

31^{ος} ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΜΑΤΑ ΤΕΛΙΚΗΣ ΦΑΣΗΣ

Θέμα 1^ο: Συμμετρικές συμβολοσειρές

[30 Μονάδες]

Μία μη κενή συμβολοσειρά S θα λέμε ότι είναι *συμμετρική* όταν είναι η συνένωση (παράθεση) μίας άλλης συμβολοσειράς W με τον εαυτό της, δηλαδή όταν $S = WW$. Θα λέμε επίσης ότι η συμβολοσειρά W είναι το *μισό* της S . Για παράδειγμα, η συμβολοσειρά "hellohello" είναι συμμετρική (και το μισό της είναι η συμβολοσειρά "hello") ενώ αντίθετα οι συμβολοσειρές "abca" και "abcab" δεν είναι συμμετρικές. Προφανώς, τα μήκη των συμμετρικών συμβολοσειρών είναι άρτιοι αριθμοί (διπλάσιοι του μήκους των αντίστοιχων μισών).

Έστω μία συμβολοσειρά S της οποίας το μήκος είναι περιττός αριθμός. Ζητείται να βρεθεί αν από την S μπορεί να αφαιρεθεί ένας ακριβώς χαρακτήρας έτσι ώστε η συμβολοσειρά που θα προκύψει να είναι συμμετρική. Επιπλέον, ζητείται να βρεθεί αν αυτό μπορεί να γίνει με μοναδικό τρόπο ή αν με διαφορετικές αφαιρέσεις μπορούν να προκύψουν περισσότερες διαφορετικές συμμετρικές συμβολοσειρές.

Πρόβλημα:

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του IOI (PASCAL, C, C++, Java) το οποίο θα διαβάζει μερικές τέτοιες συμβολοσειρές και για κάθε μία θα κάνει ένα από τα ακόλουθα:

- Αν με αφαίρεση ενός χαρακτήρα προκύπτει μία μοναδική συμμετρική συμβολοσειρά, τότε θα εκτυπώνει το μισό αυτής της συμβολοσειράς.
- Αν με αφαίρεση ενός χαρακτήρα μπορούν να προκύψουν περισσότερες διαφορετικές συμμετρικές συμβολοσειρές, τότε θα εκτυπώνει τον κωδικό αριθμό 1 (ένα).
- Αν με την αφαίρεση κανενός χαρακτήρα δεν προκύπτει συμμετρική συμβολοσειρά, τότε θα εκτυπώνει τον κωδικό αριθμό 0 (μηδέν).

Αρχεία Εισόδου:

Το αρχείο εισόδου με όνομα **symstr.in** είναι αρχείο κειμένου με την εξής δομή. Η πρώτη γραμμή έχει δύο ακέραιους αριθμούς **T** και **N**, χωρισμένους με ένα κενό διάστημα. Ο αριθμός **T** θα παριστάνει το πλήθος των συμβολοσειρών που θα ακολουθήσουν, ενώ ο αριθμός **N** θα είναι περιττός και θα παριστάνει το μήκος κάθε συμβολοσειράς. Κάθε μία από τις επόμενες **T** γραμμές θα περιέχει μία συμβολοσειρά αποτελούμενη από **N** κεφαλαία γράμματα του λατινικού αλφαβήτου.



Αρχεία Εξόδου:

Το αρχείο εξόδου με όνομα **symstr.out** είναι αρχείο κειμένου με την εξής δομή. Έχει ακριβώς **T** γραμμές που κάθε μία περιέχει την απάντηση στην αντίστοιχη συμβολοσειρά της εισόδου (βλ. παραπάνω).

