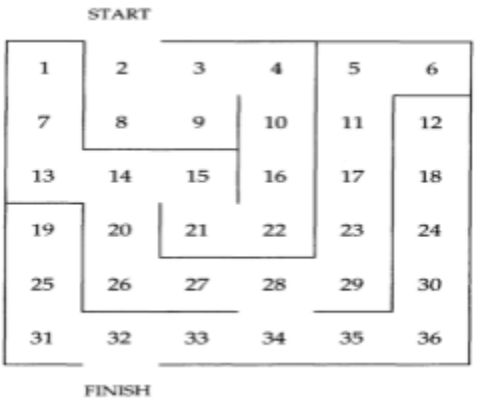
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ, ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΛΟΓΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ (ΤΛΕΣ): 3η Ατομική Εργασία

ΓΕΩΡΓΙΟΣ ΣΕΪΜΕΝΗΣ – Π19204

ΠΡΟΣΕΓΓΙΣΗ:

Ως εργασία ζητήθηκε να γραφεί κώδικας σε prolog, που θα βρίσκει λύση στον παρακάτω λαβύρινθο:



Όπως φαίνεται, η συντομότερη λύση είναι αν ακολουθήσουμε το μοτίβο:
2-3-4-10-16-22-21-15-14-20-26-27-28-34-33-32.

Προσεγγίζοντας αυτήν την λύση, θα πρέπει να φτιάξουμε μία συνάρτηση η οποία θα μπορεί να διασχίζει τον λαβύρινθο, υπό κάποια κριτήρια, και να εκτυπώνει στην οθόνη μία λίστα η οποία θα αναδεικνύει το «μονοπάτι» που ακολούθησε.

Πριν, όμως, υλοποιήσουμε τον «λύτη» του λαβυρίνθου, θα πρέπει ορίσουμε πού θα υπάρχουν τα «τοιχώματα» όπως φαίνεται στην εικόνα παραπάνω. Οπότε, θα πρέπει να ορίσουμε ποιοι αριθμοί ενώνονται μεταξύ τους, βάσει των κατηγορημάτων.

ΥΛΟΠΟΙΗΣΗ:

Με το κατηγόρημα **connect(α, β)** όπου α και β δύο αριθμοί που δεν έχουν ανάμεσά τους τοίχωμα, μπορούμε να συνδέσουμε ένα ζευγάρι αριθμών, όπου θα έχουν πρόσβαση ο ένας στον άλλον. Αυτήν τη διαδικασία θα την επαναλάβουμε 38 φορές, για 36 αριθμούς συν τα σημεία start και finish. Αυτό το έχουμε κάνει από τις γραμμές 3-148, όπως φαίνεται παρακάτω:

askisi3.pl

Άσκηση 3 > askisi3.pl

```
/* πρώτα συνδέουμε τους κόμβους μεταξύ τους */

%START
connect(start, 2).

%1
connect(1, 7).

%2
connect(2, start).
connect(2, 3).
connect(2, 8).

%3
connect(3, 2).
connect(3, 4).
connect(3, 9).

%4
connect(4, 3).
connect(4, 10).

%5
connect(5, 6).
connect(5, 11).

%6
connect(6, 5).

%7
connect(7, 1).

%8
connect(8, 2).
connect(8, 9).
```

askisi3.pl

Άσκηση 3 > askisi3.pl

```
connect(28, 29).

%29
connect(29, 23).
connect(29, 28).

%30
connect(30, 24).
connect(30, 36).

%31
connect(31, 25).
connect(31, 32).

%32
connect(32, 31).
connect(32, 33).
connect(32, finish).

%33
connect(33, 32).
connect(33, 34).

%34
connect(34, 28).
connect(34, 33).
connect(34, 35).

%35
connect(35, 34).
connect(35, 36).

%36
connect(36, 30).
connect(36, 35).
```

Ύστερα, θα πρέπει να υλοποιήσουμε την συνάρτηση η οποία θα βρίσκει το βέλτιστο δυνατό μονοπάτι ανάμεσα σε δύο σημεία. Θα έχουμε τρία πράγματα, τα οποία θα επιβλέπουμε:

- Το αρχικό σημείο
- Το τελικό σημείο (ο στόχος)
- Η λίστα με τα βήματα που ακολουθήσαμε

Με το εξής κατηγόρημα επιχειρούμε αυτόν το στόχο και μάλιστα ονομάζουμε την λίστα ως “Path”. **Όπως παρατηρείται στα ορίσματα, πρώτα μπαίνει το τελικό σημείο και μετά η αρχή.**

```
find_path(Destination, Source, Path) :- find_path(Destination, Source, [], Path).
```

