

日本語の使用

原則としてプログラム内，ファイル名等に日本語は使用しないのが無難．コメントとして日本語を使用する場合は，プログラムの冒頭に次の一文をいれておくと error, warning は出ない．

```
:- encoding(utf8).
```

変数について

- 実行時に同じ節が再帰的に何度も呼ばれると、システムによって呼び出しごとにその節の変数には異なる変数が準備される。(第 2 回講義の資料「sum の実行過程の補足」参照.)
- ‘_’(underscore) からはじまる数字はシステムが導入した変数である．同じ番号のものには同じ値が代入される．

数値の扱い方

mod(E,2) =:= 0	正しい (=:= は両辺とも評価される)
mod(E,2) is 0	間違い (is は右辺のみ評価される)
mod(E,2) = 0	間違い (= は単一化なので両辺とも評価されない)

X=1	正しい (右辺を評価する必要はなく X は '1' という項になる)
X=1+2	間違い (右辺は評価されず X は '1+2' という項になる)

head unification

(1)(2) の記述はともに正しいが，(2) は冗長．リスト処理に慣れてきたら (1) を推奨する．

```
(1) double_num([X|X1],[Y|Y1]) :- Y is 2*X, double_num(X1,Y1).
```

```
(2) double_num(L1,L2) :- L1=[X|X1], Y is 2*X, double_num(X1,Y1), L2=[Y|Y1].
```

練習問題

1. 整数を要素とするリスト L に含まれる正の数のリストが PL，負の数のリストが NL であるという関係を表す述語 pos_neg(L,PL,NL) のプログラムを作成せよ．たとえば，pos_neg([3,-10,0,16],PL,NL) は PL=[3,16], NL=[-10] となって成功する．

演習問題 (r4)

* のついている問題はオプションなのでできる者のみ解答せよ。

以下の問題において，リストの先頭は 0 番目ではなく 1 番目と数える。

- (1) リスト L1 の要素がすべて整数であるとする。N を 1 以上の整数とする。L1 の要素の中で N で割り切れる要素のリストが L2 であり，N で割り切れない要素のリストが L3 であるという関係を表す述語 `divide_by_n(N,L1,L2,L3)` のプログラムを組み込みオペレータ `mod` を利用して作成せよ。たとえば，`divide_by_n(3,[1,2,3,4,5,6],L2,L3)` は `L2=[3,6],L3=[1,2,4,5]` となって成功する。
なお，プログラム内で `do(X,Y) :- divide_by_n(3,[1,2,3,4,5,6],X,Y)`. と定義しておくとし，実行は `?- do(X,Y)`. と打ち込むだけですむ。
- (2) リスト L1 の要素がすべて整数であるとする。L1 の要素の中でその値が Min 以上 Max 以下のものから成るリストが L2 であるという関係を表す述語 `mid(L1,Min,Max,L2)` のプログラムを作成せよ。たとえば `mid([5,12,2,25,18],10,20,L2)` は `L2=[12,18]` となって成功する。
- (3) リスト L1 の 1 番目から N 番目の要素を順に並べたものがリスト L2 であるという関係を表す述語 `n_elements(L1,N,L2)` を定義せよ。ただし，N としてはリストの長さを越える値は考えなくてよい。たとえば `n_elements([a,b,c],2,L2)` を実行すると，`L2=[a,b]` となって成功する。
- (4) リスト L1 に含まれる要素 X をすべて削除したリストが L2 であるという関係を表す述語 `del_element(X,L1,L2)` のプログラムを作成せよ。たとえば，`del_element(a,[a,b,a,x],L)` は `L=[b,x]` となって成功する。
- (5) リスト L1 に含まれる要素 X をすべて Y に置き換えたリストが L2 であるという関係を表す述語 `rewrite_list(X,Y,L1,L2)` のプログラムを作成せよ。たとえば，`rewrite_list(a,x,[a,b,a,x],L)` は `L=[x,b,x,x]` となって成功する。
- (6)* `member` の定義を参考にして，長さ 1 以上のリスト L の N 番目の要素が X であるという関係を表す述語 `nth_element(L,N,X)` を定義せよ。ただし，N としてはリストの長さを越える値は考えなくてよい。また，リストの先頭は 0 番目ではなく 1 番目と数える（例：[a,b,c] というリストに対しては N は 1,2,3 しか入力されないとしてよい。）たとえば `nth_element([a,b,c],2,X)` を実行すると，`X=b` となって成功する。
[考え方] この問題では，再帰呼出しのたびに N が 1 つずつ減るとともに，リスト L の長さも 1 つずつ減る。N の制約を考えると，停止条件としては `L=[]` をとるのはなく `N=1` をとる。このとき L は長さ 1 以上のリストになるので，そのときの冒頭の値をとってくればよい。
- (7)* (6) の述語の定義 `nth_element(L,N,X)` を拡張してリストの長さが N 未満の場合は，エラーメッセージを出力するようにせよ。メッセージの出力には組み込み述語 `write` を利用し，`write('The length is less than N')` のように記述する。また，N は 1 以上の整数としてよい。
[考え方] 停止条件が 2 つあることに注意。
- (8) r3 (r4 ではない) の練習問題 2 の解答例プログラムについて (i) 各節の論理的意味を示せ。
(ii) 第 2 節を

```
sum_list([X|Y],N) :- sum_list(Y,N1), N1 is N-X.
```

と記述すると実行時にエラーが出る。この理由を説明せよ。