Παράδειγμα Αρχείων Εισόδου - Εξόδου:

symstr.in	symstr.out
4 9 ABCKDABCD ABCDEFGHI ABABABABA AAAAAAAAA	ABCD 0 1 AAAA

Εξήγηση: Στο παράδειγμα υπάρχουν 4 συμβολοσειρές που κάθε μία έχει 9 γράμματα. Για κάθε συμβολοσειρά, η απάντηση εξηγείται παρακάτω:

- Από τη συμβολοσειρά "ABCKDABCD" αν αφαιρεθεί το γράμμα "K" τότε προκύπτει η συμμετρική συμβολοσειρά "ABCDABCD" που το μισό της είναι "ABCD" — αυτή είναι η μοναδική συμμετρική συμβολοσειρά που μπορεί να προκύψει από την αφαίρεση ενός χαρακτήρα.
- Από τη συμβολοσειρά "ABCDEFGHI" όποιο γράμμα και αν αφαιρεθεί δεν προκύπτει συμμετρική συμβολοσειρά, άρα εκτυπώνεται "0" (μηδέν).
- Από τη συμβολοσειρά "ABABABABA" αν αφαιρεθεί το πρώτο "A" τότε προκύπτει η συμμετρική συμβολοσειρά "BABABABA", ενώ αν αφαιρεθεί το τελευταίο "A" τότε προκύπτει η συμμετρική συμβολοσειρά "ABABABAB". Άρα με αφαίρεση ενός χαρακτήρα δεν προκύπτει μοναδική συμμετρική συμβολοσειρά και γι' αυτό εκτυπώνεται "1" (ένα).
- Από τη συμβολοσειρά "AAAAAAAAA" όποιο γράμμα και αν αφαιρεθεί προκύπτει η συμμετρική συμβολοσειρά "AAAAAAAA", της οποίας το μισό είναι "AAAA".

Περιορισμοί:

- Για περιπτώσεις ελέγχου συνολικής αξίας 50%, θα είναι:
 $1 \leq T \leq 10$, $3 \leq N \leq 10.000$ και N περιττός αριθμός
- Για περιπτώσεις ελέγχου συνολικής αξίας 70%, δε θα χρειαστεί να εκτυπώσετε τον κωδικό "1" (ένα) — δε χρειάζεται έλεγχος μοναδικότητας.
- Για περιπτώσεις ελέγχου συνολικής αξίας 100%, θα είναι:
 $1 \leq T \leq 10$, $3 \leq N \leq 1.000.000$ και N περιττός αριθμός

Προσοχή! Φροντίστε να διαβάζετε την είσοδο και να εκτυπώνετε την έξοδο αποδοτικά, ειδικά αν προγραμματίζετε σε C++ ή Java.

Μορφοποίηση: Στην έξοδο, όλες οι γραμμές τερματίζουν με ένα χαρακτήρα newline.

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Μέγιστη διαθέσιμη μνήμη: 64 MB.

Θέμα 2^ο: Τι πληρώνει το λαχείο**[35 Μονάδες]**

Σε μία λαχειοφόρο αγορά, έστω ότι έχουν διατεθεί N λαχνοί, καθένας από τους οποίους είναι ένας φυσικός αριθμός X_i αποτελούμενος από ακριβώς K δεκαδικά ψηφία (οι λαχνοί επιτρέπεται να ξεκινούν και με το ψηφίο 0). Κληρώνεται ένας τυχερός αριθμός Y . Βάσει αυτού του αριθμού, κάθε λαχνός κερδίζει ένα χρηματικό ποσό ως εξής: αν τα M τελευταία ψηφία του λαχνού X_i είναι τα ίδια με τα M τελευταία (λήγοντα) ψηφία του τυχερού αριθμού Y , τότε ο κάτοχος του λαχνού κερδίζει $2^M - 1$ ευρώ.

Για παράδειγμα, αν $K = 6$, $X_i = 391742$ και $Y = 421742$, τότε ο λαχνός X_i έχει $M = 4$ κοινά λήγοντα ψηφία με τον τυχερό αριθμό Y (δηλαδή και οι δύο λήγουν σε 1742) και ο κάτοχός του κερδίζει $2^4 - 1 = 15$ ευρώ.

