



## Θ.Ε. ΠΛΣ50 (2013-14) – ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ Ε1

Ημερομηνία ανάρτησης	05.10.2013
Ημερομηνία αποστολής	10.11.2013 (δεκτή μέχρι την Τρίτη 12.11.2013, ώρα 23:59)
Ανακοίνωση ενδεικτικής επίλυσης	15.11.2013

### Θεματολογία-στόχος

Στην εργασία αυτή θα εξασκηθείτε σε μερικές από τις απαραίτητες γνώσεις της C που θα χρησιμοποιήσουμε, καθ' όλη τη χρονιά, σε πολλές από τις εργασίες της ΘΕ. Ειδικότερα, θα ασχοληθείτε με δομές ελέγχου ροής προγράμματος (διακλαδώσεις, επαναλήψεις κλπ), είσοδο από πληκτρολόγιο και έξοδο σε οθόνη, είσοδο από αρχείο, μονοδιάστατους πίνακες, συναρτήσεις, κ.ά.

### Παρατηρήσεις

Περιμένουμε όλες οι εργασίες να ανεβούν στο study.eap.gr, όπως σας έχει υποδείξει ο σύμβουλος καθηγητής του τμήματός σας, και να είναι γραμμένες σε επεξεργαστή κειμένου σε μορφή doc ή odt (αρχεία pdf γίνονται δεκτά μόνο όταν συνοδεύονται από το αντίστοιχο doc/odt), όπου θα συμπεριλάβετε προαιρετικές επεξηγήσεις για τον τρόπο που λύσατε την εργασία. Ο πηγαίος κώδικας C (ένα πρόγραμμα C για κάθε θέμα, ή υποερώτημα) που θα συνοδεύει την εργασία θα πρέπει να βρίσκεται σε ξεχωριστά αρχεία (\*.c) (ένα αρχείο .c για κάθε θέμα, ή υποερώτημα). Μην συμπεριλαμβάνετε εκτελέσιμα (.exe) ή αρχεία object code (.o). Στο study.eap.gr, σε κάθε περίπτωση, ανεβάζετε **ένα μόνο συμπιεσμένο αρχείο** (.zip ή .rar) που θα περιέχει όλα τα επιμέρους αρχεία.

### Εισαγωγή

Ο πηγαίος κώδικας του προγράμματός σας θα πρέπει να είναι επαρκώς σχολιασμένος και επιπλέον ANSI-συμβατός (δεν γίνονται δεκτές απαντήσεις που κάνουν χρήση συναρτήσεων από επεκτάσεις της C). Δε χρειάζεται να κάνετε τίποτε περισσότερο από όσα σας ζητούνται. Συνιστάται, πριν ασχοληθείτε με τη συγκεκριμένη εργασία, να μελετήσετε σχολαστικά τις λύσεις εργασιών προηγούμενων ετών. Στο τέλος της Εργασίας περιλαμβάνεται **ενδεικτικός** πίνακας με τα βασικά γνωστικά αντικείμενα της γλώσσας C που πρέπει να έχετε διαβάσει για να μπορείτε να επεξεργαστείτε κάθε θέμα. Τονίζεται ότι ο πίνακας έχει σκοπό να σας βοηθήσει και όχι να σας υποχρεώσει στον ακριβή τρόπο υλοποίησης κάθε θέματος.

### Θέμα 1: Δυνάμεις ακεραίων – Άθροισμα δυνάμεων

**α)** Να γράψετε πρόγραμμα C που να υπολογίζει τη δύναμη ενός θετικού ακεραίου με εκθέτη έναν επίσης θετικό ακέραιο αριθμό. Ο χρήστης θα εισάγει από το πληκτρολόγιο τη βάση, b, και τον εκθέτη, e, και το πρόγραμμα θα εμφανίζει το αποτέλεσμα της μαθηματικής πράξης  $b^e$  στην οθόνη.

