

```
> w:=proc(n) option remember; if n=0 then 2
else normal((w(n-1)+2/w(n-1))/2) end if;
end;
```

```
> seq(w(j), j=1..6);
```

```
3 17 577 665857 886731088897
-----
2' 12' 408' 470832' 627013566048'
1572584048032918633353217
-----
1111984844349868137938112
```

```
> seq(evalf(w(j)), j=1..6);
```

```
1.500000000, 1.416666667, 1.414215686, 1.414213562,
1.414213562, 1.414213562
```

```
> for j from 0 to 6 do w(j)^2-2 end do;
```

```
2
1
4
1
144
1
166464
1
-----
221682772224
```

1  

---

393146012008229658338304

1  

---

1236510294063800469693771621893337765354742124544

```
> for j from 0 while w(j)^2-2>10^(-50) do  
  print(j, evalf(w(j)^2-2)) end do;
```

0, 2.

1, .2500000000

2, .006944444444

3, .6007304883 10<sup>-5</sup>

4, .4510950445 10<sup>-11</sup>

5, .2543584240 10<sup>-23</sup>

6, .8087275980 10<sup>-48</sup>

```
> arch:=proc(n) option remember; if n=1 then  
  [4*sqrt(3), 6] else  
  [2*arch(n-1)[1]*arch(n-1)[2]/(arch(n-1)[1]+  
  arch(n-1)[2]), sqrt(2*arch(n-1)[1])*arch(n-1)  
  ) [2]/sqrt(arch(n-1)[1]+arch(n-1)[2])] end  
if end;
```

```
> arch := proc (n) option remember; if n = 1  
  then [4*sqrt(3), 6] else  
  [2*arch(n-1)[1]*arch(n-1)[2]/(arch(n-1)[1]+  
  arch(n-1)[2]),  
  sqrt(2*arch(n-1)[1])*arch(n-1)[2]/sqrt(arch
```

```
(n-1) [1]+arch(n-1) [2]]) end if end proc;
```

```
> seq(arch(n), n=1..3);
```

$$[4\sqrt{3}, 6], \left[ 48 \frac{\sqrt{3}}{4\sqrt{3}+6}, 12 \frac{\sqrt{2} 3^{1/4}}{\sqrt{4\sqrt{3}+6}} \right],$$

$$\frac{1152 \frac{3^{3/4} \sqrt{2}}{(4\sqrt{3}+6)^{3/2} \left( 48 \frac{\sqrt{3}}{4\sqrt{3}+6} + \frac{12\sqrt{2} 3^{1/4}}{\sqrt{4\sqrt{3}+6}} \right)}}{24 \frac{\sqrt{6} \sqrt{\frac{\sqrt{3}}{4\sqrt{3}+6}} \sqrt{2} 3^{1/4}}{\sqrt{4\sqrt{3}+6} \sqrt{12 \frac{\sqrt{3}}{4\sqrt{3}+6} + \frac{3\sqrt{2} 3^{1/4}}{\sqrt{4\sqrt{3}+6}}}}}$$

```
> seq(evalf(arch(n)), n=1..5);
```

```
[6.928203232, 6.], [6.430780622, 6.211657081],
```

```
[6.319319882, 6.265257228], [6.292172433, 6.278700401],
```

```
[6.285429200, 6.282063907]
```

```
> evalf(arch(8));
```

```
[6.283220347, 6.283167778]
```

```
> f:=n-> if type(n,even)=true then n/2 else 3*n+1 end if;
```

```
[ > seq((f@@j)(7), j=0..16);
```

```
7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1
```

```
[ >
```

```
[ > print('Startwerte mit Rekordzahl fuer das  
Maximum im Orbit'); maxelem:=0: for n from  
1 to 1000 by 2 do a[0]:=n: for k from 1  
while(a[k-1]>1) do a[k]:=f(a[k-1]): end do:  
;if(max(seq(a[j], j=1..k-1))>maxelem)then  
maxelem:=max(seq(a[j], j=1..k-1));print(n, ma  
xelem); end if  
end do:
```

*Startwerte mit Rekordzahl fuer das Maximum im Orbit*

3, 16

7, 52

15, 160

27, 9232

255, 13120

447, 39364

639, 41524

703, 250504

```
[ > G:=proc(j) local n; for n from 1 while  
(f@@n)(j)>1 do end do; n end;
```

```
[ > G(7);
```

16

```
> print('Startwerte mit Rekordzahl fuer das
Maximum im Orbit');maxelem:=0:for n from 1
to 100 by 2 do
if(max(seq((f@@j)(n),j=1..G(n)))>maxelem)th
en
maxelem:=max(seq((f@@j)(n),j=1..G(n)));prin
t(n,maxelem); end if
end do:
```

>

*Startwerte mit Rekordzahl fuer das Maximum im Orbit*

1, 4

3, 16

7, 52

15, 160

27, 9232

```
> g:=n-> if modp(n,3)=0 then 2*n/3 elif
modp(n,3)=1 then (4*n-1)/3 else (4*n+1)/3
end if;
```

```
> g(30);
```

20

```
> g(31);
```

41

```
> g(32);
```

43

```
> g(1);
```

