

## Άσκηση 1

Ένας ακέραιος αριθμός μεγαλύτερος του 1 ονομάζεται *πρώτος* 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. Οι ακέραιοι μεγαλύτεροι του 1 που δεν είναι πρώτοι, όπως το 15, ονομάζονται *σύνθετοι*.

Κάθε ακέραιος αριθμός μεγαλύτερος του 1 μπορεί να αναλυθεί σε *γινόμενο πρώτων παραγόντων*. Προφανώς, οι πρώτοι αριθμοί έχουν σαν μόνο παράγοντα τον εαυτό τους. Για παράδειγμα, ο αριθμός 150 μπορεί να γραφεί σαν το γινόμενο  $2 \times 3 \times 5 \times 5$ . Η διαδικασία εύρεσης των πρώτων παραγόντων ενός αριθμού ονομάζεται *παραγοντοποίηση*.

Ένας ακέραιος αριθμός μεγαλύτερος του 1 ονομάζεται *αριθμός Smith* (Smith number) όταν είναι σύνθετος και το άθροισμα των ψηφίων του ισούται με το άθροισμα των ψηφίων όλων των πρώτων παραγόντων του. Για παράδειγμα, ο αριθμός 728 είναι αριθμός Smith, αφού  $728 = 2 \times 2 \times 2 \times 7 \times 13$  και  $7 + 2 + 8 = 2 + 2 + 2 + 7 + 1 + 3 = 17$ .

Γράψτε ένα πρόγραμμα C (έστω ότι το πηγαίο αρχείο του ονομάζεται “smith.c”) το οποίο να επιλέγει με τυχαίο τρόπο 100000 θετικούς ακεραίους αριθμούς (όχι κατ’ ανάγκη διαφορετικούς μεταξύ τους) και να βρίσκει ποιοι απ’ αυτούς είναι Smith. Το πλήθος των αριθμών που θα ελεγχθούν (100000) 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/gcdlcm.c>, στις σελίδες 44–45 των διαφανειών του μαθήματος, στο <http://www.di.uoa.gr/~ip/K04.pdf> (ή σελίδες 45–46 στα εκτυπωμένα αντίτυπα). Λεπτομέρειες για τη χρήση των συναρτήσεων `srand` και `rand` μπορείτε να μάθετε δίνοντας στα μηχανήματα Linux του Τμήματος “`man 3 srand`” ή “`man 3 rand`” (πιθανότατα, σε άλλα συστήματα Unix, να χρειάζεται “`man srand`” ή “`man rand`”).

Έστω ότι γεννάτε δύο τυχαίους αριθμούς  $y$  και  $z$  με χρήση δύο διαδοχικών κλήσεων της συνάρτησης `rand`. Ο αριθμός που θα πρέπει να ελέγξετε αν είναι Smith ή όχι είναι  $\sigma^1$

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

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

Το πρόγραμμά σας να εκτυπώνει εκείνους τους αριθμούς από τους 100000 (που είναι η τιμή της συμβολικής σταθεράς `COMPUTATIONS`) που ελέγξατε οι οποίοι είναι Smith και, επίσης, να υπολογίζει και να εκτυπώνει το ποσοστό των Smith αριθμών που βρέθηκαν.

Μία ενδεικτική εκτέλεση<sup>2</sup> του προγράμματος φαίνεται στη συνέχεια:

```
$ ./smith
Current time is 1389683750

82220161 is a Smith number
5720989 is a Smith number
385160799 is a Smith number
```

<sup>1</sup>Με `mod` συμβολίζουμε το υπόλοιπο διαίρεσης.

<sup>2</sup>Η συγκεκριμένη εκτέλεση έγινε σε μηχανήμα Linux του Τμήματος. Σε άλλες πλατφόρμες (π.χ. PC/Dev-C++) τα αποτελέσματα ενδέχεται να διαφέρουν, διότι η γεννήτρια τυχαίων αριθμών μπορεί να παράγει διαφορετική ακολουθία αριθμών, ακόμα και με το ίδιο “φύτρο”.

```
347971188 is a Smith number
26191849 is a Smith number
287176552 is a Smith number
565162789 is a Smith number
651461690 is a Smith number
141972331 is a Smith number
127583625 is a Smith number
.....
68905761 is a Smith number
494992376 is a Smith number
35179965 is a Smith number
151631349 is a Smith number
168782716 is a Smith number
24240271 is a Smith number
130494597 is a Smith number
190936705 is a Smith number
282476263 is a Smith number
889592501 is a Smith number
```

Found 2.21% Smith numbers

\$

Η παράδοση της άσκησης αυτής συνίσταται στην υποβολή του πηγαίου αρχείου `smith.c` με διαδικασία που θα ανακοινωθεί σύντομα.

**Υπόδειξη:** Σε πρώτη φάση, αγνοήστε την απαίτηση της εκφώνησης για την παραγωγή τυχαίων αριθμών και επικεντρωθείτε στη διατύπωση ενός αλγορίθμου, κατ' αρχήν σε ψευδογλώσσα και στη συνέχεια σε C, που επιλύει το πρόβλημα του ελέγχου αν ένας δεδομένος αριθμός είναι Smith ή όχι. Το ποιον αριθμό θα ελέγχει το πρόγραμμα που θα γράψετε αρχικά μπορείτε να το ορίζετε μέσω `#define`. Αφού μετά από αρκετές δοκιμές βεβαιωθείτε ότι το πρόγραμμά σας δουλεύει σωστά, μπορείτε να το επεκτείνετε ώστε να εξετάζει ποιοι αριθμοί από το 200000001 έως και το 200100000, είναι αριθμοί Smith. Όταν θα σας ικανοποιεί η απόδοση του προγράμματός σας, μπορείτε να το τροποποιήσετε ώστε να κάνει ακριβώς αυτό που ζητά η εκφώνηση, δηλαδή τον έλεγχο για το σύνολο των 100000 αριθμών που θα γεννηθούν με τυχαίο τρόπο. Με άλλα λόγια, μην δοκιμάσετε ταυτόχρονα να αντιμετωπίσετε το βασικό αλγοριθμικό πρόβλημα της άσκησης, να ασχοληθείτε με την αποδοτικότητά του και να επιλύσετε τα τεχνικά προβλήματα σχετικά με τη χρήση των `srand`, `rand` και `time`.

**Σημείωση:** Στην άσκηση αυτή απαγορεύεται αυστηρά η χρήση πινάκων.