

Procedure to bubble sort a list of numbers

```
> bubble:=proc(v)
>   local n,i,j,t,w:
>   n:=nops(v): w:=v:
>   for i from 1 to n-1 do
>     for j from i+1 to n do
>       if w[i]>w[j] then
>         t:=w[i]: w[i]:=w[j]: w[j]:=t: fi:
>       od:
>     od:
>   return w; end:
> bubble([3,7,10,4,12,1,7]);
```

[1, 3, 4, 7, 7, 10, 12]

(1)

Generate the first 30 Fibonacci numbers

```
> F:=array(0..30):
> F[0]:=0: F[1]:=1:
> for n from 2 to 30 do F[n]:=F[n-1]+F[n-2]: od:
> seq(F[n],n=0..30);
```

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040

(2)

Generate all bitstrings of length  $n$ . (Do not try this if  $n$  is really large!)

```
> with(StringTools):
> S:=Generate(5,"01");
S := ["00000", "00001", "00010", "00011", "00100", "00101", "00110", "00111", "01000", "01001", "01010", "01011", "01100",
      "01101", "01110", "01111", "10000", "10001", "10010", "10011", "10100", "10101", "10110", "10111", "11000", "11001",
      "11010", "11011", "11100", "11101", "11110", "11111"]
> nops(S);
```

(3)

32

(4)

Now list just the 11-free bitstrings of length  $n$ .

```
> A:={}:
> for w in S do
>   if nops({SearchAll("11",w)})=0 then A:=A union {w}: fi: od:
> A;
```

{"00000", "00001", "00010", "00100", "00101", "01000", "01001", "01010", "10000", "10001", "10010", "10100", "10101"}

(5)

```
|> nops (A) ;
```

13

(6