**β)** Μετατρέψτε τον κώδικα του υποερωτήματος (α) σε συνάρτηση, με το παρακάτω πρότυπο:

```
int power(int b, int e)
```

Δηλαδή, η συνάρτηση θα ονομάζεται `power`, θα δέχεται ως είσοδο δύο ακεραίους αριθμούς, έστω b ο πρώτος και e ο δεύτερος, και (εφόσον οι αριθμοί είναι θετικοί) θα επιστρέφει έναν ακέραιο αριθμό που θα ισούται με  $b^e$ . Σε περίπτωση που ένας από τους δύο αριθμούς δεν είναι θετικός, η συνάρτηση θα επιστρέφει 0.

Στη συνέχεια, να γράψετε πρόγραμμα C που να υπολογίζει το παρακάτω άθροισμα:

$$S=1^e+2^e+3^e+\dots+n^e$$

όπου τα n και e είναι θετικοί ακεραίοι αριθμοί που τους εισάγει ο χρήστης από το πληκτρολόγιο, σε σχετικές ερωτήσεις του κυρίως προγράμματος. Για τον υπολογισμό των δυνάμεων το πρόγραμμά σας θα χρησιμοποιεί τη συνάρτηση `power`.



Σε όλες τις περιπτώσεις που το πρόγραμμά σας ζητά από τον χρήστη έναν ακέραιο αριθμό, θεωρείστε ότι ο χρήστης πράγματι εισάγει αριθμό (και όχι, π.χ., κείμενο, κάτι που θα μπορούσε να οδηγήσει σε σφάλμα εκτέλεσης του προγράμματος). Ωστόσο, το πρόγραμμά σας θα πρέπει να ελέγχει αν ο ακέραιος που δόθηκε είναι θετικός και, αν δεν είναι, να τερματίζει άμεσα.

## Θέμα 2: Εκτύπωση Ιστογράμματος

Να γράψετε πρόγραμμα C το οποίο θα ορίζει μονοδιάστατο πίνακα ακεραίων A,  $n=20$  θέσεων (από 0 έως  $n-1$  οι θέσεις). Το τμήμα του πίνακα A που θα χρησιμοποιήσετε θα καθορίζεται από το χρήστη. Δηλαδή το πρόγραμμά σας θα ζητά σαν είσοδο έναν ακέραιο αριθμό k, όπου  $k \leq n$ , ο οποίος θα καθορίζει πόσες θέσεις του πίνακα θα χρησιμοποιήσετε (από 0 έως  $k-1$ ). Στη συνέχεια το πρόγραμμα θα ζητά από το χρήστη k μη αρνητικούς ακεραίους αριθμούς και θα τους καταχωρεί στις πρώτες k θέσεις του πίνακα A. Οι τιμές των αριθμών αυτών θα πρέπει να κυμαίνονται μεταξύ του 0 και του 40. Αν κάποιος αριθμός που δοθεί είναι εκτός των ορίων αυτών, το πρόγραμμα θα τον ξαναζητά.

Στη συνέχεια το πρόγραμμα θα εκτυπώνει ως έξοδο τα ακόλουθα:

- Μία γραμμή στην οποία θα υπάρχουν οι λέξεις `Elements`, `Value` και `Histogram`. Η πρώτη λέξη θα εκτυπωθεί στην αρχή της γραμμής αφού πρώτα αφήσει ένα κενό χαρακτήρα. Κάθε διάστημα μεταξύ των τριών λέξεων θα είναι επτά κενό χαρακτήρες.
- Σε κάθε επόμενη γραμμή, το πρόγραμμα θα εκτυπώνει για κάθε στοιχείο  $A[i]$  του πίνακα,  $0 \leq i < k$ , τα εξής:
  - 1) Στη θέση κάτω από τη λέξη `Elements` (κάτω ακριβώς από το γράμμα `E` της λέξης), θα εκτυπώνει την θέση του πίνακα. Αν για παράδειγμα το πρόγραμμα επεξεργάζεται το πρώτο στοιχείο του πίνακα  $A[0]$ , θα εκτυπώσει την τιμή 1, ενώ για τη θέση  $A[1]$  την τιμή 2, κλπ.
  - 2) Κάτω από τη λέξη `Value` (ακριβώς κάτω από το γράμμα `V` της λέξης), θα εκτυπώνει την τιμή που είναι καταχωρημένη στην τρέχουσα θέση του πίνακα. Για παράδειγμα αν το πρόγραμμα επεξεργάζεται τη θέση  $i=0$  και το  $A[0]$  έχει την τιμή 3, θα εκτυπώσει την τιμή 3.
  - 3) Κάτω από τη λέξη `Histogram` (ακριβώς κάτω από το γράμμα `H` της λέξης), θα εκτυπώνει χαρακτήρες "\*" (αστεράκι). Το πλήθος των χαρακτήρων "\*" που θα εκτυπώνονται θα καθορίζεται από την τιμή που υπάρχει αποθηκευμένη στη θέση  $i$  του πίνακα A ( $A[i]$ ). Αν δηλαδή το πρόγραμμα επεξεργάζεται το πρώτο στοιχείο του πίνακα ( $A[0]$ ) και η τιμή του στοιχείου αυτού είναι 3, θα εκτυπώσει τρία αστεράκια (σε οριζόντια διάταξη, στην ίδια δηλαδή ευθεία).

