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

Εαρινό 2008

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι

Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 14/5/2008, 12:30

Εαυτοί αριθμοί (0.2 + 0.2 = 0.4 βαθμοί)

Το 1949 ο Ινδός μαθηματικός D.R. Kaprekar ανακάλυψε μια κλάση αριθμών τους οποίους ονόμασε *εαυτούς αριθμούς (self numbers)*. Για κάθε θετικό ακέραιο n, ορίζουμε τη συνάρτηση d(n) ως τον αριθμό n συν το άθροισμα των ψηφίων του. Το γράμμα d αναφέρεται στην *ψηφιοποίηση* του αριθμού, έναν όρο που επίσης εισήγαγε ο Kaprekar. Για παράδειγμα, d(75) = 75 + 7 + 5 = 87. Αρχίζοντας από έναν αριθμό n, μπορούμε να κατασκευάσουμε μια άπειρη αύξουσα ακολουθία από αριθμούς n, d(n), d(d(n)), d(d(d(n))), ... Για παράδειγμα, αν αρχίσουμε από τον 33, ο επόμενος αριθμός είναι ο 33 + 3 + 3 = 39, ο επόμενος ο 39 + 3 + 9 = 51, μετά ο 51 + 5 + 1 = 57, και έτσι γεννιέται η ακολουθία 33, 39, 51, 57, 69, 84, 96, 111, 114, 120, 123, 129, 141, ...

Ο αριθμός n λέγεται *γεννήτρια* του d(n). Στην παραπάνω ακολουθία, ο 33 είναι μια γεννήτρια του 39, ο 39 είναι μια γεννήτρια του 51, ο 51 του 57, κοκ. Κάποιοι αριθμοί έχουν περισσότερες από μία γεννήτριες: για παράδειγμα, ο 101 έχει δύο γεννήτριες, τον 91 και τον 100. Ένας αριθμός χωρίς γεννήτριες είναι ένας εαυτός αριθμός. Υπάρχουν δεκατρείς εαυτοί αριθμοί μικρότεροι του 100: 1, 3, 5, 7, 9, 20, 31, 42, 53, 64, 75, 86, και 97.

Αυτό που ζητάει η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε C και ένα σε ML) το καθένα από τα οποία παίρνοντας ως είσοδο δύο αριθμούς i και j να τυπώνει το πλήθος των εαυτών αριθμών μεταξύ (και συμπεριλαμβανομένων) των i και j.

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

C

```
> self 1 100
13
> self 9900 10000
10
```

ML

```
- self 1 100;
val it = 13 : int
- self 9900 10000;
val it = 10 : int
```

Μπορείτε να υποθέσετε ότι οι αριθμοί εισόδου είναι φυσικοί, ο πρώτος είναι μικρότερος ή ίσος από το δεύτερο και ο δεύτερος είναι το πολύ 100.000.000 (εκατό εκατομμύρια). Στην ιστοσελίδα υπάρχει ένα μέρος του προγράμματος σε C. Χρησιμοποιήστε το για τη λύση σας αν σας κάνει τη ζωή πιο εύκολη.

Η γελάδα ποτέ δεν πεθαίνει...[1] (0.3 + 0.3 = 0.6 βαθμοί)

Η Κλάρα είναι μια αγελάδα γνωστή και από την τηλεόραση. Όμως η καημένη, όπως και δυστυχώς πολλοί άλλοι σε αυτήν τη χώρα, έχει μείνει άνεργη εδώ και καιρό. Της πρότειναν να συμμετάσχει σε ένα νέο τηλεπαιχνίδι που αναμένεται να θέσει νέα ρεκόρ τηλεθέασης στην Ελληνική trash TV. Το νέο τηλεπαιχνίδι θα ονομάζεται "Η γελάδα ποτέ δεν πεθαίνει!" και θα είναι το εξής: Η Κλάρα θα προσπαθεί απλώς να κάνει βόλτα στο πλατό του στούντιο ενώ το συμμετέχον κοινό θα πετάει ντομάτες προς το μέρος της. Το μέλος του κοινού που θα καταφέρει να πετύχει ή ακόμα και να λερώσει την Κλάρα με τη ντομάτα του θα κερδίζει κάποιο έπαθλο.

Στο τηλεπαιχνίδι, η Κλάρα θα ξεκινάει πάντα από μια από τις τέσσερις γωνίες του πλατό του στούντιο. Ας θεωρήσουμε αυτό το σημείο ως το σημείο με συντεταγμένες (0,0). Τότε, το πλατό, αν και πεπερασμένο σε μέγεθος, μπορεί να θεωρηθεί ως ένα τεταρτημόριο οι πλευρές του οποίου εκτείνονται προς το άπειρο. Το κοινό πετάει ντομάτες που προσγειώνονται σε κάποια χρονική στιγμή σε κάποιο σημείο του πλατό και λερώνουν αυτό το σημείο και τα (συνήθως 4) γειτονικά του, δηλαδή αυτά που βρίσκονται πάνω, κάτω, αριστερά και δεξιά του (αν υπάρχουν). Η Κλάρα, η οποία προσπαθεί βεβαίως να μη λερωθεί, σκοπό της έχει να μη βρεθεί σε σημείο στο οποίο θα πέσει ή που έχει ήδη υπολείμματα από ντομάτες. (Φυσικά, μπορεί να βρεθεί σε κάποιο σημείο πριν αυτό λερωθεί.)

