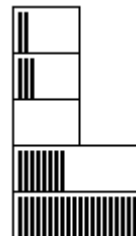


## Lista 6 z list Logo

```
Make "reg [[20 20] [8 20] [0 10] [3 10] [2 10]]
```

Pod zmienną reg zakodowany jest stan regału taki, jak na rysunku obok, czyli ma 5 półek, czwarta od dołu ma limit 10 i są na niej trzy sztuki towaru.



**Ćw. 0.** Napisz procedurę rys :L, która rysuje taki lub podobny rysunek (po wywołaniu rys :reg )

**Ćw. 1.** Uzupełnij tak, by funkcje liczyły: stan - liczbę towarów i pojemn - pojemność regału :L

```
TO stan :L                                TO pojemn :L
  IF EMPTY? :L [OP 0 STOP]                IF EMPTY? :L [OP 0 STOP]
  . . .                                    . . .
END                                          END
```

By zobaczyć aktualny stan regału, należy wywołać stan :reg

Funkcja zmian\_og :i :cosik :L nic nie zmienia tylko zwraca listę z podmienionym i-tym elementem.

```
TO zmian_og :i :cosik :L
  IF :i=1 [OP FPUT :cosik (BF :L) STOP]
  OP FPUT (FIRST :L) (zmian_og :i-1 :cosik BF :L)
END
```

**Ćw. 2.** Podaj co wydrukuje LOGO po podaniu:

- a) show zmian\_og 2 7 [1 3 4 8]      b) show zmian\_og 2 7 [[1 3] 4 8]  
c) show zmian\_og 3 2 [1 3 [4 2 8]]      d) show zmian\_og 3 [2 3] [[1 2] 4 8 9]

Poniższa procedura zmienia zmienną reg (ale dopiero po wywołaniu!).

```
TO zmiana :i :j :cosik
  MAKE "reg zmian_og :i (zmian_og :j :cosik (ITEM :i :reg)) :reg
END
```

**Ćw. 3.** Poniższa procedura zdejmie jedną sztukę towaru. Z której półki?

```
TO odej1 :L
  IF 0 = stan :L [show "oj_nie_da_sie STOP]
  IF (FIRST (LAST :L)) > 0 [zmiana (COUNT :L) 1 (FIRST (LAST :L))-1 STOP]
  odej1 BL :L
END
```

Jaka jest po wywołaniu odej1 :reg wartość :reg gdy :reg miała wartość:

- a) [[2 20] [8 20] [0 10] [3 10] [5 10]]      b) [[20 20] [8 20] [3 10] [2 10] [10 10]]  
c) [[8 10] [0 4] [3 4] [0 4]]      d) [[0 5] [8 20] [3 10] [0 10] [0 10]]

**Ćw. 4.** Napisz podobną do powyższej procedurę, która wstawia na regał jedną sztukę towaru na najwyższej półce

```
TO dodaj1 :L
```

```
END
```

**Ćw. 5.** Napisz podobną do powyższej procedurę, która dodaje jedną sztukę towaru na najniższej półce, na której to możliwe

```
TO dodaj1_nadole :L
```

```
. . .
. . .
END
```

**Ćw. 6.** Co wyczynia procedura uporz ???

```
TO czy_uporz :L
  IF EMPTY? :L [OP 1=1 STOP]
  IF (FIRST (FIRST :L)) = (LAST (FIRST :L)) [OP czy_uporz BF :L STOP]
  OP 0 = stan BF :L
END
```

```
TO uporz
```

```
IF NOT czy_uporz :reg [odej1 :reg      dodaj1_nadole :reg      uporz ]
END
```