Δίνονται οι λαχνοί που έχουν διατεθεί και ο τυχερός αριθμός. Ζητούνται πόσοι λαχνοί κερδίζουν κάποιο μη μηδενικό ποσό (δηλαδή πόσοι έχουν ένα ή περισσότερα κοινά λήγοντα ψηφία με τον τυχερό αριθμό) και ποιο είναι το συνολικό ποσό που κερδίζουν όλοι οι λαχνοί. Επειδή το συνολικό ποσό μπορεί να είναι πολύ μεγάλος αριθμός, ζητείται να υπολογίσετε το υπόλοιπο της ακέραιας διαίρεσης (modulo) αυτού με τον αριθμό $10^9 + 7$.

Πρόβλημα

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες της IOI (Pascal, C, C++, Java) το οποίο θα διαβάζει τους λαχνούς που έχουν διατεθεί και μια σειρά εναλλακτικών τυχερών αριθμών και για κάθε εναλλακτικό τυχερό αριθμό θα εκτυπώνει πόσοι λαχνοί κερδίζουν και ποιο ποσό συνολικά.

Αρχεία εισόδου:

Το αρχείο εισόδου με όνομα **lottery.in** είναι αρχείο κειμένου με την εξής δομή. Η πρώτη γραμμή του περιέχει τρεις ακέραιους αριθμούς **K**, **N**, και **Q**, χωρισμένους ανά δύο με ένα κενό διάστημα: **K** είναι το πλήθος των δεκαδικών ψηφίων των λαχνών, **N** είναι το πλήθος των λαχνών που έχουν διατεθεί και **Q** το πλήθος των εναλλακτικών τυχερών αριθμών. Κάθε μία από τις επόμενες **N** γραμμές θα περιέχει έναν ακέραιο αριθμό **X_i** με ακριβώς **K** ψηφία, που παριστάνει έναν λαχνό που έχει διατεθεί. Κάθε μία από τις επόμενες **Q** γραμμές θα περιέχει έναν ακέραιο αριθμό **Y_i** με ακριβώς **K** ψηφία, που παριστάνει έναν εναλλακτικό τυχερό αριθμό.

Αρχεία εξόδου:

Το αρχείο εξόδου με όνομα **lottery.out** είναι αρχείο κειμένου με την εξής δομή. Έχει ακριβώς **Q** γραμμές. Κάθε μία από αυτές αντιστοιχεί κατά σειρά σε έναν εναλλακτικό τυχερό αριθμό **Y_i** και περιέχει δύο ακέραιους αριθμούς **C_i** και **S_i** χωρισμένους με ένα κενό διάστημα: **C_i** είναι το πλήθος των λαχνών που κερδίζουν και **S_i** το υπόλοιπο της ακέραιας διαίρεσης (modulo) του συνολικού ποσού που κερδίζουν όλοι οι λαχνοί με τον αριθμό $10^9 + 7$.

Σελίδα 3 από 8



Παραδείγματα αρχείων εισόδου – εξόδου

1^ο

lottery.in	lottery.out
6 1 1 391742 421742	1 15

Εξήγηση: Υπάρχει ένας λαχνός (391742) και ένας εναλλακτικός τυχερός αριθμός (421742). Το παράδειγμα αυτό εξηγείται παραπάνω στην εκφώνηση.

2^ο

lottery.in	lottery.out
4 6 3 3581 2619 4904 8859 3919 4964 4904 8933 3419	2 16 0 0 3 7

Εξήγηση: Υπάρχουν έξι λαχνοί (3581, 2619, 4904, 8859, 3919, 4964) και τρεις εναλλακτικοί τυχεροί αριθμοί (4904, 8933, 3419). Αν κληρωθεί ο πρώτος (4904) τότε θα κερδίσει 15 ευρώ ο λαχνός 4904 (που συμπίπτει με τον τυχερό αριθμό) και 1 ευρώ ο 4964. Αν κληρωθεί ο δεύτερος (8933) τότε δεν κερδίζει κανένας λαχνός. Αν κληρωθεί ο τρίτος (3419) τότε οι λαχνοί 2619 και 3919 κερδίζουν από 3 ευρώ καθένας και ο 8859 κερδίζει 1 ευρώ.

