, 2, b, {c, 4}}, {1, 2, 3, b}, {2, b, a}}
```

Aufgabe 4

```
(0.304627*10^6).str(5)
```

```
'3.422200200000000000000000e7'
```

```
Zp(5,print_mode='digits')(0.304627*10^6)
```

```
...34222002
```

```
(0.304627*10^6).str(5)
```

```
'3.422200200000000000000000e7'
```

```
RR(".152005",6) * 6^6
```

```
14693.0000000000
```

```
Zp(11)(RR(".123456",7) * 7^6)
```

```
6 + 2*11^2 + 6*11^3 + 11^4 + 0(11^20)
```

```
Zp(11,print_mode='digits')(RR(".123456",7) * 7^6)
```

```
...16206
```

```
(RR(".123456",7)*7^6).str(11)
```

```
'16206.0000000000000'
```

Aufgabe 5

```
reset()
f(x) = (2*x^2 - 3*x)/(x - 4)

sol = solve(f.diff(x) == 0, x); sol

[x == -sqrt(10) + 4, x == sqrt(10) + 4]

[f.diff(x,2).subs(s).n() for s in sol]

[-1.26491106406735, 1.26491106406735]
```

Aufgabe 6

```
reset()
a = 1-2*i
b = (3+4*i)/(3*i)
c = 7-6*i
d = a*b
e = a*c/b
f = b^(a*c)

real(d), real(e), real(f)

(-2/3, 24/5, 243/3125*e^(-20*arctan(3/4))*cos(-20*log(5/3) +
5*arctan(3/4)))

real(d).n(), real(e).n(), real(f).n()

(-0.666666666666667, 4.80000000000000, 1.51031364525416e-7)

imag(d), imag(e), imag(f)

(-11/3, -57/5, 243/3125*e^(-20*arctan(3/4))*sin(-20*log(5/3) +
5*arctan(3/4)))

imag(d).n(), imag(e).n(), imag(f).n()

(-3.66666666666667, -11.4000000000000, -1.31352320039263e-7)

abs(a), arg(a)
```

```
(sqrt(5), -1.10714871779)
```

```
abs(b), arg(b)
```

```
(5/3, -0.643501108793)
```

```
abs(c), arg(c)
```

```
(sqrt(85), -0.708626272128)
```

Aufgabe 7

```
reset()
```

```
sum([sum([1/(i+j) for j in [1..i]]) for i in [1..100]])
```

```
22974356612913161309176081725476566403717260337055814733283689123494\
747952959095049293897433/3372935888329262646394657667948414074323943\
82785157234228847021917234018060677390066992000
```

```
float(sum([sum([1/(i+j) for j in [1..i]]) for i in [1..100]]))
```

```
68.113825384013424
```

Aufgabe 8

```
(10^50/3.0).n(digits=100)
```

```
3.333333333333333760735556654223821060848710005555200000000000000000\
000000000000000000000000000000000000000000000000000000000000000000e49
```

```
10^50/(3.n(digits=100))
```

```
3.333333333333333333333333333333333333333333333333333333333333333333\
3333333333333333333333333333333333333333333333333333333333333333333e49
```

```
(10^50/3).n(digits=100)
```

```
3.333333333333333333333333333333333333333333333333333333333333333333\
3333333333333333333333333333333333333333333333333333333333333333333e49
```

```
(10^50).n(digits=100)/3
```

```
3.333333333333333333333333333333333333333333333333333333333333333333\
3333333333333333333333333333333333333333333333333333333333333333333e49
```

```
(10^50).n(digits=100)/3.0
```

3.33333333333333e49

Aufgabe 10

```
reset()
def f(x):
    if x < 900:
        return f(x+1)
f(0)
```