**BÀI TẬP THỰC HÀNH**

Môn Đặc tả hình thức

---

**Bài thực hành số 02: Xử lý danh sách (MẢNG) và CHUỖI trên PROLOG**

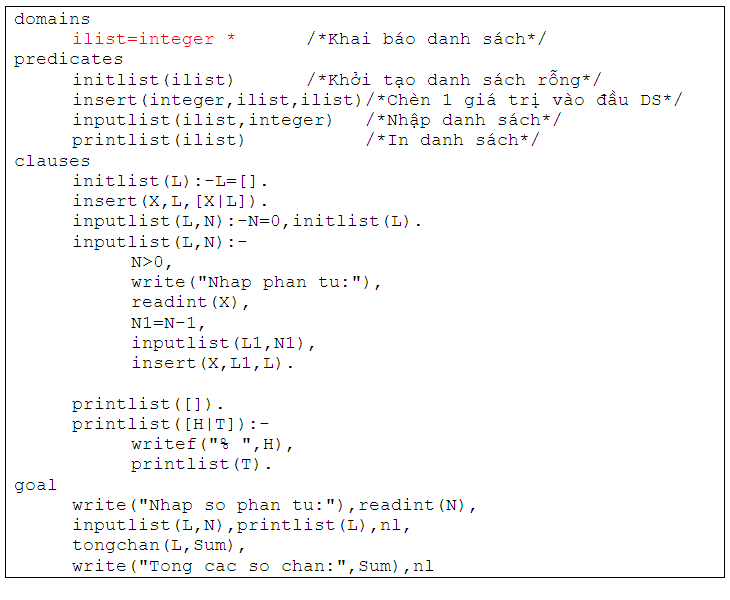
1. Khai báo danh sách, nhập và xuất danh sách:

Trong ngôn ngữ Prolog, danh sách tương đương với khái niệm mảng 1 chiều trong các ngôn ngữ lập trình khác. Một danh sách cũng giống như tập hợp, nhưng giá trị của các phần tử trong danh sách được quyền trùng nhau, trong khi tập hợp thì không.

Một danh sách có thể là danh sách rỗng hoặc chứa ít nhất 1 phần tử:

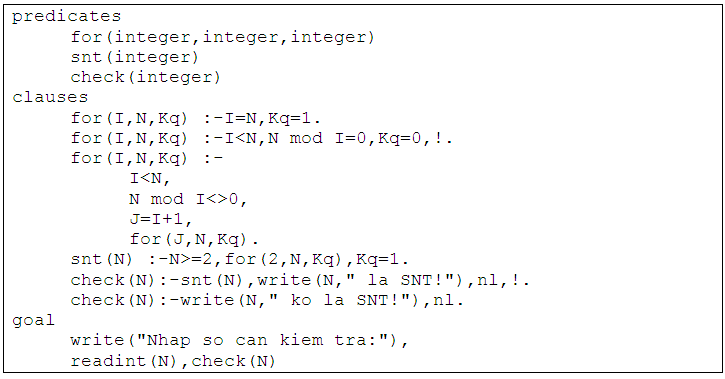
* Nếu danh sách là rỗng, ta ký hiệu là : [ ]
* Nếu danh sách có ít nhất 1 phần tử, ta xem danh sách đó bao gồm 2 phần: [H | T], trong đó H là phần tử đứng đầu danh sách, còn T là một danh sách con, chứa các phần tử còn lại trong danh sách.

Chương trình sau cho phép người dùng khai báo một danh sách, nhập giá trị cho N phần tử của danh sách và in kết quả vừa nhập ra màn hình.



