# ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
# ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ


## ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
## 5ο εξάμηνο

# 1η εργασία
# Το πρόβλημα του Knuth

**Εισαγωγικά:**

Το πρόβλημα του Knuth (Knuth, 1964) ορίζεται ως εξής: Ξεκινώντας με τον αριθμό 4, μια ακολουθία πράξεων εύρεσης τετραγωνικής ρίζας, ακέραιου μέρους (floor) και παραγοντικών μπορεί να φτάσει σε οποιονδήποτε επιθυμητό θετικό ακέραιο. Για παράδειγμα, μπορούμε να φτάσουμε στον αριθμό 5 με την παρακάτω ακολουθία πράξεων:

$$\left\lfloor \sqrt{\sqrt{\sqrt{\sqrt{\sqrt{(4!)!}}}}} \right\rfloor = 5$$

Μερικές παρατηρήσεις:

- Για να εφαρμοστεί η πράξη του παραγοντικού, πρέπει το αποτέλεσμα των προηγούμενων πράξεων να είναι ακέραιος αριθμός.
- Δεν έχει νόημα να εφαρμοστεί η πράξη floor δύο (ή περισσότερες) συνεχόμενες φορές. Η πράξη floor συνήθως είναι η τελευταία πράξη (εκτός αν δεν χρειάζεται), μπορεί όμως να εφαρμοστεί και νωρίτερα.
- Για να ελέγξετε ότι ένας αριθμός είναι ακέραιος, μπορείτε να υπολογίσετε την απόλυτη τιμή της διαφοράς του αριθμού από το ακέραιο μέρος του. Εάν η ποσότητα αυτή είναι μικρότερη από μια πολύ μικρή σταθερά ε (π.χ., ε=0.00000001), τότε ο αριθμός μπορεί να θεωρηθεί ακέραιος.

Κατασκευάστε ένα πρόγραμμα που να λύνει το πρόβλημα του Knuth χρησιμοποιώντας τους αλγορίθμους πρώτα σε πλάτος και επαναληπτική εκβάθυνση[1]. Το πρόγραμμα θα ζητά από τον χρήστη τον αριθμό που θέλουμε να επιτύχουμε (είσοδος από το πληκτρολόγιο), ενώ εφόσον βρεθεί λύση θα την εμφανίζει στην οθόνη ως την ακολουθία πράξεων. Για παράδειγμα, η λύση του προβλήματος για την επίτευξη του αριθμού 5 μπορεί να αναφερθεί ως εξής:

```
factorial
factorial
root
root
root
root
root
floor
```

[1] Σκεφτείτε γιατί δεν θα μπορούσαν να χρησιμοποιηθούν οι αλγόριθμοι πρώτα σε βάθος και αμφίδρομη αναζήτηση.

Το πρόγραμμά σας θα πρέπει να ζητά από τον χρήστη ποιον αλγόριθμο αναζήτησης θα χρησιμοποιήσει. Στο τέλος, θα πρέπει να αναφέρει το χρόνο που χρειάστηκε για να βρεθεί η λύση. Μπορείτε να θέσετε όριο στο πόσο χρόνο έχει στη διάθεσή του για να λύσει το πρόβλημα, π.χ. 60 seconds.

Δοκιμάστε τους δύο αλγορίθμους σε διάφορα προβλήματα και συγκρίνετε τους χρόνους επίλυσης. Παρουσιάστε τα αποτελέσματά/συμπεράσματά σας συγκριτικά για τους δύο αλγορίθμους.

Προσπαθήστε να λύσετε το πρόβλημα για όσο το δυνατόν περισσότερους αριθμούς-στόχους, ξεκινώντας από το 1.

Μπορείτε να χρησιμοποιήσετε οποιαδήποτε γλώσσα προγραμματισμού επιθυμείτε. Προσοχή στη χρήση πολύ μεγάλων αριθμών (που μπορεί να προκύψουν από το παραγοντικό), θα πρέπει να κάνετε έλεγχο πριν εφαρμόσετε την πράξη του παραγοντικού. Κάποιες γλώσσες (π.χ., η Python με το module BigNumber[2]) παρέχουν ειδικές δομές δεδομένων για τη χρήση πολύ μεγάλων αριθμών.

[2] https://pypi.org/project/BigNumber/