Περιορισμοί:

- Για περιπτώσεις ελέγχου συνολικής αξίας 20%, θα είναι:
 $1 \leq K \leq 9, 1 \leq N \leq 1000, 1 \leq Q \leq 1000, 0 \leq S_i \leq 10^9$
- Για περιπτώσεις ελέγχου συνολικής αξίας 40%, θα είναι:
 $1 \leq K \leq 30, 1 \leq N \leq 1000, 1 \leq Q \leq 1000, 0 \leq S_i \leq 10^9$
- Για περιπτώσεις ελέγχου συνολικής αξίας 100%, θα είναι:
 $1 \leq K \leq 100, 1 \leq N \leq 1.000.000, 1 \leq Q \leq 1.000.000$
 $(N+Q) \cdot K \leq 1.000.000$

Προσοχή! Φροντίστε να διαβάζετε την είσοδο και να εκτυπώνετε την έξοδο αποδοτικά, ειδικά αν προγραμματίζετε σε C++ ή Java.

Μορφοποίηση: Στην έξοδο, όλες οι γραμμές τερματίζουν με ένα χαρακτήρα newline.

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Μέγιστη διαθέσιμη μνήμη: 128 MB.

Θέμα 3^ο: Ελάχιστη απόσταση

[35 Μονάδες]

Δίνεται ένας κατευθυνόμενος γράφος με βάρη στις ακμές, τα οποία είναι (μεγάλες) δυνάμεις του 2. Ζητείται να υπολογιστεί η ελάχιστη απόσταση από την κορυφή S στην κορυφή T . Επειδή η απόσταση αυτή μπορεί να είναι πολύ μεγάλος αριθμός, ζητείται να υπολογίσετε το υπόλοιπο της ακέραιας διαίρεσης (modulo) της απόστασης με τον αριθμό 10^9+7 .

Πρόβλημα

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες της IOI (Pascal, C, C++, Java) το οποίο θα διαβάζει τις ακμές του γράφου και τις κορυφές S και T και θα βρίσκει το υπόλοιπο της ακέραιας διαίρεσης (modulo) της ελάχιστης απόστασης από την S στην T με τον αριθμό 10^9+7 .

Αρχεία εισόδου:

Το αρχείο εισόδου με όνομα **shortpowtwo.in** είναι αρχείο κειμένου. Η πρώτη γραμμή περιέχει δύο ακεραίους N και M χωρισμένους μεταξύ τους με ένα κενό διάστημα: το πλήθος των κορυφών και το πλήθος των ακμών του γράφου, αντίστοιχα. Κάθε μία από τις επόμενες M γραμμές περιγράφει μία ακμή: περιέχει τρεις ακεραίους αριθμούς u , v και w χωρισμένους ανά δύο με ένα κενό διάστημα, οι οποίοι σημαίνουν ότι υπάρχει μία ακμή από την κορυφή με αριθμό u προς την κορυφή με αριθμό v , με βάρος 2^w . Οι κορυφές είναι αριθμημένες από 1 μέχρι και N . Η επόμενη και τελευταία γραμμή περιέχει δύο ακεραίους αριθμούς S και T , χωρισμένους με ένα κενό διάστημα: τους αριθμούς των κορυφών αφετηρίας και προορισμού.

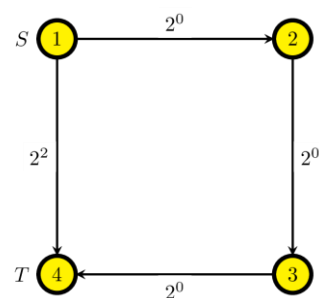
Αρχεία εξόδου:

Το αρχείο εξόδου με όνομα **shortpowtwo.out** είναι αρχείο κειμένου αποτελούμενο από μία μόνο γραμμή που περιέχει ακριβώς έναν ακεραίο αριθμό: το υπόλοιπο της ακέραιας διαίρεσης (modulo) της ελάχιστης απόστασης από την κορυφή S στην κορυφή T με τον αριθμό 10^9+7 . Αν δεν υπάρχει μονοπάτι από την κορυφή S στην κορυφή T , η γραμμή αυτή πρέπει να περιέχει τον αριθμό -1 .

