



22^{ος} ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΜΑΤΑ ΤΕΛΙΚΗΣ ΦΑΣΗΣ

Θέμα 1ο: Lines man

[25 Μονάδες]

Το ποδόσφαιρο από την ανακάλυψή του στο Πανεπιστήμιο του Cambridge, έγινε το πιο δημοφιλές αλλά και το πιο εύκολα παιζόμενο άθλημα. Ένας σχετικά επίπεδος τόπος και μια μπάλα αρκούν. Σύμφωνα με τους κανόνες του, ο διαιτητής κινείται μέσα στο γήπεδο και οι επόπτες γραμμών κατά μήκος των πλευρικών γραμμών, στο μισό γηπέδου έκαστος. Αν το γήπεδο ποδοσφαίρου έχει μήκος A , οι δύο επόπτες γραμμών ξεκινούν από το κέντρο του γηπέδου ($A/2$). Μέσα στο γήπεδο η μπάλα κινείται σε διάφορα σημεία. Οι επόπτες πρέπει να παρακολουθούν τις φάσεις, κινούμενοι μόνο κατά μήκος των πλευρικών γραμμών από το 0 έως $A/2$ και από $A/2$ έως A αντίστοιχα.

Πρόβλημα:

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του IOI το οποίο: Αφού διαβάσει το μήκος του γηπέδου, τον αριθμό των φάσεων και τη θέση (μήκος) που συντελείται κάθε φάση, θα υπολογίζει τα μέτρα που διάνυσαν οι επόπτες γραμμών κατά τη διάρκεια ενός αγώνα.

[Παρατήρηση: Οι επόπτες ξεκινούν από το κέντρο αλλά δεν επιστρέφουν υποχρεωτικά σε αυτό στο τέλος του παιχνιδιού.]

Αρχεία Εισόδου:

Τα αρχεία εισόδου με όνομα **lines_man.in** είναι αρχεία κειμένου με την εξής δομή: Στην πρώτη γραμμή έχουν έναν άρτιο ακέραιο αριθμό A ($10 \leq A \leq 1.000$): το μήκος του γηπέδου. Στη δεύτερη γραμμή έχουν έναν ακέραιο αριθμό M ($10 \leq M \leq 100.000$): το πλήθος των φάσεων. Οι επόμενες M γραμμές περιέχουν από ένα ακέραιο αριθμό Y ($0 \leq Y \leq A$): το μήκος του γηπέδου επί του οποίου εξελίσσεται η αντίστοιχη φάση.

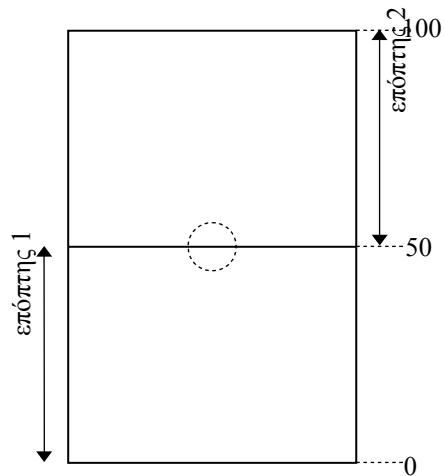
Αρχεία Εξόδου:

Τα αρχεία εξόδου με το όνομα **lines_man.out** είναι αρχεία κειμένου με την εξής δομή. Έχουν μία γραμμή με ακριβώς δύο αριθμούς, $L1$ και $L2$ ($0 \leq L1, L2 \leq 10.000.000$): τη συνολική απόσταση που διέτρεξε κάθε επόπτης γραμμών. Ο πρώτος κινείται από 0 έως $A/2$ και ο δεύτερος κινείται από $A/2$ έως A .