Για παράδειγμα, αν  $k=4$  και στις 4 πρώτες θέσεις του πίνακα είχαμε τις τιμές  $A[0]=3$ ,  $A[1]=5$ ,  $A[2]=1$  και  $A[3]=7$ , τότε το πρόγραμμά μας θα εκτύπωνε σαν έξοδο:

Element	Value	Histogram
1	3	***
2	5	*****
3	1	*
4	7	*****

## Θέμα 3: Ανάγνωση από αρχείο – Μορφοποιημένη έξοδος στην οθόνη

Να γράψετε πρόγραμμα C που να υλοποιεί τις ακόλουθες διαδικασίες:



- Θα ανοίγει για ανάγνωση ένα αρχείο κειμένου (το οποίο θα δημιουργήσετε εσείς με ένα συντάκτη κειμένου, π.χ. notepad). Το αρχείο αυτό θα περιέχει τυχαίους αριθμούς (δικής σας επιλογής), ακέραιους ή/και πραγματικούς, έναν σε κάθε σειρά, όπως φαίνεται παρακάτω:

```
2.345
-8
3.1415926
73
123.456789
```

- Κατά την ανάγνωση των αριθμών, θα υπολογίζει το πλήθος τους, τη μέγιστη τιμή τους, την ελάχιστη τιμή τους και τον μέσο όρο αυτών. Τα αποτελέσματα αυτά θα εμφανίζονται στην οθόνη, με χρήση της `printf`, με ακρίβεια 4 δεκαδικών ψηφίων και 12 συνολικά ψηφία για κάθε αριθμό, όπως φαίνεται στο παρακάτω παράδειγμα:

```
Tot:      5.0000
Max:     123.4567
Min:     -8.0000
Ave:     38.7887
```

**Σημείωση:** εκτός από το πρόγραμμα C, υποβάλετε και το αρχείο κειμένου με το οποίο το δοκιμάσατε.

## Θέμα 4: Αναζήτηση συμβολοσειράς

Στο θέμα 4 της περυσινής πρώτης εργασίας (E1) ζητήθηκε να κατασκευαστεί μια συνάρτηση, με όνομα `string_in` και δύο ορίσματα-συμβολοσειρές, η οποία ελέγχει αν η δεύτερη συμβολοσειρά εμφανίζεται μέσα στην πρώτη. Εάν εμφανίζεται, η συνάρτηση `string_in` επιστρέφει την θέση εντός της πρώτης συμβολοσειράς όπου ξεκινά η πρώτη εμφάνιση της δεύτερης συμβολοσειράς. Εάν δεν εμφανίζεται, η συνάρτηση επιστρέφει -1.

Για παράδειγμα, η κλήση:

```
string_in("This is not America", "is")
```

θα επιστρέψει 2, εφόσον το "is" περιέχεται στη λέξη "This" και ξεκινά από τη θέση 2 του πρώτου αλφαριθμητικού (αριθμούμε από το μηδέν). Αντίθετα, η κλήση

```
string_in("This is not America, this is not Earth", "Europe")
```

θα επιστρέψει -1.

