Zadanie z Prologu (JPP 2023)

Majowe (i nie tylko) wyprawy turystyczne.

Napisać program, który wczyta z pliku o podanej (jako argument programu) nazwie informacje o dostępnych trasach turystycznych, a następnie wykona sekwencję poleceń podanych przez użytkownika (wprowadzonych z klawiatury).

Wykonanie każdego polecenia polega na znalezieniu wszystkich wypraw spełniających podane wymagania. Opis każdej znalezionej wyprawy powinien zawierać informacje o wszystkich miejscach, przez które prowadzi wyprawa, numerach kolejnych tras, ich rodzajach oraz długość całej wyprawy (patrz: przykład poniżej).

Program w trybie interaktywnym powinien zapytać o:
- miejsce początku wyprawy,
- miejsce końca wyprawy,
- warunki jakie ma spełniać wyprawa.

Każdą z podawanych informacji należy zakończyć kropką.

Miejsce początku i końca wyprawy: stała. Przyjmujemy, że `nil` oznacza dowolne miejsce, a `koniec` oznacza koniec pracy programu.

Warunki: sekwencja warunków, oddzielonych przecinkami, postaci:
- `rodzaj(R)` — trasa może być rodzaju `R`
- `dlugosc(war, k)`, gdzie `war = eq, lt, le, gt, ge`,
  `k` — liczba naturalna

Długość całej wyprawy ma spelniać podany warunek (tzn. ma być równa, mniejsza, mniejsza lub równa, większa, większa lub równa `k`).

Brak warunków (czyli pusta sekwencja): stała `nil`.

Można podać kilka rodzajów tras, jedna wyprawa może składać się z tras różnych rodzajów. Natomiast wolno podać co najwyżej jeden warunek na długość wyprawy.

Brak warunków oznacza, że podczas wyprawy można wybierać dowolne trasy.

W ramach jednej wycieczki każdą trasą można przejść co najwyżej raz (turysta nie chodzi w kółko tymi samymi trasami). Wyjątek stanowią trasy, którymi można przechodzić w dwóch kierunkach. Dozwolone jest zatem przejście dwa razy tą samą trasą, pod warunkiem, że przechodzimy nią w dwóch różnych kierunkach.

Program powinien sprawdzać poprawność danych wprowadzanych interaktywnie, przy założeniu, że podane dane są poprawnymi składniowo termami. Natomiast można założyć całkowitą poprawność danych w pliku.

Format danych: opis tras turystycznych.

Plik z opisami dostępnych tras jest zbiorem opisów tras, a opis każdej trasy jest prologowym termem zakończonym kropką.

Opis jednej trasy jest postaci:

```
trasa(Tid, Skąd, Dokąd, Rodzaj, Kierunek, IleKm).
```

gdzie

- `Tid` — jednoznaczny identyfikator trasy (stała)
- `Skąd` — początek trasy (stała)
- `Dokąd` — koniec trasy (stała)
- `Rodzaj` — rodzaj trasy (stała)
- `Kierunek` — informacja czy trasa jest dostępna w obu kierunkach.
  Stała `oba` oznacza, że można iść w obu kierunkach,
  a stała `jeden` że można iść tylko `Skąd Dokąd`.
- `IleKm` — długość trasy w km (liczba całkowita).

Przykładowe dane (plik tatry.txt)

```
trasa(r1, zakopane, brzeziny, rower, oba, 25).
trasa(r2, brzeziny, gasienicowa, rower, oba, 15).
trasa(r3, brzeziny, poroniec, rower, oba, 10).
trasa(r4, poroniec, rusinowa, rower, oba, 6).
trasa(g1, zakopane, kuznice, gorska, oba, 7).
trasa(g2, zakopane, kalatowki, gorska, oba, 5).
trasa(g3, kuznice, gasienicowa, gorska, oba, 7).
trasa(g4, gasienicowa, zawrat, gorska, oba, 6).
trasa(g5, gasienicowa, czarnystaw, gorska, oba, 3).
trasa(g6, zawrat, kozia, gorska, jeden, 5).
trasa(g7, kozia, gasienicowa, gorska, jeden, 7).
trasa(p1, zakopane, gubalowka, piesza, oba, 5).
```

Ważne uwagi dodatkowe

1. **Programy muszą poprawnie działać pod SICStus Prologiem na komputerze students. Programy niespełniające powyższego kryterium nie będą sprawdzane.**

2. W programie wolno korzystać ze standardowych predykatów Prologu używanych na ćwiczeniach (np. `member/2, append/3`) oraz z biblioteki o nazwie `lists` (operacje na listach).
   (Załadowanie biblioteki: na początku pliku źródłowego należy umieścić dyrektywę `:- ensure_loaded(library(lists))`.