Παράδειγμα Αρχείων Εισόδου – Εξόδου

lines_man.in	lines_man.out
100	150 10
10	
49	
30	
25	
0	
50	
55	
40	
30	
20	
0	



Εξήγηση παραδείγματος:

Το μήκος του γηπέδου είναι 100, άρα ο πρώτος επόπτης κινείται μεταξύ 0 και 50 και ο δεύτερος μεταξύ 50 και 100 (βλ. σχήμα). Γίνονται συνολικά 10 φάσεις. Και οι δύο επόπτες ξεκινούν από τη θέση 50. Στις πρώτες τέσσερις φάσεις (49, 30, 25 και 0) ο πρώτος επόπτης τρέχει μέχρι τη θέση 0 διανύοντας συνολικά 50 μέτρα, ενώ ο δεύτερος επόπτης παραμένει ακίνητος στη θέση 50. Στην πέμπτη φάση (50), ο πρώτος επόπτης επιστρέφει στη θέση 50 διανύοντας άλλα 50 μέτρα και ο δεύτερος παραμένει ακίνητος στη θέση 50. Στην έκτη φάση (55), ο δεύτερος επόπτης τρέχει μέχρι τη θέση 55 διανύοντας 5 μέτρα, ενώ ο πρώτος παραμένει ακίνητος στη θέση 50. Στην έβδομη φάση (40), ο δεύτερος επόπτης επιστρέφει στη θέση 50 διανύοντας άλλα 5 μέτρα και ο πρώτος επόπτης τρέχει μέχρι τη θέση 40 διανύοντας 10 μέτρα. Στις τελευταίες τρεις φάσεις (30, 20 και 0), ο πρώτος επόπτης τρέχει μέχρι τη θέση 0 διανύοντας συνολικά 40 μέτρα ενώ ο δεύτερος επόπτης παραμένει ακίνητος στη θέση 50. Οι συνολικές αποστάσεις που διανύθηκαν από τους δύο επόπτες είναι: $L1 = 50+50+10+40 = 150$ και $L2 = 5+5 = 10$.

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

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Μέγιστη διαθέσιμη μνήμη: 16 MB.



ΘΕΜΑ 2ο: Servers

[35 Μονάδες]

Το Δίκτυο των Ελληνικών Πανεπιστημίων και Α-ΤΕΙ (<http://www.gunet.gr>) έχει αναπτύξει μια συστοιχία (cluster) από N web servers. Κάθε server:

- διαχειρίζεται τις ίδιες M σελίδες,
- κρατάει ένα μετρητή για κάθε σελίδα, που αυξάνει κατά 1 όταν ζητηθεί η σελίδα αυτή από τον server, και
- κρατάει τις τιμές αυτών των μετρητών ταξινομημένες σε φθίνουσα σειρά.

Πρόβλημα:

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του IOI το οποίο: Όταν δίνεται το πλήθος αιτημάτων σε κάθε σελίδα, σε κάθε server και σε ταξινομημένη σειρά, θα βρίσκει τις σελίδες με τις K πιο πολλές αναζητήσεις σε όλους τους servers.

Παράδειγμα:

Για $N=5$ servers και $M=5$ σελίδες

S1	S2	S3	S4	S5
3 ↦ 99	1 ↦ 91	1 ↦ 92	3 ↦ 74	3 ↦ 67
1 ↦ 66	3 ↦ 90	3 ↦ 75	1 ↦ 56	4 ↦ 67
0 ↦ 63	0 ↦ 61	4 ↦ 70	2 ↦ 56	1 ↦ 58
2 ↦ 48	4 ↦ 07	2 ↦ 16	0 ↦ 28	2 ↦ 54
4 ↦ 44	2 ↦ 01	0 ↦ 01	4 ↦ 19	0 ↦ 35

Επεξήγηση: Οι σελίδες έχουν αριθμηθεί από 0 έως $M-1$. Στο server S1 η σελίδα 3 έχει ζητηθεί 99 φορές, η 1 66, η 0 63 κοκ. Συνολικά τα αιτήματα που υποβλήθηκαν για τις 5 σελίδες ήταν:

Top 5
3 ↦ 405
1 ↦ 363
4 ↦ 207
0 ↦ 188
2 ↦ 175

Για $K=2$, οι δύο δημοφιλέστερες σελίδες είναι η 3 και η 1.

