

PODSTAWY PROGRAMOWANIA DEKLARATYWNEGO

PROLOG

Ćwiczenia 12

Zadanie 1. (2p.)

Zdefiniować procedurę, która dla dwóch list takiej samej długości tworzy nową w taki sposób, że element na i -tym miejscu jest sumą i -tych elementów list składowych.

```
?-sm([1,2,3,4],[3,4,5,6],X).  
X=[4,6,8,10].
```

Zadanie 2. (2p.)

Zdefiniować procedurę, która sprawdza, czy dwie listy mają taką samą liczbę elementów - na dwa sposoby:

- a) z wykorzystaniem predykatu *length*,
- b) bez użycia *length*.

```
?-rowne([a,b,c],[2,3,4]).  
true.  
?-rowne([], [1,2,3,4]).  
false.  
?-rowne([x,y,z,x],[0]).  
false.
```

Zadanie 3. (2p.)

Zdefiniować procedurę znajdującą największą wartość w liście liczbowej.

```
?-max([1,4,2,7,3,0],B).  
B=7.
```

Zadanie 4. (2p.)

Zdefiniować procedurę sprawdzającą, czy podana lista stanowi początek innej (podanej jako drugi argument) listy.

```
?-początek([1,3], [1,3,2,4,3]).  
true.  
?-początek([3,1], [1,3,2,4,3]).  
false.
```

Zadanie 5. (11p.)

Napisać procedury (nazwy procedur własne) działające jak procedury wbudowane:

- a) **permutation**(L1,L2), (2p.)
- b) **flatten**(L1,L2), (2p.)
- c) **sumlist**(L,S), (1p.)
- d) **numlist**(M,N,L), (2p.)
- e) **sort**(L1,L2), (2p.)
- f) **msort**(L1,L2). (2p.)

Zadanie 6. (6p.)

Zdefiniować procedury **iloczyn**, **suma**, **roznica** działające jak wbudowane intersection, union, subtract (odpowiednio: iloczyn zbiorów, suma zbiorów, różnica zbiorów) .

Przykład.

1 ?- intersection([1,2,3,4],[6,3,7,2,5],I).

I = [2, 3].

2 ?- union([1,2,3,4],[6,3,7,2,5],U).

U = [1, 4, 6, 3, 7, 2, 5].

3 ?- subtract([1,2,3,4],[6,3,7,2,5],S).

S = [1, 4].

Uwaga:

Wykonane zadania należy przekazać do **26.05.2019, 23:59** przez OLAT „Ćwiczenia 12-pn”.
Nazwa pliku ma zawierać nazwisko Studenta i numer ćwiczeń.