

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ  
«ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ»

ΠΡΟΒΛΗΜΑ  
**4.Πύργος του Ανόι**

ΟΜΑΔΑ  
**ΣΙΑΦΛΕΚΗΣ ΑΝΑΣΤΑΣΙΟΣ (Α.Μ.: 4634)**

**Πάτρα 2015-2016**

(α) Στο συγκεκριμένο πρόβλημα έχουμε πέντε δίσκους διαφορετικού μεγέθους να βρίσκονται σε τρεις πασσάλους, ώστε ο κάθε δίσκος να βρίσκεται πάνω από μεγαλύτερο δίσκο κι επιτρέπεται να μετακινηθεί μόνο ο πάνω πάνω δίσκος από κάθε πάσσαλο. Οι δίσκοι στην αρχή βρίσκονται σ' έναν πάσσαλο και σκοπός μας είναι να τους μετακινήσουμε, ώστε να βρεθούν κι οι πέντε ταυτόχρονα σε άλλο πάσσαλο. Τα κατηγορήματα που χρησιμοποιήθηκαν για να μπορούν να εκφραστούν τα γεγονότα του προβλήματος είναι τα εξής:

- `arxiki_katastasi`: Ορίζει την αρχική κατάσταση του προβλήματος.
- `katastasi`: Ορίζει τον πάσσαλο στον οποίο βρίσκεται ο κάθε δίσκος. Έχει δυο μεταβλητές, τον πάσσαλο και τον αριθμό του δίσκου.
- `metakinisi`: Ορίζει την διαδικασία της μετακίνησης του κάθε δίσκου. Τυπώνει τα κατάλληλα μηνύματα κι έχει τρεις μεταβλητές. Τον δίσκο προς μετακίνηση, τον πάσσαλο στον οποίο βρίσκεται και τον πάσσαλο στον οποίο θέλουμε να τον μετακινήσουμε.
- `print_state_a`, `print_state_b`, `print_state_c`: Τυπώνουν τους δίσκους που βρίσκονται σε κάθε πάσσαλο.
- `win`: Ορίζει την κατάσταση στην οποία τερματίζει το πρόβλημα, έχοντας επιτευχθεί ο στόχος του.
- `check`: Ορίζει τον περιορισμό του να μετακινηθεί μεγαλύτερος δίσκος πάνω σ' έναν μικρότερο, διακόπτοντας την εκτέλεση του προγράμματος και τυπώνοντας μήνυμα λάθους.
- `updisk`: Ορίζει τον περιορισμό ότι μπορεί να μετακινηθεί μόνο ο πάνω δίσκος από κάθε πάσσαλο, διακόπτοντας την εκτέλεση του προγράμματος και τυπώνοντας μήνυμα λάθους.
- `move`: Ορίζει την διαδικασία μετακίνησης κάθε δίσκου, αφού έχουν γίνει οι έλεγχοι για τους περιορισμούς που δίνονται από το πρόβλημα.
- `lysi`: Περιέχει τα βήματα για την λύση του παιχνιδιού που μας ζητείται από το τελευταίο ερώτημα.

---

(β) Για την εκτύπωση της κατάστασης του προβλήματος στην οθόνη, χρησιμοποιούμε τις `print_state_a`, `print_state_b`, `print_state_c`, οι οποίες τυπώνουν τους δίσκους που βρίσκονται στους αντίστοιχους πασσάλους. Οι συναρτήσεις αυτές, εκτός από την κλήση τους από τον χρήστη, καλούνται και μετά από τον ορισμό της αρχικής κατάστασης και κάθε μετακίνησης. Ο κώδικάς τους είναι ο εξής:

```
print_state_a:-
    findall(X,katastasi(a,X),Passalosa),
    write('passalos A: '),write(Passalosa),nl.

print_state_b:-
    findall(Y,katastasi(b,Y),Passalob),
    write('passalos B: '), write(Passalob),nl.

print_state_c:-
    findall(Z,katastasi(c,Z),Passaloc),
    write('passalos C: '), write(Passaloc),nl.
```

Και τυπώνουν στην οθόνη τα εξής:

```
2 ?- print_state_a,print_state_b,print_state_c.  
passalos A: [5,4,3,2,1]  
passalos B: []  
passalos C: []  
true.
```

```
3 ?- |
```

---

(γ),(δ),(ε)

Οι κώδικες που απαντάνε στα ερωτήματα αυτά είναι οι “hanoi tower\_1”, “Hanoi tower\_2” & “Hanoi tower\_3”. Πιο συγκεκριμένα:

