

# ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2011-12)

## Άσκηση 1

Έστω ότι δίνεται ένας θετικός ακέραιος  $n = n_0$ . Ορίζουμε την ακολουθία  $n_0, n_1, n_2, n_3 \dots$ , όπου κάθε όρος  $n_{i+1}$  ισούται με το άθροισμα των τετραγώνων των ψηφίων του  $n_i$ . Για  $n = 19$ , η συγκεκριμένη ακολουθία είναι η:

19, 82, 68, 100, 1, 1, 1, ...

Είναι προφανές ότι αν ένας όρος της ακολουθίας ισούται με 1, τότε και όλοι οι υπόλοιποι, μετά από αυτόν, θα είναι 1. Οι θετικοί ακέραιοι, όπως το 19, για τους οποίους ισχύει ότι η ακολουθία που δημιουργείται σύμφωνα με τον προηγούμενο ορισμό, έχει από ένα σημείο και μετά σαν μόνο όρο το 1, ονομάζονται *ευτυχισμένοι αριθμοί* (*happy numbers*).

Όμως, για  $n = 29$ , η ακολουθία που ορίσαμε θα ήταν η:

29, 85, 89, 145, 42, 20, 4, 16, 37, 58, 89, 145, 42, 20, 4, 16, 37, 58, 89, 145, ...

Παρατηρήστε ότι στο παράδειγμα αυτό, η ακολουθία εγκλωβίστηκε σε έναν κύκλο που αποτελείται από τους αριθμούς 89, 145, 42, 20, 4, 16, 37, 58, άρα ποτέ δεν πρόκειται να έχει σαν όρο το 1. Γι' αυτόν το λόγο, το 29 είναι *δυστυχισμένος αριθμός* (*unhappy numbers*). Αποδεικνύεται ότι για κάθε αριθμό  $n$ , στην εν λόγω ακολουθία είτε θα επαναλαμβάνεται συνεχόμενα από κάποιο σημείο και μετά ο αριθμός 1, οπότε το  $n$  είναι ευτυχισμένος, είτε αυτή θα εγκλωβιστεί στον προαναφερθέντα κύκλο αριθμών, οπότε το  $n$  είναι δυστυχισμένος. Οι ευτυχισμένοι αριθμοί μέχρι το 100 είναι οι εξής:

1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49, 68, 70, 79, 82, 86, 91, 94, 97, 100

Μία ειδική περίπτωση ευτυχισμένων αριθμών είναι εκείνοι που είναι ταυτόχρονα και πρώτοι (δηλαδή έχουν σαν μόνους διαιρέτες το 1 και τον εαυτό τους). Οι *ευτυχισμένοι πρώτοι* (ΕΠ) μέχρι το 300 είναι οι εξής:

7, 13, 19, 23, 31, 79, 97, 103, 109, 139, 167, 193, 239, 263, 293

Γράψτε ένα πρόγραμμα C (έστω ότι το πηγαίο αρχείο του ονομάζεται "happypr.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 ΕΠ. 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> (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 μηχανήματα Linux του Τμήματος "man 3 srand" ή "man 3 rand" (πιθανότατα, σε άλλα συστήματα Unix, όπως για παράδειγμα στα μηχανήματα Sun του Τμήματος, 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 ΕΠ ή όχι είναι ο<sup>1</sup>

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

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

Για την αρχικοποίηση της γεννήτριας των τυχαίων αριθμών, να χρησιμοποιήσετε την τρέχουσα ώρα, μέσω της συνάρτησης `time2`, όπως ακριβώς γίνεται και στο πρόγραμμα `gcdlcm.c` που προαναφέρθηκε.

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

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

```
% ./happypr
Current time is 1321818703
```

```
217193689 is happy prime
 35155961 is happy prime
191045471 is happy prime
209637191 is happy prime
136410991 is happy prime
 14818561 is happy prime
 42281131 is happy prime
27172009 is happy prime
48724589 is happy prime
```

```
.....
 19194631 is happy prime
 77955503 is happy prime
191429839 is happy prime
268636201 is happy prime
119608301 is happy prime
387215489 is happy prime
182265067 is happy prime
 66811001 is happy prime
 79373881 is happy prime
```

```
Found 1.44% happy prime numbers
```

```
%
```

Η παράδοση της άσκησης αυτής συνίσταται στην υποβολή του πηγαίου αρχείου `happypr.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 ΕΠ 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`.

<sup>2</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>3</sup>Η συγκεκριμένη εκτέλεση έγινε σε μηχανήμα Linux του Τμήματος. Στα μηχανήματα Sun του Τμήματος, η εκτέλεση, για το δεδομένο “φύτρο” (1321818703) της γεννήτριας τυχαίων αριθμών, δίνει ποσοστό ΕΠ αριθμών ίσο με 1.37%. 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”.