**Ghi chú: Thử dùng lệnh write(L) thay cho printlist(L) và quan sát kết quả!**

1. Chương trình kiểm tra số nguyên tố (cải tiến):



1. Hãy sử dụng kiểu danh sách để viết các chương trình bằng Prolog thực hiện những công việc sau đây:
   1. Tính tổng các số chẵn có trong danh sách.
   2. Đếm số lượng số nguyên tố có trong danh sách.
   3. Tìm xem 1 giá trị X có trong danh sách hay không.
   4. Tìm giá trị lớn nhất của danh sách.
   5. Kiểm tra danh sách có gồm toàn những số chẵn hay không.
   6. Kiểm tra danh sách có thứ tự tăng hay không.
   7. Sắp xếp danh sách tăng dần theo phương pháp chèn (InsertionSort)
   8. Xóa tất cả các giá trị X khỏi danh sách.
   9. Tạo danh sách mới chứa tập hợp các phần tử của danh sách đã cho (giá trị các phần tử không lặp lại).
2. Chuỗi trong PROLOG được khai báo là kiểu string. Hằng chuỗi được biểu diễn trong cặp dấu ngoặc kép “”.

Một số vị từ xử lý chuỗi:

1. frontchar(Str1, Ch, Str2): thường dùng để lấy ký tự đầu tiên của chuỗi, trong đó:
   * Str1 là chuỗi cho trước.
   * Ch là ký tự đầu tiên của chuỗi Str1.
   * Str2 là phần phần còn lại của chuỗi Str1 (sau khi loại bỏ ký tự Ch đầu tiên).

Thử chạy đoạn chương trình ví dụ sau và quan sát kết quả:

constants

c="TURBO PROLOG"

predicates

run

clauses

run:-frontchar(c,X,\_),write("X=",X),nl.

run:-frontchar(c,\_,Y),write("Y=",Y),nl.

goal

run,fail

1. fronttoken(Str1,Token,Str2): thường dùng để lấy TỪ đầu tiên của chuỗi, trong đó:
   * Str1 là chuỗi cho sẵn
   * Token là từ đầu tiên của chuỗi
   * Str2 là phần còn lại của chuỗi
   * Ký tự ngăn cách là một ký tự trắng, hoặc 1 ký tự không phải là CHỮ CÁI.

Chạy đoạn chương trình sau và quan sát kết quả:

constants

c="TURBO PROLOG"

predicates

run

clauses

run:-fronttoken(c,X,\_),write("X=",X),nl.

run:-fronttoken(c,\_,Y),write("Y=",Y),nl.

goal

run,fail

Chương trình cắt các TỪ trong một chuỗi và đưa vào danh sách:

domains

namelist=name\*

name=symbol

predicates

str\_namelist(string,namelist)

clauses

str\_namelist(S,[H|T]):-

fronttoken(S,H,S1),!,

str\_namelist(S1,T).

str\_namelist(\_,[]).

goal

str\_namelist("I go to school",Z),

write("Z=",Z)

1. concat(Str1, Str2, Str3): nối 2 chuỗi Str1 và Str2 để tạo ra Str3
2. char\_int(Ch, N): đổi ký tự Ch thành mã ASCII tương ứng của nó

Ví dụ:

goal

char\_int(‘a’,X),

char\_int(Y,65),

write(“X=”,X),nl,

write(“Y=”,Y),nl

1. str\_char(Str, Ch), str\_int(Str, N), str\_real(Str, X): dùng để đổi 1 chuỗi gồm 1 ký tự thành ký tự, 1 chuỗi gồm 1 số nguyên thành số nguyên và 1 chuỗi chứa 1 số thực thành số thực tương ứng.
2. str\_len(Str, N): cho biết chiều dài của chuỗi.
3. upper\_lower(StrUpper, StrLower): dùng để đổi chữ hoa thành chữ thường và ngược lại.
4. Viết chương trình thực hiện các công việc sau:
5. Đếm số từ có trong 1 chuỗi (từ được định nghĩa là 1 dãy các ký tự là CHỮ CÁI liên tiếp nhau).
6. Đếm số lượng ký tự là NGUYÊN ÂM có trong chuỗi.
7. Cho biết từ dài nhất trong chuỗi dài bao nhiêu ký tự.
8. Cho biết từ dài nhất trong chuỗi là từ nào.