Οι παραγωγοί του "υπερθεάματος" θέλουν να υπολογίσουν από πριν τον τηλεοπτικό χρόνο της κάθε βόλτας της Κλάρα έτσι ώστε να ξέρουν πότε θα ρίξουν διαφημίσεις. Για το λόγο αυτό ζητάνε από κάθε μέρος του κοινού να τους πει από πριν σε ποιο σημείο του πλατό θα ρίξει τη ντομάτα του και σε ποια χρονική στιγμή του σόου. Για να δικαιολογήσουν τον τίτλο της εκπομπής και να μη δίνουν πολλά δώρα στους νικητές, αυτή την πληροφορία θα τη δίνουν στην Κλάρα έτσι ώστε να έχει την ευκαιρία να ξεφύγει από τις ντομάτες. Με άλλα λόγια, η Κλάρα στην αρχή της κάθε βόλτας θα λαμβάνει ως είσοδο μια λίστα από συντεταγμένες και χρονικές στιγμές που θα πέσουν ντομάτες σε κάποιο σημείο. Σκοπός της Κλάρα είναι να αποφύγει να λερωθεί, παρόλο που αυτό δεν είναι πάντα δυνατό. Επιπλέον, επειδή η Κλάρα βαριέται να περπατάει, θέλει να υπολογίσει τον ελάχιστο αριθμό βημάτων που χρειάζεται για να βρεθεί σε κάποιο σημείο του πλατό ασφαλές μέχρι τη ρίψη της τελευταίας ντομάτας. Ένα σημείο είναι ασφαλές σε κάποια χρονική στιγμή T εάν δε λερώνεται ή δεν είναι ήδη λερωμένο εκείνη τη στιγμή. Η Κλάρα, σε κάθε χρονική στιγμή μπορεί είτε να μείνει στο σημείο που βρίσκεται είτε να κάνει ένα βήμα προς ένα άλλο σημείο του πλατό που συνορεύει (πάνω, κάτω, αριστερά και δεξιά) με αυτό στο οποίο βρίσκεται.

Θεωρείστε ότι ο χρόνος είναι διακριτός. Τα μέλη του κοινού δε συνεννοούνται μεταξύ τους οπότε είναι πιθανό να πέσουν περισσότερες από μία ντομάτες σε κάποιο σημείο.

Γράψτε ένα πρόγραμμα σε C και ένα σε ML που να βοηθάει την Κλάρα να υπολογίσει τον ελάχιστο αριθμό βημάτων που αρκούν για να βρεθεί σε ένα ασφαλές σημείο. Το πρόγραμμα πρέπει είτε να επιστρέφει αυτόν τον αριθμό είτε -1 εάν δεν υπάρχει κάποιο τέτοιο ασφαλές σημείο.

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

```
- gelada [(0,0,2), (2,1,2), (1,1,2), (0,3,5)];
val it = 5 : int
- gelada [(1,1,1), (3,1,4)];
val it = 0 : int
 - gelada [(0,0,1)];
val it = -1 : int
```

Εξηγούμε το πρώτο από αυτά. Η είσοδός του λέει ότι θα πέσουν κάποιες ντομάτες στα σημεία (0,0),

1 Ευχαριστώ την ιστοσελίδα http://crazycows.wordpress.com/ τόσο για το γέλιο που έχω ρίξει διαβάζοντάς την όσο και για την έμπνευση του τίτλου της άσκησης.

(2,1) και (1,1) τη χρονική στιγμή 2 και μια ντομάτα στο σημείο (0,3) τη χρονική στιγμή 5. Αυτό σημαίνει ότι τις δύο αυτές χρονικές στιγμές τα λερωμένα τετράγωνα είναι αυτά που φαίνονται παρακάτω (με * δείχνουμε το σημείο πτώσης και το # τις παράπλευρες απώλειες κάποιας ντομάτας):

5						
4						
3						
2		#	#			
1	#	*	*	#		
0	*	#	#			
	0	1	2	3	4	5

Table 1: T = 2

5						
4	#					
3	*	#				
2	#	#	#			
1	#	#	#	#		
0	#	#	#			
	0	1	2	3	4	5

Table 2: T = 5

Μελετώντας τα σχεδιαγράμματα βλέπουμε ότι το κοντινότερο ασφαλές σημείο είναι το (3,0) αλλά ο δρόμος προς αυτό μπλοκάρεται πολύ γρήγορα. Το επόμενο ασφαλές σημείο είναι το (4,0) το οποίο έχει επίσης το ίδιο πρόβλημα. Τα κοντινότερα στη συνέχεια είναι τα σημεία στη (0,5)-(5,0) διαγώνιο. Από αυτά οποιοδήποτε από τα (0,5), (1,4) και (2,3) είναι προσπελάσιμα σε 5 χρονικές μονάδες.

Μπορείτε να υποθέσετε ότι όλοι οι αριθμοί εισόδου είναι τέτοιοι που χωράνε σε έναν **`unsigned int`** της C (σε 32 bits). Στη C, το πρόγραμμά σας πρέπει να διαβάζει την είσοδο από το stdin ως εξής: Η πρώτη γραμμή έχει τον αριθμό από γραμμές που ακολουθούν και οι υπόλοιπες γραμμές έχουν τις χρόνο-συντεταγμένες. Δηλαδή η είσοδος για την πρώτη περίπτωση είναι:

```
4
0 0 2
2 1 2
1 1 2
0 3 5
```

Η αντίστοιχη έξοδος είναι:

```
5
```

Περαιτέρω οδηγίες για την άσκηση

- 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 με μια εντολή της μορφής: **`gcc -O3 -o file file.c`**
- Τα προγράμματα σε ML πρέπει να δουλεύουν σε SML/NJ
- Η αποστολή των προγραμμάτων θα γίνει ηλεκτρονικά. Θα υπάρξει σχετική ανακοίνωση στο `moodle` για την ακριβή διαδικασία υποβολής.