

# ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2019-20)

## Εργασία 1

Ένας ακέραιος αριθμός μεγαλύτερος του 1 ονομάζεται *πρώτος* (prime) όταν έχει σαν μόνους διαιρέτες το 1 και τον εαυτό του. Για παράδειγμα, το 13 είναι πρώτος αριθμός, ενώ το 15 ( $= 3 \times 5$ ) δεν είναι.

*Παραγοντοποίηση* (factorization) ενός ακεραίου αριθμού μεγαλύτερου του 1 είναι η εύρεση των πρώτων αριθμών που το γινόμενό τους ισούται με τον αριθμό. Μερικά παραδείγματα παραγοντοποίησης είναι τα εξής:

$$\begin{aligned} 6 &= 2 \times 3 \\ 60 &= 2^2 \times 3 \times 5 \\ 187 &= 11 \times 17 \\ 5234 &= 2 \times 2617 \\ 23157 &= 3^2 \times 31 \times 83 \\ 128377 &= 128377 \\ 5489126 &= 2 \times 2744563 \\ 89000362 &= 2 \times 11 \times 29 \times 199 \times 701 \\ 127870872 &= 2^3 \times 3 \times 17 \times 313409 \\ 2001234568 &= 2^3 \times 1439 \times 173839 \end{aligned}$$

Ένας θετικός ακέραιος αριθμός μεγαλύτερος του 1 ονομάζεται *πλήρης-δυνάμειων* (powerful), ή ΠΔ για τη συνέχεια, όταν αναλύοντάς τον σε γινόμενο δυνάμειων πρώτων παραγόντων, για κάθε πρώτο παράγοντά του, ο εκθέτης του είναι μεγαλύτερος του 1. Για παράδειγμα, το 9000 είναι ΠΔ αριθμός, αφού  $9000 = 2^3 \times 3^2 \times 5^3$ , ενώ το 279 δεν είναι, αφού  $279 = 3^2 \times 31$  (το 31 έχει εκθέτη ίσο με 1). Κανένας από τους αριθμούς που δόθηκε παραπάνω η παραγοντοποίησή τους δεν είναι ΠΔ. Πιο μαθηματικά, αν ο αριθμός  $x$  αναλύεται σε πρώτους παράγοντες σαν

$$x = p_1^{e_1} \times p_2^{e_2} \times \dots \times p_n^{e_n}$$

όπου τα  $p_i$  ( $1 \leq i \leq n$ ) είναι πρώτοι, ικανή και αναγκαία συνθήκη για να είναι ο  $x$  ΠΔ είναι να ισχύει  $e_i > 1$ , για κάθε  $i$  με  $1 \leq i \leq n$ .

Ένας ΠΔ αριθμός ονομάζεται *Αχιλλεύς* (Achilles) όταν δεν είναι τέλεια δύναμη ενός ακεραίου αριθμού. Για παράδειγμα, ο ΠΔ αριθμός  $21600 = 2^5 \times 3^3 \times 5^2$  είναι Αχιλλεύς, ενώ ο ΠΔ αριθμός  $21952 = 2^6 \times 7^3 = 28^3$  δεν είναι Αχιλλεύς. Εύκολα μπορεί να διαπιστώσει κάποιος ότι ο ΠΔ αριθμός  $x$  της μορφής που αναφέρθηκε παραπάνω είναι Αχιλλεύς αν και μόνο αν

$$\text{MK}\Delta(e_1, e_2, \dots, e_n) = 1$$

όπου με MKΔ συμβολίζουμε τον μέγιστο κοινό διαιρέτη.

Γράψτε, κατ' αρχήν, ένα πρόγραμμα C (έστω ότι το πηγαίο αρχείο του ονομάζεται "achilles.c") το οποίο να βρίσκει και να εκτυπώνει όλους τους Αχιλλεύς αριθμούς που είναι μικρότεροι από το MAXNUM, το οποίο θα ορίσετε στο πρόγραμμά σας με #define.

Μία ενδεικτική εκτέλεση του προγράμματος, για MAXNUM ίσο με 1000 φαίνεται στη συνέχεια:

```
$ ./achilles
    72 is achilles number
   108 is achilles number
   200 is achilles number
```

```

288 is achilles number
392 is achilles number
432 is achilles number
500 is achilles number
648 is achilles number
675 is achilles number
800 is achilles number
864 is achilles number
968 is achilles number
972 is achilles number

```

Found 13 achilles numbers up to 1000

\$

Στη συνέχεια, αυξάνετε διαδοχικά το **MAXNUM**, για να διαπιστώσετε τον ρυθμό με τον οποίο αυξάνει και ο χρόνος εκτέλεσης του προγράμματός σας. Για να επιβεβαιώσετε την ορθότητα των αποτελεσμάτων σας, δείτε τον παρακάτω πίνακα, όπου φαίνονται τα πλήθη των Αχιλλείων αριθμών μέχρι το εκάστοτε μέγιστο **MAXNUM**.<sup>1</sup>

MAXNUM	#Achilles
1000	13
10000	60
100000	252
1000000	916
10000000	3158
100000000	10553
1000000000	34561

Τέλος, τροποποιήστε το πρόγραμμά σας έτσι, ώστε αντί να βρίσκει όλους τους Αχιλλείους αριθμούς μέχρι ένα μέγιστο **MAXNUM** (μπορείτε, πλέον, να αφαιρέσετε αυτή τη συμβολική σταθερά από το πρόγραμμά σας, γιατί δεν θα χρειάζεται στο εξής), να επιλέγει με τυχαίο τρόπο 10000000 θετικούς ακεραίους αριθμούς (όχι κατ' ανάγκη διαφορετικούς μεταξύ τους) και να βρίσκει ποιοι απ' αυτούς είναι Αχιλλείοι. Το πλήθος των αριθμών που θα ελεγχθούν (10000000) να μην τοποθετηθεί απ' ευθείας μέσα στο πρόγραμμά σας, αλλά να ορισθεί μέσω **#define** σαν τιμή της συμβολικής σταθεράς **COMPUTATIONS**. Ο τυχαίος τρόπος επιλογής των αριθμών που θα ελεγχθούν πρέπει να ακολουθεί τη διαδικασία που περιγράφεται στη συνέχεια.