Παραδείγματα αρχείων εισόδου – εξόδου

1^ο

shortpowtwo.in	shortpowtwo.out
4 4 1 4 2 1 2 0 2 3 0 3 4 0 1 4	3



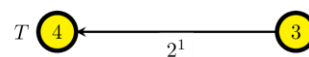
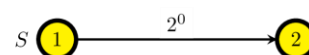
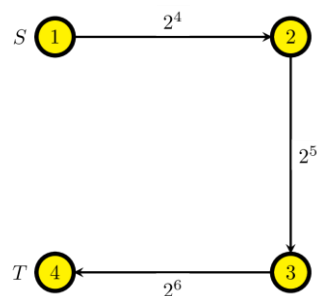
Σελίδα 5 από 8

2°

shortpowtwo.in	shortpowtwo.out
4 3 1 2 4 2 3 5 3 4 6 1 4	112

3°

shortpowtwo.in	shortpowtwo.out
4 2 1 2 0 3 4 1 1 4	-1



Περιορισμοί:

- Για περιπτώσεις ελέγχου συνολικής αξίας 20%, θα είναι:
 $1 \leq N \leq 1000, 0 \leq M \leq 1000, 0 \leq w \leq 10$
- Για περιπτώσεις ελέγχου συνολικής αξίας 40%, θα είναι:
 $1 \leq N \leq 100.000, 0 \leq M \leq 100.000, 0 \leq w \leq 10$
- Για περιπτώσεις ελέγχου συνολικής αξίας 65%, θα είναι:
 $1 \leq N \leq 100.000, 0 \leq M \leq 100.000, 0 \leq w \leq 500$
- Για περιπτώσεις ελέγχου συνολικής αξίας 100%, θα είναι:
 $1 \leq N \leq 100.000, 0 \leq M \leq 100.000, 0 \leq w \leq 100.000$

Προσοχή! Φροντίστε να διαβάζετε την είσοδο και να εκτυπώνετε την έξοδο αποδοτικά, ειδικά αν προγραμματίζετε σε C++ ή Java.

Μορφοποίηση: Στην έξοδο, όλες οι γραμμές τερματίζουν με ένα χαρακτήρα newline.

Μέγιστος χρόνος εκτέλεσης: 2 sec.

Μέγιστη διαθέσιμη μνήμη: 256 MB.



ΝΑ ΑΠΑΝΤΗΣΕΤΕ ΚΑΙ ΣΤΑ ΤΡΙΑ ΘΕΜΑΤΑ

ΔΙΑΡΚΕΙΑ ΕΞΕΤΑΣΗΣ ΤΕΣΣΕΡΙΣ (4) ΩΡΕΣ

ΚΑΛΗ ΣΑΣ ΕΠΙΤΥΧΙΑ

Ακολουθούν χρήσιμες οδηγίες !

Διαβάστε τις ακόλουθες παρατηρήσεις προσεκτικά!

- ✓ Ερωτήσεις που αφορούν τις παρατηρήσεις αυτές δεν θα απαντηθούν. Η πιστή τήρηση των αναφερόμενων οδηγιών είναι απαραίτητη.
- ✓ Οι αναφερόμενοι σε κάθε θέμα χρόνοι είναι ενδεικτικοί. Η επιτροπή μπορεί να τους αυξομειώσει προκειμένου να επιτύχει καλύτερη κλιμάκωση της βαθμολογίας.

