

ΠΡΟΧΩΡΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι:

ΕΙΣΑΓΩΓΗ ΣΤΗ C++

Θέματα Εξετάσεων Ιανουαρίου 2018

- [2.5] 1. Βρείτε με πρόγραμμα το μοναδικό θετικό ακέραιο που το τετράγωνό του είναι δεκαπήφιος αριθμός της μορφής $1_2_3_4_5_$. Το $_$ συμβολίζει απλό ψηφίο (πιθανώς διαφορετικό σε κάθε θέση).

[Απάντηση: 34934].

- [3.5] 2. Ένα μυρμήγκι βρίσκεται σε ορθογώνιο πλέγμα από 128×128 τετράγωνα. Τα τετράγωνα μπορούν να είναι είτε άσπρα είτε μαύρα. Αρχικά είναι όλα άσπρα. Το μυρμήγκι έχει αρχική θέση το κέντρο του πλέγματος (το σημείο $(64, 64)$), κατεύθυνση προς τα επάνω και κινείται σε κάθε βήμα του σύμφωνα με τους ακόλουθους κανόνες:

- Αν βρίσκεται σε μαύρο τετράγωνο, αλλάζει το χρώμα του τετραγώνου σε άσπρο, στρέφει κατά 90° αριστερόστροφα και προχωρά κατά ένα τετράγωνο.
- Αν βρίσκεται σε άσπρο τετράγωνο, αλλάζει το χρώμα του τετραγώνου σε μαύρο, στρέφει κατά 90° δεξιόστροφα και προχωρά κατά ένα τετράγωνο.

Γράψτε πρόγραμμα που να προσομοιώνει την κίνηση του μυρμηγκιού για 12000 βήματα. Κάθε 100 βήματα να αποθηκεύετε την εικόνα του πλέγματος σε αρχείο με τη διαμόρφωση `plain.pbm` (κάθε τετράγωνο θα είναι ένα pixel).

Η διαμόρφωση `pbm` για αποθήκευση ασπρόμαυρης εικόνας σε αρχείο είναι η ακόλουθη:

- Η πρώτη γραμμή του αρχείου πρέπει να γράφει: `P1`.
- Η δεύτερη να γράφει τις διαστάσεις της εικόνας: πλάτος ύψος (δηλαδή τους δύο αριθμούς με κενό μεταξύ τους).
- Να ακολουθούν οι αριθμοί 0 ή 1· αυτοί αντιπροσωπεύουν τα pixels της εικόνας *κατά γραμμές*: κάθε λευκό pixel αντιστοιχεί στο 0 και κάθε μαύρο στο 1. Οι αριθμοί μπορούν να διαχωρίζονται από κενά αλλά δεν είναι απαραίτητο. Σε κάθε σειρά του αρχείου μπορούμε να έχουμε έως 70 χαρακτήρες.

[4.0]

3. Γράψτε συνάρτηση που να δέχεται ως ορίσματα

- ένα ζεύγος από iterators εισόδου, `beg1`, `end1`, που ορίζουν μια ταξινομημένη ακολουθία στοιχείων.
- ένα δεύτερο ζεύγος από iterators εισόδου, `beg2`, `end2`, πιθανώς διαφορετικού τύπου από τα `beg1`, `end1`. Τα `beg2`, `end2` ορίζουν δεύτερη ταξινομημένη ακολουθία στοιχείων.
- ένα iterator εξόδου, `beg3`.

Η συνάρτηση θα αντιγράφει σε θέσεις από το `beg3` και μετά, τα κοινά στοιχεία των διαστημάτων `[beg1,end1)` και `[beg2,end2)`, διατηρώντας τη σειρά τους. Αν κάποιο στοιχείο εμφανίζεται m_1 φορές στο πρώτο διάστημα και m_2 στο δεύτερο, να αντιγράφονται τα πρώτα $\min\{m_1,m_2\}$ από αυτά. Θα επιστρέφει iterator στην ακολουθία εξόδου, στην επόμενη θέση από την τελευταία που έχει γραφεί. Θεωρούμε ότι υπάρχουν όσες θέσεις χρειάζονται μετά το `beg3`.

Συμπληρώστε επομένως τον κώδικα

```
template<typename InputIterator1, typename InputIterator2,
        typename OutputIterator>
OutputIterator
f(InputIterator1 beg1, InputIterator1 end1,
  InputIterator2 beg2, InputIterator2 end2,
  OutputIterator beg3)
{
    ...
}
```

Να θεωρήσετε ότι δύο στοιχεία είναι «ίσα» όταν κανένα δεν είναι μικρότερο από το άλλο. Δηλαδή, μη χρησιμοποιήσετε τον τελεστή `'=='` για τη σύγκριση δύο στοιχείων `a,b`· η ισότητα τους να ελέγχεται με τον κώδικα `!(a<b) && !(b<a)`.

Γράψτε πρόγραμμα στο οποίο θα δημιουργείτε δύο containers της επιλογής σας που θα αποθηκεύουν τις ακέραιες τιμές `{5, 5, 5, 6, 7, 8, 1, 2, 3, 4}` και `{11, 5, 9, 7, 5, 3, 3, 3}` αντίστοιχα. Χρησιμοποιήστε τη συγκεκριμένη συνάρτηση για να βρείτε και να τυπώσετε στην οθόνη τα κοινά στοιχεία τους.

Να στείλετε τους κώδικες που θα γράψετε, ως συνημμένους σε email στο ety215@materials.uoc.gr.

Διάρκεια: 3 ώρες

Καλή επιτυχία!