

Τεχνολογίες και Μεθοδολογίες Προγραμματισμού Ι

2η Εργαστηριακή Άσκηση (Εαρινό εξάμηνο 2012)

Διδάσκοντες: Δημήτρης Λέκκας (dlek@aegean.gr),
Δημήτρης Ζήσης (dzissis@aegean.gr)
Ηλίας Ξυδιάς (xidias@aegean.gr)

Σημαντικές οδηγίες – Διαβάστε προσεκτικά:

Ο βαθμός της παρούσας εργασίας θα υπολογισθεί ως ποσοστό **15%** της τελικής βαθμολογίας του μαθήματος. Η εργασία μπορεί να αναληφθεί από μεμονωμένους φοιτητές ή από ομάδες 2 το πολύ φοιτητών. Η καταληκτική ημερομηνία παράδοσης της εργασίας σας είναι η **Κυριακή 27 Μαΐου 2012**. Φροντίστε να έχετε ολοκληρώσει την εργασία σας τουλάχιστον 2 ημέρες πριν από την προθεσμία, ώστε να μην προκύψουν προβλήματα της τελευταίας στιγμής. Παραδόσεις εργασιών μετά από αυτή την ημερομηνία δεν θα γίνουν δεκτές. Κατά τη κρίση των διδασκόντων κάποιοι φοιτητές είναι πιθανό να κληθούν σε συνέντευξη-παρουσίαση της εργασίας τους.

Για την άσκηση ζητούνται τα παρακάτω:

- Να περιγράψετε τον αλγόριθμο που ακολουθεί σε ψευδοκώδικα (στην Ελληνική γλώσσα) ή σε διάγραμμα ροής.
- Να υλοποιήσετε ένα πρόγραμμα σε γλώσσα C, που να επιλύει το πρόβλημα.
- Να παραδώσετε την άσκηση μέσω του συστήματος eclass στο σύνδεσμο «Εργασίες Φοιτητών» αφού έχετε συμπίεσει τα παραδοτέα αρχεία σε ένα αρχείο zip. Τα αρχεία που πρέπει να παραδοθούν είναι α) ένα Word αρχείο με τον ψευδοκώδικα ή το διάγραμμα ροής και τυχόν άλλες παρατηρήσεις, και β) ένα αρχείο με τον κώδικα C (αρχείο με κατάληξη .c).
- Είναι επιθυμητό να έχετε χρησιμοποιήσει τον compiler Bloodshed C++ (<http://www.bloodshed.net/devcpp.html>) και να έχετε δοκιμάσει την σωστή εκτέλεση του προγράμματος που έχετε γράψει.

Οι προϋποθέσεις για να βαθμολογηθεί η εργασία σας με άριστα είναι οι παρακάτω:

1. Να έχει σωστή λογική, να παράγει σωστά αποτελέσματα και να υλοποιεί αυτό που περιγράφει η εκφώνηση της άσκησης.
2. Ο κώδικας να είναι εύκολα αναγνώσιμος. Δηλαδή να χρησιμοποιούνται περιγραφικά ονόματα μεταβλητών, να υπάρχει οδόντωση (εσοχές στις ομάδες εντολών μέσα σε δομές ερωτήσεων και επαναλήψεων) και να υπάρχουν επαρκή σχόλια που να επεξηγούν τα βήματα του προγράμματος.
3. Το πρόγραμμα να έχει φιλική διεπαφή προς το χρήστη, δηλαδή ο χρήστης να καταλαβαίνει εύκολα τι πρέπει να κάνει.
4. Να περιλαμβάνει 'Αμυντικό Προγραμματισμό', δηλαδή να γίνεται έλεγχος ορθότητας για κάθε τιμή που ζητείται να εισάγει ο χρήστης και σε περιπτώσεις λάθους να εμφανίζονται ενημερωτικά μηνύματα.
5. Να έχει γίνει προσπάθεια βελτιστοποίησης του κώδικα, δηλαδή να μην περιέχει περιττά βήματα και μεταβλητές.
6. Να έχουν παραδοθεί έγκαιρα τα πλήρη και σωστά αρχεία.

Εκφώνηση Προβλήματος:

Έστω ότι θέλετε να ταξινομήσετε ένα πλήθος από γραπτά φοιτητών σε αύξουσα αλφαβητική σειρά με βάση το όνομα του φοιτητή/φοιτήτριας. Μία πιθανή προσέγγιση που μπορείτε να ακολουθήσετε είναι η εξής:

Βήμα 1 - Προεπεξεργασία:

Βήμα 1.1: Βάλε όλα τα γραπτά που αρχίζουν από 'A' σε μια ομάδα.

