

1.2 Αναδρομικοί αλγόριθμοι

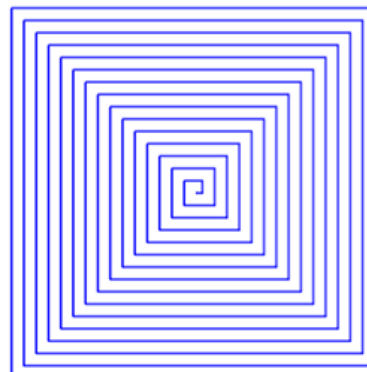
Θέλουμε να σχεδιάσουμε το διπλανό σχήμα. Παρατηρούμε ότι, ενώ φαίνεται πως κάθε φορά η στροφή είναι 90° (όπως όταν σχεδιάζεται ένα τετράγωνο), το μήκος κάθε πλευράς αλλάζει βαθμιαία. Ένας πρώτος αλγόριθμος σχεδιασμού θα μπορούσε να είναι ο εξής:

Ορισμός Σχεδίασε_Σπирάλ βήματα

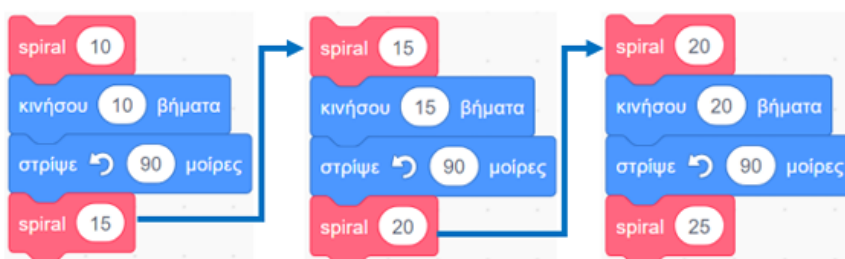
Στρίψε αριστερά 90°

Σχεδίασε_Σπирάλ βήματα + 5

Παρατηρούμε ότι η εντολή **Σχεδίασε_Σπирάλ** αναφέρεται ξανά στον ε-αυτό της, κάτι που δεν έχουμε δει ξανά. Αυτού του είδους οι αυτοαναφορικοί ορισμοί λέγονται αναδρομικοί.



Μετά την κλήση της εντολής `spiral(10)` θα ακολουθήσει η παρακάτω αλληλουχία κλήσεων.



Αν δοκιμάσουμε να εκτελέσουμε αυτή την εντολή στο Scratch, θα διαπιστώσουμε ότι δε σταματάει ποτέ, άρα παραβιάζει την περατότητα.



Ωστόσο, επειδή οι υπολογιστές έχουν πεπερασμένα όρια στην αναπαράσταση αριθμών, το πρόγραμμα θα σταματήσει, όταν ξεπεράσει τον μέγιστο αριθμό που μπορεί να αναπαραστήσει η γλώσσα προγραμματισμού που χρησιμοποιούμε με απρόβλεπτη συμπεριφορά. Γι' αυτό θα χρειαστεί να προσθέσουμε έναν έλεγχο για το μήκος της γραμμής, ώστε ο σχεδιασμός να σταματάει όταν ξεπεράσει αυτό το μήκος. Μια λογική τιμή για αυτό είναι το μέγιστο μήκος της σκηνής στον κατακόρυφο άξονα, δηλαδή 360 βήματα. Άρα, η αναδρομική κλήση του επόμενου βήματος `spiral(μήκος + 5)` θα γίνει μόνο αν δεν έχουμε ξεπεράσει τα όρια της οθόνης. Σε αντίθετη περίπτωση, η εκτέλεση του προγράμματος θα σταματήσει.



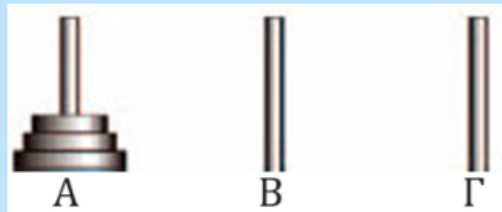


Δραστηριότητα 1 - Οι πύργοι του Ανόι

Το πρόβλημα των πύργων του Ανόι είναι χαρακτηριστικό παράδειγμα αναδρομής και θεωρείται από τα θεμελιώδη προβλήματα στην επιστήμη της Πληροφορικής. Έχετε μια μικρή συλλογή από δίσκους και τρεις στύλους πάνω στους οποίους μπορείτε να τους τοποθετήσετε (ο κάθε δίσκος έχει στη μέση μία οπή ώστε να τοποθετείται στο στύλο).

Οι δίσκοι είναι όλοι τοποθετημένοι στον αριστερό στύλο σε αύξουσα σειρά ανάλογα με το μέγεθός τους (δηλαδή ο μικρότερος είναι πάνω) και πρέπει να μετακινηθούν στον Γ. Ο Β μπορεί να χρησιμοποιηθεί ως βοηθητικός στύλος. Κανένας δίσκος δεν μπορεί να τοποθετηθεί πάνω από δίσκο που είναι μικρότερος από αυτόν.

Μόνο ένας δίσκος μπορεί να μετακινηθεί κάθε φορά.



Βήμα 1

Να λύσετε το πρόβλημα των πύργων του Ανόι για τρεις δίσκους.

Βήμα 2

Να γράψετε έναν αλγόριθμο για μια ιδεατή γλώσσα προγραμματισμού στην οποία η μόνη επιτρεπτή εντολή είναι η

Μετακίνηση (αρχή, προορισμός)

η οποία μετακινεί τον πάνω δίσκο από το στύλο **αρχή** στο στύλο **προορισμός**. Το πρόγραμμά σας θα μετακινεί τους τρεις δίσκους από το στύλο Α στο στύλο Γ μέσω του Β.

Μετακίνηση_3_δίσκους (Α, Γ)

Μετακίνηση (Α, Γ)	
Μετακίνηση (Α, Β)	

Βήμα 3

Να γράψετε έναν αντίστοιχο αλγόριθμο για την περίπτωση των τεσσάρων δίσκων.

Παρατηρείτε κάποια σχέση μεταξύ του προβλήματος των τριών και αυτού των τεσσάρων δίσκων;

Αν μπορούσατε να χρησιμοποιήσετε την εντολή **Μετακίνηση_3_δίσκους** (X, Y) για την υλοποίηση του αλγορίθμου **Μετακίνηση_4_δίσκους** (X, Y), τι θα άλλαζε στην περιγραφή του αλγορίθμου;

Όμοια χρησιμοποιήστε τον αλγόριθμο **Μετακίνηση_4_δίσκους** (X, Y), για να λύσετε το πρόβλημα των 5 δι-

Όμοια χρησιμοποιήστε τον αλγόριθμο **Μετακίνησε_4_δίσκους** (X, Y), για να λύσετε το πρόβλημα των 5 δίσκων.

Βήμα 4

Μπορείτε να γενικεύσετε για την περίπτωση των N δίσκων;

Πόσες κινήσεις πιστεύετε ότι θα χρειαστούν για: **α)** 3 δίσκους **β)** 4 δίσκους **γ)** 5 δίσκους **δ)** 64 δίσκους;

ΠΑΡΑΔΕΙΓΜΑ ΑΝΑΔΡΟΜΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΕΝΟΣ ΣΠΙΡΑΛ ΣΕ SCRATCH

[How to draw Spiral Square pattern in scratch and pictoblox| - YouTube](#)

How to draw Spiral Square pattern in scratch and pictoblox| - YouTube