

## 6η Σειρά Ασκήσεων

### Άσκηση 14.

Γράψτε ένα πρόγραμμα που θα ελέγχει αν δύο πίνακες  $N$  διαφορετικών ακεραίων αριθμών  $A$  και  $B$  έχουν όλα τα στοιχεία τους ίδια ή όχι. Συγκεκριμένα, το πρόγραμμά σας πρέπει να:

- Διαβάζει από την πρώτη γραμμή της εισόδου το πλήθος των στοιχείων  $N \leq 50,000$ .
- Διαβάζει από την δεύτερη γραμμή της εισόδου τα  $N$  στοιχεία του πίνακα  $A$  και από την τρίτη γραμμή της εισόδου τα  $N$  στοιχεία του πίνακα  $B$ . Τα στοιχεία των  $A$  και  $B$  είναι ακέραιοι αριθμοί που χωρίζονται με κενό και δίνονται σε τυχαία σειρά (δεν είναι ταξινομημένοι). Όλα τα στοιχεία του πίνακα  $A$  θα είναι διαφορετικά μεταξύ τους και όλα τα στοιχεία του πίνακα  $B$  είναι διαφορετικά μεταξύ τους.
- Ελέγχει αν οι πίνακες  $A$  και  $B$  έχουν όλα τα στοιχεία τους ίδια. Αν ναι, το πρόγραμμα τυπώνει τη λέξη “yes”. Αν όχι, το πρόγραμμα τυπώνει (στην ίδια γραμμή, χωρισμένα με κενό) τη λέξη “no”, το μικρότερο στοιχείο που υπάρχει στον ένα πίνακα και δεν υπάρχει στον άλλο, και το μεγαλύτερο στοιχείο που υπάρχει στον ένα πίνακα και δεν υπάρχει στον άλλο.

**Προσοχή:** Το πρόγραμμά σας πρέπει να εκτελείται σε λιγότερο από 1 sec. Αν δε χρησιμοποιήσετε αποδοτικό αλγόριθμο, είναι πιθανόν αυτό να μη συμβαίνει.

#### Παραδείγματα εισόδου:

```
7          | 8          | 9
33 48 11 26 8 1 42 | 81 92 44 18 2 55 70 26 | 1 2 3 4 5 6 7 8 10
1 26 42 8 48 11 33 | 28 18 55 92 44 70 2 80 | 2 3 4 5 6 7 8 9 10
```

#### Παραδείγματα εξόδου:

```
yes          | no 26 81          | no 1 9
```

► Να υποβληθεί στο αυτόματο σύστημα υποβολής και ελέγχου μέχρι την Παρασκευή 22/12/2017

### Άσκηση 15.

Για την άσκηση αυτή, ονομάζουμε λέξη μια ακολουθία χαρακτήρων που δεν περιέχει το κενό. Θεωρήστε ότι κάθε λέξη δεν υπερβαίνει τους 20 χαρακτήρες και ότι δεν μπορεί να κόβεται σε δύο γραμμές.

Γράψτε ένα πρόγραμμα το οποίο θα μορφοποιεί κείμενα. Το αρχικό κείμενο θα διαβάζεται από ένα αρχείο με όνομα “in.txt” και το τελικό (μορφοποιημένο) κείμενο θα γράφεται σε ένα αρχείο με όνομα “out.txt”. Η μορφοποίηση πρέπει να γίνεται προσθέτοντας ή αφαιρώντας κενά διαστήματα μεταξύ των λέξεων, με κατάλληλο τρόπο. Συγκεκριμένα, στο τελικό κείμενο:

- Το μήκος κάθε γραμμής (εκτός ίσως της τελευταίας) πρέπει να είναι 60 χαρακτήρες.
- Κάθε γραμμή (εκτός ίσως της τελευταίας) πρέπει να στοιχίζεται και στο αριστερό και στο δεξιό περιθώριο.
- Τα κενά μεταξύ των λέξεων πρέπει να ισοκατανέμονται (κατά το δυνατόν). Αν ονομάσουμε διάκενο το πλήθος των κενών διαστημάτων μεταξύ δύο διαδοχικών λέξεων, τότε πρέπει σε κάθε γραμμή:
  - το μεγαλύτερο διάκενο να διαφέρει από το μικρότερο το πολύ κατά ένα κενό διάστημα, και
  - αν ένα διάκενο είναι μεγαλύτερο από κάποιο άλλο, τότε πρέπει να βρίσκεται δεξιότερα.

Για την υποβολή στο αυτόματο σύστημα υποβολής και ελέγχου, θα χρειαστεί να τροποποιήσετε λίγο το πρόγραμμά σας ώστε να διαβάζει το αρχικό κείμενο από το πληκτρολόγιο και να τυπώνει το τελικό (μορφοποιημένο) κείμενο στην οθόνη.

### Παράδειγμα εισόδου 1:

Once upon a time in China, some believe, around the year one double-ought three, head priest of the White Lotus Clan, Pai Mei was walking down the road, contemplating whatever it is that a man of Pai Mei's infinite power contemplates - which is another way of saying "who knows" - when a Shaolin monk appeared, traveling in the opposite direction. As the monk and the priest crossed paths, Pai Mei, in a practically unfathomable display of generosity, gave the monk the slightest of nods. The nod was not returned.

### Παράδειγμα εξόδου 1:

Once upon a time in China, some believe, around the year one double-ought three, head priest of the White Lotus Clan, Pai Mei was walking down the road, contemplating whatever it is that a man of Pai Mei's infinite power contemplates - which is another way of saying "who knows" - when a Shaolin monk appeared, traveling in the opposite direction. As the monk and the priest crossed paths, Pai Mei, in a practically unfathomable display of generosity, gave the monk the slightest of nods. The nod was not returned.

### Παράδειγμα εισόδου 2:

The first electronic computers were monstrous contraptions, filling several rooms, consuming as much electricity as a good-size factory, and costing millions of 1940s dollars (but with the computing power of a modern hand-held calculator). The programmers who used these machines believed that the computer's time was more valuable than theirs.

They programmed in machine language.

Machine language is the sequence of bits that directly controls a processor, causing it to add, compare, move data from one place to another, and so forth at appropriate times.

Specifying programs at this level of detail is an enormously tedious task.

The following program calculates the greatest common divisor (GCD) of two integers, using Euclid's algorithm.

It is written in machine language, expressed here as hexadecimal (base 16) numbers, for the MIPS R4000 processor.

### Παράδειγμα εξόδου 2:

The first electronic computers were monstrous contraptions, filling several rooms, consuming as much electricity as a good-size factory, and costing millions of 1940s dollars (but with the computing power of a modern hand-held calculator). The programmers who used these machines believed that the computer's time was more valuable than theirs. They programmed in machine language. Machine language is the sequence of bits that directly controls a processor, causing it to add, compare, move data from one place to another, and so forth at appropriate times. Specifying programs at this level of detail is an enormously tedious task. The following program calculates the greatest common divisor (GCD) of two integers, using Euclid's algorithm. It is written in machine language, expressed here as hexadecimal (base 16) numbers, for the MIPS R4000 processor.

► Να υποβληθεί στο αυτόματο σύστημα υποβολής και ελέγχου μέχρι την Παρασκευή 22/12/2017