Βήμα 1.2: Βάλε όλα τα γραπτά που αρχίζουν από 'B' σε μια άλλη ομάδα.

Βήμα 1.3: Επανάλαβε την ίδια διαδικασία για τα υπόλοιπα γραπτά τοποθετώντας τα στις αντίστοιχες ομάδες.

Βήμα 2 - Ταξινόμηση: Ταξινόμησε την κάθε ομάδα ξεχωριστά. Χρησιμοποίησε οποιονδήποτε αλγόριθμο επιθυμείς.

Βήμα 3 - Συνένωση: Ξεκινώντας από την πρώτη ομάδα (και συνεχίζοντας με τις υπόλοιπες κατά σειρά), τοποθέτησε τα γραπτά σε μια νέα, τελική ομάδα. Η ομάδα αυτή θα περιέχει όλα τα γραπτά ταξινομημένα.

Παρά τη φαινομενική πολυπλοκότητα της τεχνικής αυτής, ενδέχεται ότι καταμερίζοντας τα δεδομένα κατά το στάδιο της προεπεξεργασίας θα έχετε ευκολότερο ή τουλάχιστον πιο διαχειρίσιμο έργο στη συνέχεια.

Χρησιμοποιήστε την πιο πάνω τεχνική ώστε να ταξινομήσετε ένα πλήθος N φυσικών αριθμών. (Για λόγους ευκολίας, θεωρήστε ότι οι αριθμοί αυτοί βρίσκονται στο διάστημα $[0, 150]$, δηλαδή από το 0 μέχρι και το 150.)

Να γραφούν διαδικασίες σε ψευδοκώδικα, σύμφωνα με τις ακόλουθες προδιαγραφές:

- Το πλήθος N των αριθμών δίνεται ως σταθερά. Θεωρήστε ότι όλοι οι αριθμοί παράγονται τυχαία από την συνάρτηση **rand()** που σας δίνεται στην συνέχεια και τοποθετούνται σε έναν μονοδιάστατο πίνακα με όνομα **DATA**.
- Χρησιμοποιήστε ακριβώς 15 ομάδες για το στάδιο της προεπεξεργασίας. Η πρώτη ομάδα θα περιέχει τους αριθμούς που ανήκουν στο διάστημα $[0, 9]$, η δεύτερη ομάδα θα περιέχει τους αριθμούς που ανήκουν στο διάστημα $[10, 19]$ κ.ο.κ. (**ΠΡΟΣΟΧΗ** όχι θέσεις στον πίνακα δηλ. $DATA[0], DATA[1], \dots, DATA[9]$ αλλά τιμές). Χρησιμοποιήστε έναν δισδιάστατο πίνακα με όνομα **GROUPS** για να αποθηκεύσετε αυτές τις ομάδες. Ο πίνακας αυτός θα έχει 15 γραμμές (μία για κάθε ομάδα). Το πλήθος των στηλών του πίνακα ανήκει στα ζητούμενα της υποεργασίας και θα πρέπει να σχολιάσετε την επιλογή σας. Η προεπεξεργασία των δεδομένων να αναπτυχθεί στο υποπρόγραμμα **FIRST()**.
- Στο υποπρόγραμμα **SORT()**, ταξινομήστε τα δεδομένα που έχουν ομαδοποιηθεί στο στάδιο της προεπεξεργασίας. Χρησιμοποιήστε τον αλγόριθμο bubble sort για την ταξινόμηση του συνόλου.

- Στο υποπρόγραμμα **CONCAT()**, συνενώστε τα ταξινομημένα δεδομένα εισάγοντάς τα στον αρχικό πίνακα.
- Τυπώστε τον ταξινομημένο πίνακα με το υποπρόγραμμα **PRINT()**.

Υπόδειξη 1: Η συνάρτηση που ακολουθεί επιστρέφει έναν τυχαίο ακέραιο από min έως και max χρησιμοποιώντας τη συνάρτηση rand() της C, η οποία βρίσκεται στη βιβλιοθήκη

<stdlib.h>:

```
int gen_rand(int min, int max)
{
    int n;
    n = ((rand() % (max - min + 1)) + min);
    return(n);
}
```

Υπόδειξη 2: Μπορείτε να χρησιμοποιήσετε *καθολικές* μεταβλητές για να αντιμετωπίσετε τα προβλήματα με την εμβέλεια των μεταβλητών.

Η λογοκλοπή και η αντιγραφή απαγορεύονται αυστηρά και θα τιμωρούνται παραδειγματικά. Σε περίπτωση αντιγραφής θα μηδενίζονται αυτόματα όλες οι εμπλεκόμενες εργασίες