Στην C μπορούμε να “γεννήσουμε” τυχαίους αριθμούς με χρήση των συναρτήσεων **srand** και **rand**, όπως επιδεικνύεται στο πρόγραμμα <http://www.di.uoa.gr/~ip/cprogs/gcd1cm.c>, στις σελίδες 44–45 των διαφανειών του μαθήματος, στο <http://www.di.uoa.gr/~ip/K04.pdf>.

Έστω ότι γεννάτε δύο τυχαίους αριθμούς  $y$  και  $z$  με χρήση δύο διαδοχικών κλήσεων της συνάρτησης **rand**. Ο αριθμός  $x$  που θα πρέπει να ελέγξετε αν είναι Αχιλλείος ή όχι δίνεται από τον τύπο<sup>2</sup>

$$x = ((y \bmod 32768) + 1) \cdot ((z \bmod 32768) + 1) + 1$$

Για την αρχικοποίηση της γεννήτριας των τυχαίων αριθμών, να χρησιμοποιήσετε την τρέχουσα ώρα, μέσω της συνάρτησης **time**, όπως ακριβώς γίνεται και στο πρόγραμμα **gcd1cm.c** που προαναφέρθηκε.

<sup>1</sup>Ο υπολογισμός της τελευταίας γραμμής του πίνακα χρειάζεται, πραγματικά, πολλή ώρα για να ολοκληρωθεί.

<sup>2</sup>Με **mod** συμβολίζουμε το υπόλοιπο διαίρεσης.

Το πρόγραμμά σας να εκτυπώνει εκείνους τους αριθμούς από τους 10000000 (που είναι η τιμή της συμβολικής σταθεράς **COMPUTATIONS**) που ελέγχθηκαν οι οποίοι είναι Αχίλλειοι και, επίσης, να υπολογίζει και να εκτυπώνει το πλήθος τους.

Μία ενδεικτική εκτέλεση<sup>3</sup> του προγράμματος φαίνεται στη συνέχεια:

```
$ ./achilles
Current time is 1572712634

21807104 is Achilles number
72595656 is Achilles number
205789993 is Achilles number
4121741 is Achilles number
171478296 is Achilles number
.....
192589600 is Achilles number
65802492 is Achilles number
7469451 is Achilles number
245000 is Achilles number
62989088 is Achilles number

Found 255 Achilles out of 10000000 tested numbers
$
```

### Σημειώσεις/Απαγορεύσεις:

1. Στο πρόγραμμα που θα παραδώσετε να θέσετε ως τιμή της συμβολικής σταθεράς **COMPUTATIONS**, όχι κατ' ανάγκη το 10000000, αλλά τη μέγιστη δύναμη του 10, για την οποία το πρόγραμμά σας τερματίζει σε λιγότερο από ένα λεπτό σε μηχάνημα Linux του εργαστηρίου του Τμήματος.
2. Για να χρονομετρήσετε το πρόγραμμά σας, μπορείτε να το καλέσετε (σε μηχάνημα Linux) ως **"time ./achilles"**. Ο πρώτος αριθμός που θα εκτυπωθεί από την **time** είναι ο χρόνος, σε δευτερόλεπτα, που αφιέρωσε ο επεξεργαστής του υπολογιστή σας στο πρόγραμμά σας. Εναλλακτικά, μπορείτε να χρησιμοποιήσετε τη συνάρτηση **clock**, τη χρήση της οποία μπορείτε να δείτε στο πρόγραμμα <http://www.di.uoa.gr/~ip/cprogs/sorting.c>, στις σελίδες 167–169 των διαφανειών του μαθήματος.
3. Αν θέλετε να συγκρίνετε τα αποτελέσματά σας με αυτά της ενδεικτικής εκτέλεσης που δόθηκε προηγουμένως, μπορείτε να αρχικοποιήσετε την γεννήτρια τυχαίων αριθμών όχι με τον τρέχοντα χρόνο, αλλά με τη συγκεκριμένη τιμή που φαίνεται στην ενδεικτική εκτέλεση και, φυσικά, να δουλέψετε σε μηχάνημα Linux του εργαστηρίου του Τμήματος.
4. Στην εργασία αυτή **απαγορεύονται αυστηρά η χρήση πινάκων, μεταβλητών κινητής υποδιαστολής (double, κλπ.) και συναρτήσεων της μαθηματικής βιβλιοθήκης της C**.
5. Η παράδοση της εργασίας αυτής συνίσταται στην υποβολή του πηγαίου αρχείου **achilles.c** με διαδικασία που θα ανακοινωθεί σύντομα.

---

<sup>3</sup>Η συγκεκριμένη εκτέλεση έγινε σε μηχάνημα Linux του εργαστηρίου του Τμήματος. Σε άλλες πλατφόρμες (π.χ. Windows/Dev-C++, MacOS) τα αποτελέσματα ενδέχεται να διαφέρουν, διότι η γεννήτρια τυχαίων αριθμών μπορεί να παράγει διαφορετική ακολουθία αριθμών, ακόμα και με το ίδιο "φύτρο".