

Παρατηρήσεις πάνω στο lab5

Έξοδος προγράμματος:

Προσέξτε τη μορφοποίηση της εξόδου του προγράμματός σας. Να ακολουθείτε πάντα τις οδηγίες για το πώς πρέπει να εμφανίζεται η έξοδος.

Έλεγχος εγκυρότητας επιθυμητού σφάλματος:

Το επιθυμητό σφάλμα πρέπει να είναι θετική ποσότητα μικρότερη του 1 και θέλουμε να επαναλάβουμε την εισαγωγή τιμής όσο αυτή **δεν είναι** εντός του επιθυμητού εύρους. Το επιθυμητό εύρος είναι `error > 0 && error < 1`, άρα η συνθήκη επανάληψης είναι άρνηση της παραπάνω έκφρασης: `error <= 0 || error >= 1`

Εφόσον το πλήθος επαναλήψεων δεν είναι γνωστό εκ των προτέρων θα πρέπει να χρησιμοποιήσουμε `while` ή `do-while`. Εφόσον θέλουμε να γίνει τουλάχιστον μία επανάληψη, η καλύτερη επιλογή είναι `do-while`.

Παρατήρηση 1: Ένα συχνό λάθος είναι η συνθήκη `error <= 0 && error >= 1`. Παρατηρήστε ότι δεν υπάρχει τιμή του `error` για την οποία αυτή η συνθήκη είναι αληθής. Μια καλή τακτική για να ελέγχετε την ορθότητα σύνθετης λογικής έκφρασης είναι να σκιαγραφείτε σε μια γραμμή τις τιμές για τις οποίες η έκφραση είναι αληθής. Αν δεν τέμνονται (αριστερό σχήμα) τότε δεν είναι ποτέ αληθής.



Παρατήρηση 2: Αποφεύγετε τη χρήση `flags` αν δεν είναι απαραίτητες. Το αριστερό loop στο παρακάτω σχήμα λειτουργεί όπως και το δεξί, αλλά η χρήση της `flag` το κάνει πιο δυσανάγνωστο. Επιπλέον, το όνομα `flag` δε δίνει καμία ένδειξη για τι ακριβώς σημαίνει η τιμή 0 ή 1 για αυτή τη μεταβλητή.

```
flag = 1;
do {
    /* do things */
    if (condition) flag = 0;
} while (flag == 1)
```



```
do {
    /* do things */
} while (!condition)
```



Υπολογισμός αθροίσματος:

Ο συμβατικός τρόπος υπολογισμού ενός αθροίσματος της μορφής $\Sigma = x + \text{term}_1 + \text{term}_2 + \text{term}_3 + \dots + \text{term}_N$ είναι:

Αν γνωρίζουμε το N εκ των προτέρων:

```
sum = x;
for (i = 1; i <= N; i++) {
    sum += termi;
}
```

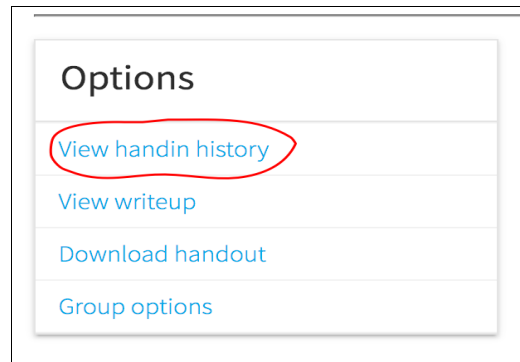
Αν δε γνωρίζουμε το N εκ των προτέρων:

```
sum = x;
i = 1;
while (κάποια συνθήκη) {
    sum += termi;
    i++;
}
```

Στις συγκεκριμένες ασκήσεις δε γνωρίζαμε το N , άρα ήταν καταλληλότερη η χρήση `while`. Εφόσον θέλαμε η επανάληψη να συνεχιστεί μέχρι να πλησιάσουμε ένα αποδεκτό σφάλμα, η συνθήκη έπρεπε να ήταν της μορφής `fabs(sum - target) > error`, δηλαδή συνεχίζουμε όσο η διαφορά μας από το στόχο είναι ακόμη μεγαλύτερη από το αποδεκτό σφάλμα.

Πώς να βλέπετε τα σχόλια βαθμολόγησης

Για να δείτε τα σχόλια βαθμολόγησης, κάντε login στο autolab, πηγαίνετε στο lab που σας ενδιαφέρει και επιλέξτε View Handin History:



Κάτω από κάθε κατηγορία βαθμολόγησης θα δείτε το βαθμό σας σε αυτή. Κάνετε κλικ πάνω στο βαθμό για να δείτε συγκεκριμένες παρατηρήσεις. Κάντε το ακόμη κι αν έχετε πάρει το μέγιστο δυνατό βαθμό.

Submission (0.0)	a_compilation (0.0)	a_tests (21.0)	a_results (24.0)	a_style (30.0)	b_compilation (0.0)	b_tests (16.0)	b_style (9.0)	PARATHRHSEIS (0.0)
0.0	-15.0	9.0	24.0	21.0	-5.0	16.0	9.0	0.0

Ακόμη κι αν έχετε μηδέν στις PARATHRHSEIS, κάντε click σε αυτό για να δείτε αν έχουμε προσθέσει επιπλέον σχόλια για τον κώδικά σας.

Αντιστοιχία βαθμού lab5 και πόντων εργαστηρίου

Βαθμός lab4	Πόντοι εργαστηρίου
0-19	0
20-44	1
45-100	3