

1 Для

Факториал числа:

```
Seq
  Read n
  Seq
    Assign ans 1
  Seq
    While > n 0
      Seq
        Assign ans * ans n
        Assign n - n 1
      Write ans
```

Фибоначчи:

```
Seq
  Read n
  Seq
    Assign f 0
  Seq
    Assign s 1
  Seq
    While > n 0
      Seq
        Assign t + f s
      Seq
        Assign f s
      Seq
        Assign s t
        Assign n - n 1
      Write f
```

Нашел маленький баг в документации. Там было написано, что Read нужно пользоваться в виде

```
V0 Read
```

Хотя на самом деле надо пользоваться так:

```
Read V0
```

Язык очень удобный, потому что не надо ставить всякие вспомогательные ключевые слова типа фигурных скобок. Но следствием из этого является проблема с Seq. Неклассно его везде писать. И Nor — весьма странный костыль. Могут быть проблемы с написанием больших арифметических выражениях из-за польской записи. В остальном... Если ставить отступы, то очень удобно писать на языке.

2 Олень

Факториал числа:

```
./
  read n;
  bind ans (1);
  while (n>0)
  ./
    bind ans (ans*n);
    bind n (n-1);
  \.;
  write (ans);
\.
```

Фибоначчи:

```
./
  read n;
  bind f (0);
  bind s (1);
  while (n>0)
  ./
    bind t (ans+ans');
    bind f (s);
    bind s (t);
    bind n (n-1);
  \.;
  write (f);
\.
```

Из-за ограничения на длину названия переменных наверняка будут проблемы с тем, что названия переменных — непонятный случайный набор букв. Использование странных наборов символов вместо фигурных скобок выглядит сомнительным решением, потому что надо писать два символа, но okay... В выражениях нельзя ставить пробелы :(Это очень грустно. В остальном все замечательно. Писать точно удобнее, чем на предыдущем языке, потому что не надо бесконечно думать о Seq'ах.