Μπορείτε να βρείτε την περυσινή πρώτη εργασία, μαζί με την ενδεικτική της λύση, στο portal του μαθήματος: [study.eap.gr](http://study.eap.gr) -> Συμπληρωματικό υλικό -> Εργασίες Παρελθόντων ετών-> 2012-13.rar

Στην παρούσα εργασία σας ζητούνται τα εξής:

- Να τροποποιήσετε την `string_in`, έτσι ώστε αυτή να δέχεται τρία ορίσματα. Το τρίτο όρισμα θα είναι ένας μη-αρνητικός ακέραιος αριθμός, ο οποίος δεν θα πρέπει να ξεπερνά το μέγεθος της πρώτης συμβολοσειράς. Η αναζήτηση της δεύτερης συμβολοσειράς εντός της πρώτης θα γίνεται ξεκινώντας όχι από την αρχή της πρώτης, αλλά από τη θέση που υποδεικνύει το τρίτο όρισμα της συνάρτησης.

Για παράδειγμα, η κλήση:

```
string_in("This is not America", "is", 0)
```

θα επιστρέψει 2, αφού η αναζήτηση του "is" ξεκινά από τη θέση 0 της πρώτης συμβολοσειράς και αυτό συναντάται για πρώτη φορά στη θέση 2. Από την άλλη, η κλήση:

```
string_in("This is not America", "is", 4)
```

θα επιστρέψει 5, αφού η αναζήτηση του "is" ξεκινά από τη θέση 4 της πρώτης συμβολοσειράς και αυτό συναντάται για πρώτη φορά στη θέση 5. Τέλος, η κλήση:

```
string_in("This is not America", "is", 6)
```

θα επιστρέψει -1, αφού η αναζήτηση του "is" ξεκινά από τη θέση 6 της πρώτης συμβολοσειράς και αυτό δεν συναντάται από τη θέση 6 μέχρι το τέλος της συμβολοσειράς.



- Να γράψετε ένα πρόγραμμα C το οποίο θα δέχεται από το πληκτρολόγιο δύο συμβολοσειρές και θα μετρά πόσες φορές εμφανίζεται η δεύτερη συμβολοσειρά μέσα στην πρώτη, χρησιμοποιώντας την τροποποιημένη `string_in`.

Για να το πετύχει αυτό, το πρόγραμμά σας θα καλεί επαναληπτικά (μετρώντας το πλήθος των επαναλήψεων) την `string_in`, αρχικά ξεκινώντας την αναζήτηση από τη θέση 0 και στη συνέχεια από την επόμενη θέση κάθε τιμής που επιστρέφει η `string_in`, μέχρι η `string_in` να επιστρέψει -1. Στο τέλος θα εμφανίζει στην οθόνη το πλήθος των επαναλήψεων.

## Γνωστικά Αντικείμενα ανά θέμα

Γνωστικά αντικείμενα	Θ1	Θ2	Θ3	Θ4
Τύποι και μεταβλητές	X	X	X	X
Είσοδος πληκτρολογίου και έξοδος οθόνης	X	X	X	X
Μορφοποιημένη είσοδος και έξοδος			X	
Μονοδιάστατοι πίνακες		X		
Συναρτήσεις και πέρασμα παραμέτρων	X			X
Συναρτήσεις χειρισμού αλφαριθμητικών				X
Είσοδος και έξοδος σε αρχείο κειμένου			X	
Μαθηματικές συναρτήσεις	X			

ΚΡΙΤΗΡΙΑ ΑΞΙΟΛΟΓΗΣΗΣ	
Θέμα 1 Δυνάμεις ακεραίων – Άθροισμα δυνάμεων	25
Θέμα 2: Εκτύπωση Ιστογράμματος	25
Θέμα 3: Ανάγνωση από αρχείο – Μορφοποιημένη έξοδος στην οθόνη	15
Θέμα 4: Αναζήτηση συμβολοσειράς	25
Εικόνα εργασίας - σχολιασμός	10
<b>ΣΥΝΟΛΟ</b>	<b>100</b>
Ο συνολικός βαθμός θα διαιρεθεί δια 10, ώστε να προκύψει ο τελικός βαθμός της εργασίας.	

**Καλή Επιτυχία!!!**