

Διαίρεση ακεραίου με ακέραιο και με υπόλοιπο

Η μέθοδος που χρησιμοποιεί η Divide.m είναι αυτή που μαθαίνουμε στο Δημοτικό (χωρίς αυτό να σημαίνει ότι η υλοποίηση της Divide είναι εύκολη υπόθεση). Ο καλύτερος τρόπος να εξηγήσω τι κάνει, είναι να χρησιμοποιήσω ένα απλό παράδειγμα:

Πως μπορώ να χωρίσω το 123 σε 7 ίσα μέρη; Ο τρόπος που χρησιμοποιούμε εξετάζει ένα-ένα τα ψηφία του διαιρετέου (123).

- Βήμα 1. Δοκιμάζω το πρώτο ψηφίο του αριθμού 123 από τα αριστερά, δηλαδή το 1. Το 7 δεν χωράει στο ένα.
- Βήμα 2. Κολλούμε και το επόμενο ψηφίο, δηλαδή το 2. Κι έτσι σχηματίζεται ο αριθμός 12.
- Βήμα 3. Το 7 στο 12 χωράει μια φορά και γι αυτό το πρώτο ψηφίο του αποτελέσματος θα είναι το ψηφίο 1.
- Βήμα 4. Αφαιρούμε το 7 από το 12. Ο αριθμός μας τώρα είναι το 5.
- Βήμα 5. Συνεχίζουμε με παρόμοιο τρόπο (όπως το βήμα 2). Κολλούμε το επόμενο ψηφίο, δηλαδή το 3. Κι ο αριθμός που σχηματίζεται είναι το 53.
- Βήμα 6. Το 7 στο 53 χωράει επτά φορές και γι αυτό το επόμενο ψηφίο του αποτελέσματος θα είναι το ψηφίο 7.
- Βήμα 7. Αφαιρούμε το 49 (δηλαδή επτά φορές το 7) από το 53 και ο αριθμός που εξετάζουμε είναι τώρα το 4.
- Βήμα 8. Μας έχουν τελειώσει τα ψηφία του διαιρετέου κι εδώ τελειώνει ο υπολογισμός.. Ας δούμε το αποτέλεσμα . Το αποτέλεσμα της διαίρεσης είναι ο αριθμός 17 και το υπόλοιπο είναι ο αριθμός 4.

Ορίστε τα πιο πάνω γραμμένα λίγο διαφορετικά:

123	7
-7	-----
----	17
53	
-49	

4	

Στον κώδικα οφείλουμε να ελέγχουμε κατά πόσο έχουμε βρει το πρώτο μη-μηδενικό ψηφίο του αποτελέσματος ώστε να μην προσθέτουμε αχρείαστα μηδενικά στην αρχή του αποτελέσματος (με άλλα λόγια αγνοούμε τα leading zeros του αποτελέσματος). Το αντίστοιχο στο πιο πάνω παράδειγμα είναι το πρώτο βήμα του υπολογισμού μας (Βήμα 1). Σε εκείνο το βήμα, διαπιστώνουμε ότι το 7 δεν χωράει στο 1. Κανονικά θα έπρεπε να κρατήσουμε το ψηφίο μηδέν και να το προσθέσουμε στην αρχή του αριθμού. Δεν θα είχε νοήμα να επιστρέψουμε τον αριθμό 017 ή το διάνυσμα (vector) [0,1,7]. Γι αυτό χρησιμοποιούμε τη found_first_nonzero_digit στον κώδικα.

```

function [q,r]=Divide(x,y)

q = [];

r = 0;

found_first_nonzero_digit = 0;

i = 1;

while i <= length(x)

    % add i-th element (digit) of vector x to r
    r = r * 10 + x(i);

    % check whether r is big enough to divide by y
    if r >= y

        % how many times does y go into r
        num_times_y_in_r = floor(r / y);

        % new remainder
        r = r - num_times_y_in_r * y;

        % we have found the first non-zero digit of the quotient
        found_first_nonzero_digit = 1;

    else

        % y is greater than r
        num_times_y_in_r = 0;

    endif

    % ignore leading zeros
    if found_first_nonzero_digit == 1

        q(end+1) = num_times_y_in_r;

    endif

    i = i + 1;

endwhile

```