***Hanoi tower\_1.***

Στον κώδικα αυτό, είναι ορισμένη η αρχική κατάσταση του προβλήματος κι ο χρήστης προσπαθεί να το λύσει.

```
%dilosi synartisewn dynamic  
  
:-dynamic arxiki_katastasi/1.  
:-dynamic katastasi/2.  
:-dynamic metakinisi/3.  
:-dynamic print_state/5.  
:-dynamic print_state_a/5.  
:-dynamic print_state_b/5.  
:-dynamic print_state_c/5.  
  
:-dynamic updisk/3.  
:-dynamic check/3.  
:-dynamic move/3.  
%Eisagwgiko mhnyma pou typonei to programma  
:-  
    write('-----Welcome-----'),nl,  
    write('-----Hanoi Towers-----'),nl,  
    write('Vasikes odhgies: '),nl,  
    write('1)Metakinisi diskou: ?-move(A,B,C). '),nl,  
    write('\t opou A=diskos pros metakinisi, '),nl,  
    write('\t opou B=passalos pou vrisketai o diskos'),nl,  
    write('\t opou C=passalos pou tha metakinithei'),nl,  
    write('2)Periorismo: '),nl,
```

```

        write('-Mporei na metakinitheí mono o teleftaios diskos'),nl,
        write('-Megalyteros diskos den mporei na mpei panw apo mikrotero'),nl.

%katastasi(passalos,Thesi ston passalo,diskos)
arxiki_katastasi:-
retractall(katastasi(_,_)),
assert(katastasi(a,5)),
assert(katastasi(a,4)),
assert(katastasi(a,3)),
assert(katastasi(a,2)),
assert(katastasi(a,1)),
print_state_a,print_state_b,print_state_c.

%metakinisi me periorismous
move(N,X,Y):- updisk(N,X,Y),check(N,X,Y),metakinisi(N,X,Y).

%metakiniseis
metakinisi(N,X,Y):-
    write('Metakinise ton disko '), write(N),
    write(' apo ton passalo '), write(X), write(' '),
    write('ston '), write(Y),nl,
    retract(katastasi(X,N)),
    assert(katastasi(Y,N)),print_state_a,
    print_state_b,print_state_c,nl,win.

%ektyposi katastasewn
%print_state:- print_state_a,print_state_b,print_state_c,nl.

print_state_a:-
    findall(X,katastasi(a,X),Passalosa),
    write('passalos A: '),write(Passalosa),nl.

print_state_b:-
    findall(Y,katastasi(b,Y),Passalob),
    write('passalos B: '), write(Passalob),nl.

print_state_c:-
    findall(Z,katastasi(c,Z),Passalosc),
    write('passalos C: '), write(Passalosc),nl.

```

```

%synthiki nikis
win:- (katastasi(K,1),katastasi(L,2),
      katastasi(M,3),katastasi(N,3),
      katastasi(O,4),katastasi(P,5),
      K==L,L==M,M==N,N==O,O==P,
      write('---MPRAVO KERDISES!---'),nl);nl.

%\\\\\\-----Periorismoι-----\\\\\\
% 1.den mporei megalyteros diskos na mpei panw apo mikrotero
%kata analogia tou metakinisi(N,X,Y)

check(N,X,Y):- (katastasi(Y,M),katastasi(X,N),
                N>M->write('Lathos kinisi'),nl,!fail);true.

% 2.mono o panw diskos mporei na metakinithe

updisk(N,X,Y):- (katastasi(X,N),katastasi(X,M),
                 N>M->write('Den mporei na metakinithe o diskos '),
                 write(N),write(' ston passalo '), write(Y),
                 write(' giati den einai o panw diskos'),nl,!fail);
true.

```

### ***Hanoi tower\_2***

Ο κώδικας αυτός είναι μια τροποποίηση του “hanoi tower\_1”, όπου ο χρήστης ορίζει εκείνος την αρχική κατάσταση του προβλήματος κι έπειτα προσπαθεί να το λύσει. Επί της ουσίας η τροποποίηση είναι στην «arxiki\_katastasi» κι η οποία φαίνεται παρακάτω:

```

arxiki_katastasi:-
retractall(katastasi(_, _)),
write('Poios passalos thes na einai o arxikos? '),nl,
read(X),
assert(katastasi(X,5)),
assert(katastasi(X,4)),
assert(katastasi(X,3)),
assert(katastasi(X,2)),
assert(katastasi(X,1)),
print_state_a,print_state_b,print_state_c.

```

### ***Hanoi tower\_3***

Ο κώδικας αυτός είναι μια τροποποίηση του “hanoi tower\_2”, όπου εμπεριέχεται το κομμάτι του κώδικα, που ορίζει το πρόβλημα και το λύνει, σύμφωνα με τα ζητούμενα του ερωτήματος (ε). Πιο συγκεκριμένα η τροποποίηση είναι η εξής:

lysi:-

```
arxiki_katastasi(a),nl,  
move(1,a,c),  
move(2,a,b),  
move(1,c,b),  
move(3,a,c),  
move(1,b,a),  
move(2,b,c),  
move(1,a,c),  
move(4,a,b),  
move(1,c,a),  
move(2,c,b),  
move(1,a,b),  
move(3,c,a),  
move(1,b,a),  
move(2,b,c),  
move(1,a,c),  
move(3,a,b),  
move(1,c,a),  
move(2,c,b),  
move(1,a,b),  
move(5,a,c),  
move(1,b,c),  
move(2,b,a),  
move(1,c,a),  
move(3,b,c),  
move(1,a,c),  
move(2,a,b),  
move(1,c,b),  
move(3,c,a),  
move(1,b,c),  
move(2,b,a),  
move(1,c,a),  
move(4,b,c),  
move(1,a,c),  
move(2,a,b),  
move(1,c,b),  
move(3,a,c),  
move(1,b,a),  
move(2,b,c),  
move(1,a,c).
```