

Lista 4 z list Logo

Zad. 1. Co wyczyniają te programiki? (np po podaniu listy [1 2 7 3 5])

a)

```
TO mmin :L
  IF 1=COUNT :L [OP FIRST :L ]
  IF 1<COUNT :L [IF (FIRST :L) < (mmin BF :L) [OP FIRST :L]
                    IF NOT ((FIRST :L) < (mmin BF :L)) [OP mmin BF :L] ]
END

TO mmin1 :L
  IF 1=COUNT :L [OP FIRST :L STOP]
  IF (FIRST :L) < (mmin1 BF :L) [OP FIRST :L STOP]
  OP mmin1 BF :L
END

TO fmmin :L
  IF 1=COUNT :L [OP :L ]
  IF 1<COUNT :L [IF (FIRST :L) < (mmin BF :L) [OP :L ]
                    IF NOT ((FIRST :L) < (mmin BF :L)) [OP LPUT (FIRST :L) fmmin BF :L]]
END

b)
```

```
TO upo :L
  IF 1=COUNT :L [OP :L ]
  IF 1<COUNT :L [IF (FIRST :L) < (FIRST upo BF :L) [OP FPUT (FIRST :L) upo BF :L]
                    IF NOT((FIRST :L) < (FIRST upo BF :L)) [
                      OP FPUT (FIRST upo BF :L) upo FPUT (FIRST :L) (BF upo BF :L)]
                    ]
END

TO upo1 :L
  IF 1=COUNT :L [OP :L STOP]
  IF (FIRST :L) < (FIRST upo1 BF :L) [OP FPUT (FIRST :L) upo1 BF :L STOP]
  OP FPUT (FIRST upo1 BF :L) upo1 FPUT (FIRST :L) (BF upo1 BF :L)
END

TO upo2 :L
  IF 1=COUNT :L [OP :L STOP]
  MAKE "tym upo2 BF :L
  IF (FIRST :L) < (FIRST :tym) [OP FPUT (FIRST :L) :tym STOP]
  OP FPUT (FIRST :tym) upo2 FPUT (FIRST :L) BF :tym
END

c)
```

```
TO fib :n
  IF 3>:n [OP 1 STOP]
  OP (fib :n-1 ) + fib :n-2
END

TO fib2 :n
  IF 1=:n [OP [0 1] STOP]
  IF 2=:n [OP [1 1] STOP]
  MAKE "tym fib2 :n-2
  OP LPUT (FIRST :tym) + 2 * (LAST :tym) FPUT (FIRST :tym) + (LAST :tym) []
END
```

Zad. 2. Napisz funkcję obliczającą n -ty wyraz ciągu:

- a)** $a_n = 2 a_{n-1} + 3 a_{n-2}$ dla $n > 2$ i $a_2 = 3, a_1 = 1$
b) $b_n = 3 b_{n-1} + 4 b_{n-2} + 17$ dla $n > 2$ i $b_2 = -2, b_1 = 3$
c) $c_n = 3 c_{n-1} + 2 c_{n-2} + 1 c_{n-3}$ dla $n > 3$ i $c_3 = 4, c_2 = -2, c_1 = 3$

Zrób to **też** analogicznie do przykładu fib2