

1. Rozhodněte, zda je možné následující výrazy unifikovat; pokud ano, napište nějakou možnou unifikaci (ve formě přiřazení proměnných):

- (a) $p(p(A,B),A)$ a $p(X,p(Y,2))$
- (b) $[X, [2|X] | [X|X]]$ a $[A, [B] | C]$
- (c) $[X,Y,Y | [Y,Z]]$ a $[A | [3 | [B,4|C]]]$
- (d) $[a,b,c|X]-X$ a $[a,b,C,D,E]-[d]$
- (e) $[a,X,X|X]-X$ a $[a,A,Y,d]-Y$

(0,5 bodu)

2. Mějme dána fakta v databázi:

```
rodic(a,b). rodic(a,c). rodic(d,c). rodic(e,f). rodic(c,f).
rodic(c,g). rodic(f,h). rodic(f,i).
```

a tento predikát:

```
potomek(X,Y) :- rodic(Y,X).
potomek(X,Y) :- potomek(X,Z), rodic(Y,Z).
```

Napište, jak bude prolog postupně odpovídat při zadání následujícího dotazu a postupném vyžadování dalších odpovědí pomocí ‘;’

```
potomek(X,c).
```

(0,5 bodu)

3. Napište predikát `flatten(+Vstup,-Vystup)`, který „zploští“ předaný seznam. `Vstup` je seznam, který může obsahovat vnořené seznamy; `Vystup` bude obsahovat tytéž prvky ve stejném pořadí, ale bez vnořených seznamů. Příklad:

```
?- flatten([1,[a,b,[4]],5,[],6], X).
X = [1,a,b,4,5,6];
false.
```

(1 bod)

Bonus: další bod získáte, pokud bude predikát fungovat v lineárním čase s velikostí vstupu (tj. počet prvků plus počet vnořených seznamů).

4. Napište predikát `diff_reverse(+Vstup,-Vystup)`, který obrátí pořadí prvků v rozdílovém seznamu na vstupu až na konec vyznačený za pomlčkou, který zachová (ať je to volná proměnná nebo už konkrétní seznam). Není-li na vstupu korektní rozdílový seznam, měl by program vrátit `false`.

```
?- diff_reverse([a, b|X]-X, V).
V = [b, a|X]-X.
?- L=[a, b|X]-X, diff_reverse(L, M), X=[c].
L = [a, b, c]-[c], X = [c], M = [b, a, c]-[c].
?- L=[a, b, c]-[c], diff_reverse(L, M).
L = [a, b, c]-[c], M = [b, a, c]-[c].
?- diff_reverse([a, b|X]-Y, V).
false.
```

Program by měl fungovat v lineárním čase.

(1 bod)