

# ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2008-09)

## Άσκηση 1

Ένας θετικός ακέραιος αριθμός ονομάζεται *ελεύθερος-τετραγώνου* (square-free), ή ET για τη συνέχεια, όταν δεν έχει κανένα διαιρέτη που να είναι τέλειο τετράγωνο, εκτός, φυσικά, από το 1. Για παράδειγμα το 30 είναι ET αριθμός, ενώ το 18 δεν είναι (αφού έχει σαν διαιρέτη το 9 που είναι τέλειο τετράγωνο).

Γράψτε ένα πρόγραμμα C (έστω ότι το πηγαίο αρχείο του ονομάζεται “sqfree.c”) το οποίο να επιλέγει με τυχαίο τρόπο 1000 θετικούς ακεραίους αριθμούς και να βρίσκει για καθένα απ’ αυτούς αν είναι ET ή όχι. Το πλήθος των αριθμών που θα ελεγχθούν (1000) να μην τοποθετηθεί απ’ ευθείας μέσα στο πρόγραμμά σας, αλλά να ορισθεί μέσω `#define` σαν τιμή της συμβολικής σταθεράς `NUMBERS`. 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> (ή σελίδες 45–46 των διαφανειών του μαθήματος, στο <http://www.di.uoa.gr/~ip/K04.pdf>). Λεπτομέρειες για τη χρήση των συναρτήσεων `srand` και `rand` μπορείτε να μάθετε δίνοντας στα Suns του Τμήματος “`man srand`” ή “`man rand`” (πιθανότατα, σε άλλα συστήματα Unix, όπως για παράδειγμα στα μηχανήματα Linux του Τμήματος, να χρειάζεται “`man 3 srand`” ή “`man 3 rand`”).

Έστω ότι γεννάτε έναν τυχαίο αριθμό  $x$  με χρήση της συνάρτησης `rand`. Ο αριθμός που θα πρέπει να ελέγξετε αν είναι ET ή όχι είναι ο  $y = (x \bmod 32768)^2 + 1$ . Για την αρχικοποίηση της γεννήτριας των τυχαίων αριθμών, να χρησιμοποιήσετε την τρέχουσα ώρα, μέσω της συνάρτησης `time`<sup>1</sup>, όπως ακριβώς γίνεται και στο πρόγραμμα `gcdlcm.c` που προαναφέρθηκε.

Για κάθε αριθμό  $y$ , από τους 1000 (ή όποια είναι η τιμή της συμβολικής σταθεράς `NUMBERS`) που θα ελέγχετε, να εκτυπώνετε το κατάλληλο μήνυμα και στο τέλος να υπολογίσετε και να εκτυπώσετε το ποσοστό των ET αριθμών που βρέθηκαν.

Τμήμα μίας ενδεικτικής εκτέλεσης<sup>2</sup> του προγράμματος φαίνεται στη συνέχεια:

```
% ./sqfree
Current time is 1224756321

609299857 is square-free
361874530 is square-free
336722501 is square-free
394419601 is square-free
 52200626 is square-free
877699877 is square-free
414488882 is square-free
500909162 is square-free
142134085 is square-free
203062501 is square-free
947469962 is square-free
```

<sup>1</sup>Πληροφοριακά, η τιμή που επιστρέφει η `time` ισούται με το πλήθος των δευτερολέπτων που έχουν περάσει από την 1/1/1970, ώρα 00:00, μέχρι τη στιγμή που κλήθηκε.

<sup>2</sup>Η συγκεκριμένη εκτέλεση έγινε σε μηχανήματα Sun του Τμήματος. Στα μηχανήματα Linux του Τμήματος, η εκτέλεση, για το δεδομένο “φύτρο” (1224756321) της γεννήτριας τυχαίων αριθμών, δίνει ποσοστό ET αριθμών ίσο με 88.50%. 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”.

```
827597825 is not square-free
406425601 is square-free
114982730 is square-free
943656962 is square-free
150307601 is square-free
136095557 is square-free
 13162385 is square-free
204976490 is square-free
476636225 is not square-free
721137317 is square-free
.....
1035423685 is square-free
 247590226 is square-free
 154455185 is square-free
 486070210 is square-free
 371409985 is square-free
```

Found 87.90% square-free numbers

Η παράδοση της άσκησης αυτής συνίσταται στην υποβολή του πηγαίου αρχείου `sqfree.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 ΕΤ 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 κάνει τον έλεγχο για το σύνολο των αριθμών που θα γεννηθούν με τυχαίο τρόπο. Με άλλα λόγια, μην δοκιμάσετε να αντιμετωπίσετε ταυτόχρονα το βασικό αλγοριθμικό πρόβλημα της άσκησης και τα τεχνικά προβλήματα σχετικά με τη χρήση των `srand`, `rand` και `time`.