ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εαρινό 2007

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι

Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής αποστολής: 21/6/2007, 12:30

Το πρόβλημα 3n+1: η εικασία του Collatz (0.2 + 0.2 = 0.4 βαθμοί)

Στα μαθηματικά κάποιες προτάσεις έχουν απλή διατύπωση αλλά πολύπλοκη απόδειξη. Το παρακάτω πρόβλημα αποτελεί απλή απόδειξη της προηγούμενης πρότασης.

> *Διαλέξτε στην τύχη έναν οποιονδήποτε φυσικό αριθμό.*
> *Εάν είναι άρτιος, διαιρέστε τον με το 2.*
> *Εάν είναι περιττός, πολλαπλασιάστε τον με το 3 και προσθέστε 1.*
> *Επαναλάβετε την παραπάνω διαδικασία για τον αριθμό που προκύπτει.*
> *Τελικά, ο αριθμός θα καταλήξει να είναι ο 1. (Οπότε σταματάμε.)*

Για παράδειγμα, έστω ο αριθμός 3. Η παραπάνω διαδικασία θα παραγάγει την ακολουθία των αριθμών 3, 10, 5, 16, 8, 4, 2 και 1. Τώρα προσπαθήστε ξανά με τον αγαπημένο σας φυσικό αριθμό. Καταλήξατε σε 1 πάλι, έτσι δεν είναι; Κάντε το ξανά για εξάσκηση, αυτή τη φορά με το φυσικό αριθμό που δε συμπαθείτε αρκετά. Ο δικός μου είναι ο 19, και με δυσκόλεψε κάπως περισσότερο: 19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Μπορείτε να χάσετε πολλές ώρες παίζοντας με αυτό το πρόβλημα, αλλά δεν είστε οι μόνοι. Πολλοί μαθηματικοί προσπάθησαν να αποδείξουν ότι η ακολουθία πάντα τελειώνει σε 1 (ή να βρουν κάποιον αριθμό για τον οποίο δεν ισχύει). Το πρόβλημα αυτό είναι γνωστό με πολλά ονόματα. Θα χρησιμοποιήσουμε το όνομα η εικασία του Collatz: "Η διαδικασία που περιγράφεται πιο πάνω τελικώς θα φτάσει πάντα στον αριθμό 1, ανεξάρτητα του φυσικού αριθμού από τον οποίο ξεκίνησε."

Η Wikipedia γράφει τα παρακάτω για αυτό το πρόβλημα.

> *The Collatz conjecture is an unsolved conjecture in mathematics. It is named after Lothar Collatz, who first proposed it in 1937. The conjecture is also known as the 3n + 1 conjecture, the Ulam conjecture (after Stanislaw Ulam), the Syracuse problem, as the hailstone sequence or hailstone numbers, or as Wondrous numbers as per Gödel, Escher, Bach. It asks whether a certain kind of number sequence always ends in the same way, regardless of the starting number.*
>
> *Paul Erdős said about the Collatz conjecture, "Mathematics is not yet ready for such problems." He offered $500 for its solution.*

Μπορείτε να προσπαθήσετε για το χρηματικό βραβείο στον ελεύθερό σας χρόνο, αλλά για την άσκηση των γλωσσών προγραμματισμού θα πρέπει να κάνετε κάτι διαφορετικό.

Ονομάζουμε *μήκος κύκλου Collatz* ενός φυσικού αριθμού το πλήθος των αριθμών που πρέπει να εξεταστούν μέχρι να εμφανιστεί ο αριθμός 1. Για παράδειγμα, το μήκος κύκλου Collatz του 3 είναι 8 και του 19 είναι 21. Αυτό που η άσκηση ζητάει είναι να γραφούν δύο προγράμματα (ένα σε γλώσσα C και ένα σε ML) που δεχόμενα δύο φυσικούς αριθμούς i και j να υπολογίζουν το μέγιστο μήκος κύκλου Collatz όλων των φυσικών μεταξύ (και συμπεριλαμβανομένων) των i και j.

Παρακάτω, δείχνουμε μερικά παραδείγματα από τη χρήση του προγράμματος σε C και αυτού σε ML. (Παρακαλούμε ονοματίστε το εκτελέσιμο πρόγραμμα σας σε C και τις αρχικές συναρτήσεις σας στην ML με τα παρακάτω ονόματα.)

