

Zadanie 8.1

Do pliku **rodzina.pl** (patrz zadanie 7.1) dopisz reguły:

1. **siostra(X, Y)** - kończy się sukcesem, jeśli X jest siostrą Y;
2. **para(X, Y)** - kończy się sukcesem, jeżeli X jest kobietą, Y jest mężczyzną i mają wspólne dziecko;
3. (*) **brat(X, Y)** - kończy się sukcesem, jeśli X jest bratem Y;
4. (*) **rodzina(X, Y, Z1, Z2)** - kończy się sukcesem, jeżeli X jest kobietą, Y jest mężczyzną i Z1, Z2 są ich wspólnymi (różnymi) dziećmi.

UWAGA: Nie można być siostrą (bratem) samego siebie.

Zadanie 8.2

Do pliku **rodzina.pl** dopisz regułę **rodzice(X)**, która wypisuje informacje o rodzicach X wg. podanych przykładów:

```
?- rodzice(max).  
ojciec: tomek  
matka: ola  
Yes  
  
?- rodzice(lila).  
ojciec: tomek  
Yes  
  
?- rodzice(radek).  
matka: kasia  
Yes  
  
?- rodzice(ola).  
brak danych  
Yes
```

Zadanie 8.3

Do pliku **rodzina.pl** dopisz rekurencyjną regułę:

1. **potomek(X, Y)**, która kończy się sukcesem, jeśli X jest potomkiem Y.
2. (*) **potomkowie(X)**, która wypisuje na ekranie wszystkich potomków X (każdy w oddzielnej linii) i zawsze kończy się sukcesem. Należy skorzystać z predykatu fail do wyszukania wszystkich rozwiązań.

Zadanie 8.4

Do pliku **zabytki.pl** (patrz zadanie 7.3) dopisz reguły:

1. **zabytki_m(M)**, która wypisuje na ekranie wszystkie zabytki znajdujące się w mieście M (każdy w oddzielnej linii).
2. **zabytki_mp(M)**, która wypisuje na ekranie wszystkie zabytki znajdujące się w mieście M (każdy w oddzielnej linii, łącznie z nazwą miasta i państwa).
3. (*) **zabytki_p(P)**, która wypisuje na ekranie wszystkie zabytki znajdujące się w państwie P (każdy w oddzielnej linii).
4. (*) **zabytki_ps(P)**, która wypisuje na ekranie wszystkie zabytki znajdujące się w państwie P (każdy w oddzielnej linii, łącznie z nazwą miasta i państwa) i w państwach sąsiednich.

Uwaga: We wszystkich przypadkach reguły zawsze mają kończyć się sukcesem.

Zadanie 8.5

Napisz program obliczający wynik równania kwadratowego $ax^2 + bx + c = 0$ w dziedzinie liczb rzeczywistych. Zaimplementuj predykaty:

- **delta/4** – obliczający deltę, argumenty kolejno: a, b, c, wynik,
- **trojmian/4** – obliczający wynik równania kwadratowego, argumenty kolejno: a, b, c, wynik.

Uwaga: Zwróć uwagę na niedeterminizm w predykatcie **trojmian/4**, który znajduje zero, jedno, albo dwa rozwiązania.

Zadanie 8.6

(*) Zdefiniuj rekurencyjny predykat **seq/2**, wyznaczający wartość n-tego wyrazu ciągu liczbowego zdefiniowanego następująco:

- $seq_1 = 3$
- $seq_2 = 4$
- $seq_n = seq_n-1 + 3 * seq_n-2$, dla $n > 2$

Zadanie 8.7

Przeanalizuj poniższy kod. Jakie zadanie realizuje predykat **liczba/1**?

```
liczba(X) :-  
    between(100, 999, X),  
    X mod 3 == 0,  
    X mod 5 == 0,  
    X mod 7 == 0.
```

(*) Napisz program w Prologu rozwiązujący następującą zagadkę: *W skrzyni znajdują się złote monety. Jest ich mniej niż 100. Jeśli będą wyjmowane po 3, to w skrzyni zostanie 1. Jeśli będą wyjmowane po 4, to w skrzyni zostaną 2. Jeśli będą wyjmowane po 5, to w skrzyni zostaną 3. Jeśli będą wyjmowane po 6, to w skrzyni zostaną 4. Ile monet jest w skrzyni?*

Zadanie 8.8

(*) Napisz program w Prologu rozwiązujący następujące zadanie: W pewnej liczbie trzycyfrowej zamieniono cyfrę dziesiątek z cyfrą jedności, tworząc w ten sposób nową liczbę trzycyfrową. Suma obu liczb jest równa 1187. Znajdź te liczby. Czy rozwiązanie jest jednoznaczne?