1. Στην αρχή του πηγαίου κώδικά σας, θα πρέπει να χρησιμοποιήσετε τις επικεφαλίδες, ανάλογα με το πρόβλημα πχ.:

```
/*
USER: username
LANG: C
TASK: symstr
*/                                για κώδικα σε C

/*
USER: username
LANG: C++
TASK: symstr
*/                                για κώδικα σε C++

(*
USER: username
LANG: PASCAL
TASK: lottery
*)                                για κώδικα σε PASCAL

/*
USER: username
LANG: Java
TASK: shortpowtwo
*/                                για κώδικα σε Java
```

2. Έλεγχος τιμών δεν απαιτείται. Οι τιμές των αρχείων ελέγχου είναι πάντα έγκυρες.
3. Το σύστημα αξιολόγησης «τρέχει» σε **Linux**. Σας προτείνουμε να δοκιμάζετε τις λύσεις σας στο σύστημα. Έχετε δικαίωμα πολλαπλών υποβολών μέχρι το τέλος του

Σελίδα 7 από 8



διαγωνισμού. Μετά από κάθε υποβολή θα λαμβάνετε την αξιολόγηση της λύσης σας, σε τμήμα των Αρχείων Ελέγχου.

4. Οι επιλογές του μεταγλωττιστή που χρησιμοποιούνται για τη βαθμολόγηση είναι οι εξής:
 - C: `gcc -std=c99 -O2 -DCONTEST -s -static -lm`
 - C++: `g++ -std=c++11 -O2 -DCONTEST -s -static -lm`
 - Free Pascal: `fpc -O2 -dCONTEST -XS`
 - Java: `javac`
 5. Το Linux ξεχωρίζει μεταξύ κεφαλαίων και πεζών γραμμάτων. Ελέγξτε ότι τα ονόματα των αρχείων εισόδου και εξόδου είναι γραμμένα με μικρά (πεζά) γράμματα.
 6. Τα προγράμματά σας πρέπει να επιστρέφουν ως κωδικό εξόδου το μηδέν.
 7. Για προγραμματισμό σε C και C++ η συνάρτηση `main()` πρέπει πάντα να τερματίζει με τις εντολές `"return(0);"` ή `"exit(0);"`.
 8. Οι προγραμματιστές σε Pascal πρέπει να χρησιμοποιούν την εντολή `"halt"` μόνο με κωδικό εξόδου το μηδέν (μόνο δηλαδή με την μορφή `"halt;"` ή `"halt(0);"`).
 9. Το πρόγραμμα αξιολόγησης θα εξετάσει την τιμή που επιστρέφει το πρόγραμμά σας. Εάν η τιμή αυτή δεν είναι μηδέν, τότε το πρόγραμμα δεν θα βαθμολογηθεί για το συγκεκριμένο test.
 10. Κανένας άλλος χαρακτήρας εκτός του χαρακτήρα νέας γραμμής (newline) (χαρακτήρας 0A στο ASCII εκφρασμένο στο δεκαεξαδικό σύστημα αρίθμησης, \n για προγραμματιστές C, C++ ή Java, \$0A για προγραμματιστές Pascal) δεν θα υπάρχει μετά τον τελευταίο αριθμό κάθε γραμμής των αρχείων εισόδου και εξόδου. Δηλαδή, κάθε γραμμή των αρχείων εισόδου και εξόδου, συμπεριλαμβανομένης και της τελευταίας, τερματίζεται με τον χαρακτήρα νέας γραμμής όπως ορίστηκε παραπάνω.
 11. Για προγραμματισμό σε Java, το αρχείο πηγαίου κώδικα πρέπει να περιέχει μόνο μία `public class`, με όνομα το ίδιο με το κωδικό όνομα του προβλήματος (με πεζά γράμματα, π.χ. `lottery`). Το αρχείο πρέπει να ονομάζεται με το ίδιο όνομα της κλάσης και κατάληξη `.java`. Αν χρειαστεί να ορίσετε επιπλέον κλάσεις, θα πρέπει να φροντίσετε να μην είναι `public`.
- ✓ Κάθε απόπειρα κακόβουλης εισόδου ή ακόμα και εξερεύνησης του συστήματος, εκτός της παρεχόμενης διεπαφής, θα εντοπίζεται και θα επιβάλλονται κυρώσεις.

Με τη συνεργασία:

Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης, Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών, Εθνικού Μετσόβιου Πολυτεχνείου, Πανεπιστημίου Αιγαίου, Πανεπιστημίου Ιωαννίνων, Πανεπιστημίου Πατρών, Πανεπιστημίου Πειραιώς, ΤΕΙ Αθήνας.