C	ML
`> tnp1 1 10`	`- tnp1 1 10;`
`20`	`val it = 20 : int`
`> tnp1 900 1000`	`- tnp1 900 1000;`
`174`	`val it = 174 : int`
`> tnp1 80000 100000`	`- tnp1 80000 100000;`
`333`	`val it = 333 : int`
`> tnp1 100 1000000`	`- tnp1 100 1000000;`
`525`	`val it = 525 : int`
`> tnp1 100 100000000`	`- tnp1 100 100000000;`
`950`	`val it = 950 : int`

Στο πρόγραμμά σας μπορείτε να υποθέσετε ότι οι τιμές είναι φυσικοί αριθμοί, ο πρώτος είναι μικρότερος ή ίσος από τον δεύτερο, και ο δεύτερος είναι το πολύ 1000000000 (ένα δισεκατομμύριο). Για τόσο “μικρούς” αριθμούς, η εικασία του Collatz ισχύει, άρα το πρόγραμμα τερματίζει. Θα παρατηρήσατε βεβαίως τη γραμμή στα αποτελέσματα. Δεν πρόκειται για δαίμονα του τυπογραφείου, υπάρχει για κάποιο λόγο που κατά πάσα πιθανότητα θα καταλάβετε μόνοι σας όπως θα προγραμματίζετε τη λύση του προβλήματος. Στο σημείο εκείνο θα πρέπει να κατανοήσετε λίγο βαθύτερα τη C και την ML και, για να πετύχετε το σκοπό σας, να χρησιμοποιήσετε κάποιες από τις δυνατότητες που προσφέρουν (πιθανώς διαφορετικές σε κάθε γλώσσα).

Στην ιστοσελίδα υπάρχει ένα μέρος του προγράμματος σε C. Χρησιμοποιήστε το για τη λύση σας. Προγράμματα που δε δουλεύουν για όλες τις τιμές μέχρι το 1000000000 είτε στη C είτε στην ML, θα βαθμολογηθούν με τους μισούς το πολύ πόντους της άσκησης.

Ο Βρασίδας ήπιε λιγάκι παραπάνω (0.2 + 0.2 = 0.4 βαθμοί)

Ο Βρασίδας είναι ένας προγραμματιστής που γράφει προγράμματα ελέγχου ενός ρομπότ το οποίο χρησιμοποιείται ως “ζωγράφος” για πατώματα. Το ρομπότ ξεκινάει τη δουλειά του πάντα από το σημείο με συντεταγμένες (0,0) σε ένα επίπεδο και κοιτάει προς το (0,∞). Οι εντολές που καταλαβαίνει το ρομπότ είναι δύο και πολύ απλές:

- η εντολή **s** που κάνει το ρομπότ να προχωρήσει ένα βήμα (δηλαδή αυξήσει ή μειώσει κατά ένα την τιμή της x ή της y συντεταγμένης) προς την κατεύθυνση που εκείνη τη στιγμή κοιτάει, και
- η εντολή **r** που στρέφει το ρομπότ 90 μοίρες προς την κατεύθυνση του ρολογιού.

Το ρομπότ όταν προχωρά προς μία κατεύθυνση αφήνει μια λωρίδα χρώματος στο πάτωμα. Οι λωρίδες αυτές έχουν όλες το ίδιο πάχος και ύψος ανεξάρτητα από πόσες φορές το ρομπότ περνάει από πάνω τους.

Ο Βρασίδας χθες το βράδυ βγήκε να τα πιει με τους φίλους του και γύρισε σπίτι λιγάκι μεθυσμένος. Αντί να πέσει να κοιμηθεί, άρχισε να γράφει προγράμματα για το ρομπότ δίνοντας εντολές **s** και **r** με εντελώς τυχαίο τρόπο. Περιέργως, κάποια από τα αποτελέσματα των λωρίδων που ζωγράφισε το ρομπότ στο πάτωμα είχαν εικαστικό ενδιαφέρον. Ο Βρασίδας θέλει να τα χρησιμοποιήσει, αλλά δε θέλει να δείξει στο αφεντικό του προγράμματα μεγάλου μήκους και με προφανείς πλεονασμούς. Θέλει λοιπόν να γράψει το ίδιο πρόγραμμα σε δύο γλώσσες, ένα σε ML και ένα σε C, που θα δέχεται ως είσοδο ένα πρόγραμμα του ρομπότ με πλεονασμούς και θα επιστρέφει το μικρότερο δυνατό πρόγραμμα που κάνει το ρομπότ να σχεδιάσει ακριβώς το ίδιο σύνολο λωρίδων στο πάτωμα, πιθανώς με διαφορετική σειρά, ξεκινώντας όμως πάντα από το σημείο (0,0).

