

ΑΝΑΔΡΟΜΗ

Να γράψετε έναν κανόνα, για τη σχέση `count_till / 2`, η οποία επιστρέφει στο δεύτερο όρισμα, το άθροισμα των αριθμών από το 1 μέχρι το πρώτο όρισμα.

```
count_till(0,0).
```

```
count_till(1,1).
```

```
count_till(N,X) :- M1 is N-1, count_till(M1,X1), X is N+X1.
```

ΛΙΣΤΕΣ

Να ορισθεί το κατηγορημα `length_list /2`, το οποίο έχει ως πρώτο όρισμα τη λίστα και ως δεύτερο θα επιστρέφει το μήκος της.

```
length_list([],0).
```

```
length_list([H|T],X) :- length_list(T,X1), X is X1+1.
```

ΑΠΟΚΟΠΗ

Ορίστε το κατηγορημα `proximo/2`, με πρώτο όρισμα έναν αριθμό και δεύτερο την ένδειξη `thetikos`, `arnitikos` ή `miden`, αναλόγως με το πρόσημο του αριθμού. Ξαναορίστε το κατηγορημα με χρήση αποκοπής και αποφεύγοντας τους περιττούς ελέγχους.

```
proximo(X, thetikos) :- X>0, !.
```

```
proximo(X, arnitikos) :- X<0, !.
```

```
proximo(X, miden).
```

ΦΥΣΙΚΟΙ ΑΡΙΘΜΟΙ ΩΣ ΣΥΝΘΕΤΟΙ ΟΡΟΙ

Να ορισθεί το κατηγορημα `megalytero_kata_dyo/2`, το οποίο επιστρέφει `true` όταν το πρώτο όρισμα είναι μεγαλύτερο από το δεύτερο όρισμα κατά δύο.

```
megalytero_kata_dyo(s(s(0)),0).
```

```
megalytero_kata_dyo(s(X),s(Y)):-megalytero_kata_dyo(X,Y).
```

ΕΙΣΟΔΟΣ/ΕΞΟΔΟΣ ΑΠΟ ΑΡΧΕΙΑ

Να δημιουργηθεί πρόγραμμα `cube/0` το οποίο θα δέχεται τιμές από το πληκτρολόγιο και θα υπολογίζει τον κύβο τους. Το πρόγραμμα θα τερματίζεται όταν ο χρήστης δώσει την τιμή `stop`.

```
cube :- read(X), calc(X).
```

```
calc(stop) :- !.
```

```
calc(X) :- C is X * X * X, write(C), cube.
```