Πριν αρχίσουμε να βρίσκουμε το μονοπάτι, πρέπει πρώτα να ορίσουμε ότι υπάρχει μονοπάτι εκατέρωθεν των κόμβων, ανεξάρτητα από την σειρά που ορίζονται. Δηλαδή αν έχουμε το 2 και το 8 συνδεδεμένα, σημαίνει ότι όποια κατεύθυνση κι αν ακολουθήσουμε, υπάρχει μονοπάτι. Αυτό το διασφαλίζει η παρακάτω συνάρτηση:

```
is_combo(X, Y) :- connect(X, Y); connect(Y, X).
```

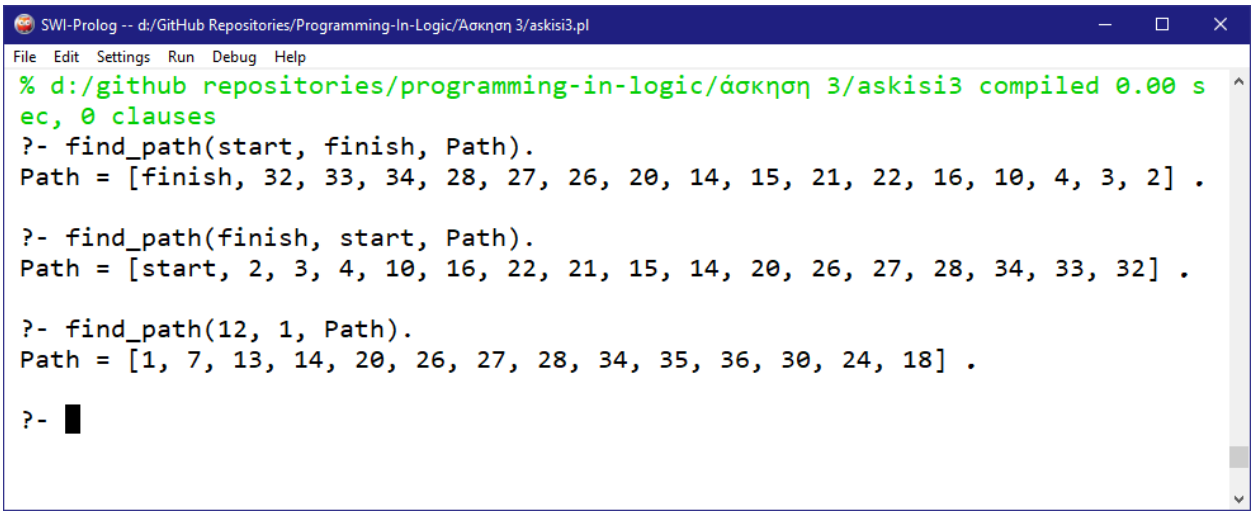
Ύστερα, χρησιμοποιώντας το ίδιο κατηγόρημα μπορούμε να ορίσουμε την διαδικασία ώστε να ακολουθήσουμε το μονοπάτι. Για να βρούμε το μονοπάτι, στην ουσία, πρέπει να ισχύουν τα εξής:

- Αν θέλουμε να πάμε από έναν κόμβο στον εαυτό του, τότε υπάρχει μονοπάτι και η λίστα θα είναι άδεια.
- Αν θέλουμε να πάμε από έναν κόμβο α σε έναν άλλο κόμβο β, θα πρέπει να βρούμε έναν άλλο κόμβο γ, ο οποίος θα σχηματίζει μονοπάτι με το α και το β, χωρίς να υπάρχει ακόμη στη λίστα με τους προσπελασμένους κόμβους.
- Επαναλαμβάνουμε τη διαδικασία μέχρι να ευρεθεί μέσω των κόμβων γ, ο κόμβος β (που είναι ο στόχος μας).

```
find_path(A, A, Path, Path).
find_path(A, B, List, Path) :- is_combo(A, C), \+ member(C, T), find_path(C, B,
[C|List], Path).
```

ΣΤΗΝ ΠΡΑΞΗ:

Το μόνο που μένει σε αυτό το σημείο, είναι να καλέσουμε την find_path/4 και να δούμε ότι πράγματι δουλεύει.



Βλέπουμε παραδείγματα δύο μονοπατιών. Στο πρώτο πάμε από το start μέχρι το finish (και μετά βλέπουμε την αντίστροφη διαδρομή). Στο δεύτερο βλέπουμε πως να πάμε από το 1 μέχρι το 12 (που είναι ένα σχετικά μεγάλο μονοπάτι). Όπως βλέπουμε, ο κώδικας εκτελείται με ακρίβεια και εκτυπώνεται ολόκληρο το μονοπάτι (χωρίς τον τελικό κόμβο), βρίσκοντας πάντα την καλύτερη δυνατή διαδρομή για να φτάσουμε από το σημείο α στο σημείο β.

ΤΕΛΟΣ ΠΑΡΟΥΣΙΑΣΗΣ

~ Π19204, ΓΕΩΡΓΙΟΣ ΣΕΪΜΕΝΗΣ