Μερικά από τα αποτελέσματα του προγράμματος σε ML δείχνουν ως εξής:

```
- vrasidas [r,r,r,r,r,r,r,r,r,r,r];
val it = [] : command list
- vrasidas [r,r,s,s,r,r,s,s,r,r,r];
val it = [r,r,s,s] : command list
 - vrasidas [r,r,r,s,s,r,r,s,s,s,r,r,r,r,r,r,r];
val it = [r,s,r,r,s,s,s] : command list
```

Αν σας βοηθάει κάπου, υποθέστε ότι τα προγράμματα με πλεονασμό που έγραψε ο Βρασίδας το βράδυ που μέθυσε έχουν το πολύ 1000 εντολές. Το πρόγραμμά σας σε C πρέπει να δέχεται τα προγράμματα ως command line strings (π.χ. το πρώτο πρόγραμμα ως **"rrrrrrrrrrr"**, φυσικά χωρίς αποστρόφους) και να τα τυπώνει επίσης ως strings ίδιας μορφής**.**

Θέματα συζήτησης (0.1 + 0.1 = 0.2 βαθμοί)

Συζητήστε σύντομα τα παρακάτω θέματα:

- Τι γνώμη έχετε για την υλοποίηση των ακεραίων στη C και στην ML; Τι κάνατε για να παρακάμψετε το πρόβλημα σε κάθε γλώσσα; Ποια λύση σας φαίνεται πιο εύκολη και γιατί;
- Τι γνώμη έχετε για την ευκολία χειρισμού ακολουθιών δεδομένων ως λιστών στην ML και ως συμβολοσειρών στη C; Για να βοηθήσετε τον Βρασίδα, χρησιμοποιήσατε τον ίδιο αλγόριθμο και στις δύο γλώσσες; Αν υπάρχουν (μικρό)διαφορές ποιος τρόπος επίλυσης σας φαίνεται πιο προσιτός και γιατί;

Οδηγίες για την αποστολή των λύσεων

- Μπορείτε να δουλέψετε σε ομάδες το πολύ 2 ατόμων
- Δεν επιτρέπεται να μοιράζεστε ασκήσεις με άλλους συμφοιτητές σας ή να βάλετε τις ασκήσεις σας σε μέρος που άλλοι μπορούν να τις βρουν εύκολα (π.χ. στο διαδίκτυο, σε ιστοτόπους συζητήσεων, …)
- Τα προγράμματα σε C πρέπει να είναι σε ένα αρχείο και να μπορούν να μεταγλωττιστούν με gcc 4.x.x (ή 3.x.x) με μια εντολή της μορφής: **`gcc -O3 -o file file.c`**
- Τα προγράμματα σε ML πρέπει να δουλεύουν σε SML/NJ
- Η αποστολή των ασκήσεων θα πρέπει να γίνει ηλεκτρονικά μέσω του `moodle`. Το καλύτερο είναι να στείλετε ένα `.gz` ή `.zip` αρχείο (παρακαλούμε όχι `.rar`) το οποίο να περιέχει τα 4 αρχεία με τα προγράμματα, ένα αρχείο με όνομα AUTHORS που να έχει τα ονόματα και τους αριθμούς μητρώων σας, ένα αρχείο `DISCUSSION` που να απαντάει σύντομα στα παραπάνω θέματα συζήτησης, και προαιρετικά ένα αρχείο README εάν υπάρχει κάτι το οποίο είναι άξιο ανάγνωσης που αφορά στο πως τα προγράμματα πρέπει να μεταγλωττιστούν και να τρέξουν.