3. Nie wolno korzystać z predykatów `findall/3 (bagof/3, setof/3)`, ani też z predykatów typu `assert, retract` itp.

4. W programie wolno (poprawnie) używać negacji, odcięcia, konstrukcji `if-then-else`, predykatu `if/3` itp.

5. Program powinien być czytelnie sformatowany, m.in. długość każdego wiersza nie powinna przekraczać 80 znaków. Sposób formatowania programów w Prologu (definicja algorytmu QuickSort):

```
qsort([], []).
qsort([X | L], S) :-      % komentarz niezasłaniający kodu
   partition(L, X, M, W),  % podział listy na podlisty
   qsort(M, SM),           % sortowanie podlist
   qsort(W, SW),
   append(SM, [X|SW], S).  % scalenie wyników
```

6. Program powinien zawierać (krótkie, zwięzłe) **komentarze** opisujące (deklaratywne) znaczenie ważniejszych predykatów oraz przyjęte (podstawowe) rozwiązania.

Przesłanie rozwiązania

Rozwiązanie zadania powinno składać się z jednego pliku o nazwie `<identyfikator_studenta>.pl` (np. `ab123456.pl`), który należy przesłać przez moodle'a. Pierwszy wiersz pliku powinien zawierać komentarz z nazwiskiem autora (anonimów nie czytamy ;).

Przykład działania programu

```
[user@students Prolog]$ wyprawy tatry.txt

 Podaj miejsce startu:  zakopane.
 Podaj miejsce koncowe: rusinowa.
 Podaj warunki : costam.
   Error: niepoprawny warunek - costam.
 Podaj warunki : nil.

zakopane -(r1,rower)-> brzeziny -(r3,rower)-> poroniec -(r4,rower)-> rusinowa
Dlugosc trasy: 41.

 Podaj miejsce startu:  zakopane.
 Podaj miejsce koncowe: nil.
 Podaj warunki : rodzaj(gorska), dlugosc(lt, 4), dlugosc(gt, 5).
  Error: za duzo warunkow na dlugosc.
 Podaj warunki : rodzaj(gorska), dlugosc(le, 9).

zakopane -(g1,gorska)-> kuznice
Dlugosc trasy: 7.

zakopane -(g2,gorska)-> kalatowki
Dlugosc trasy: 5.


 Podaj miejsce startu:  nil.
 Podaj miejsce koncowe: nil.
 Podaj warunki : rodzaj(piesza).

zakopane -(p1,piesza)-> gubalowka
Dlugosc trasy: 5.

gubalowka -(p1,piesza)-> zakopane
Dlugosc trasy: 5.

zakopane -(p1,piesza)-> gubalowka -(p1,piesza)-> zakopane
Dlugosc trasy: 10.

gubalowka -(p1,piesza)-> zakopane -(p1,piesza)-> gubalowka
Dlugosc trasy: 10.
```

```
 Podaj miejsce startu:  brzeziny.
 Podaj miejsce koncowe: nil.
 Podaj warunki : rodzaj(gorska), rodzaj(rower), dlugosc(lt,20).

brzeziny -(r2,rower)-> gasienicowa
Dlugosc trasy: 15.

brzeziny -(r3,rower)-> poroniec
Dlugosc trasy: 10.

brzeziny -(r2,rower)-> gasienicowa -(g5,gorska)-> czarnystaw
Dlugosc trasy: 18.

brzeziny -(r3,rower)-> poroniec -(r4,rower)-> rusinowa
Dlugosc trasy: 16.


 Podaj miejsce startu:  gasienicowa.
 Podaj miejsce koncowe: gasienicowa.
 Podaj warunki : rodzaj(gorska), dlugosc(lt,20).

gasienicowa -(g4,gorska)-> zawrat -(g6,gorska)-> kozia
            -(g7,gorska)-> gasienicowa
Dlugosc trasy: 18.

gasienicowa -(g4,gorska)-> zawrat -(g4,gorska)-> gasienicowa
Dlugosc trasy: 12.

gasienicowa -(g5,gorska)-> czarnystaw -(g5,gorska)-> gasienicowa
Dlugosc trasy: 6.

gasienicowa -(g3,gorska)-> kuznice -(g3,gorska)-> gasienicowa
Długosc trasy: 14.

 Podaj miejsce startu:  koniec.

Koniec programu. Milych wedrowek!
```