Αρχεία Εισόδου:

Τα αρχεία εισόδου με όνομα **servers.in** είναι αρχεία κειμένου με την εξής δομή: Στην πρώτη γραμμή έχουν τρεις αριθμούς M , N , K . Τον αριθμό M των φιλοξενούμενων σελίδων ($5 \leq M \leq 10.000$). Σημειώνεται ότι η πρώτη σελίδα αριθμείται με 0 . Τον αριθμό N των

Σελ. 3 από 8



servers ($5 \leq N \leq 1.000$). Τον αριθμό K των δημοφιλέστερων σελίδων που ζητούνται ($2 \leq K \leq M$). Οι αριθμοί ξεχωρίζουν μεταξύ τους με ένα κενό. Στις επόμενες M γραμμές υπάρχουν N ζεύγη ακεραίων. Η πρώτη γραμμή έχει για κάθε server τη σελίδα που ζητήθηκε περισσότερο, η δεύτερη την αμέσως επόμενη, κ.ο.κ. Σε κάθε ζεύγος, ο πρώτος ακεραίος είναι ο αριθμός της σελίδας και ο δεύτερος πόσες φορές αυτή ζητήθηκε. Όλοι οι αριθμοί ξεχωρίζουν μεταξύ τους με ένα κενό. Καμία σελίδα δε θα έχει ζητηθεί από κάποιον server περισσότερες από 500.000 φορές.

Αρχεία Εξόδου:

Τα αρχεία εξόδου με το όνομα **servers.out** είναι αρχεία κειμένου με την εξής δομή. Έχουν ακριβώς K γραμμές. Σε κάθε γραμμή υπάρχουν δύο ακεραίοι: ο αριθμός σελίδας και ο συνολικός αριθμός των φορών που αυτή αναζητήθηκε. Οι σελίδες είναι ταξινομημένες με φθίνουσα σειρά συνολικού αριθμού αιτήσεων (η δημοφιλέστερη πρώτη). Σελίδες με ίσο συνολικό αριθμό αιτήσεων μπορούν να εμφανίζονται με οποιαδήποτε σειρά. Επίσης, σε περίπτωση που η $(K+1)$ -οστή δημοφιλέστερη σελίδα έχει ίσο συνολικό αριθμό αιτήσεων με την K -οστή, είναι αδιάφορο ποια από τις δύο θα εμφανιστεί στο αρχείο εξόδου και ποια δε θα εμφανιστεί καθόλου.

Παράδειγμα Αρχείων Εισόδου – Εξόδου

servers.in	servers.out
5 5 2	3 405
3 99 1 91 1 92 3 74 3 67	1 363
1 66 3 90 3 75 1 56 4 67	
0 63 0 61 4 70 2 56 1 58	
2 48 4 07 2 16 0 28 2 51	
4 44 2 01 0 01 4 19 0 35	

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

Μέγιστος χρόνος εκτέλεσης: 1 sec

Μέγιστη διαθέσιμη μνήμη: 64 MB.



ΘΕΜΑ 3ο: Get out!

[40 Μονάδες]

Ένα τετράγωνος χώρος στάθμευσης (parking) αυτοκινήτων έχει μήκος πλευράς N μέτρα. Σε αυτόν βρίσκονται σταθμευμένα αυτοκίνητα διαστάσεων 2×1 μέτρα. Τα αυτοκίνητα είναι σταθμευμένα παράλληλα προς τις πλευρές του τετραγώνου και σε ακέραιες συντεταγμένες, άλλα οριζόντια (κατά μήκος σειρών) και άλλα κάθετα (κατά μήκος στηλών). Κάθε αυτοκίνητο μπορεί να μετακινείται μόνο κατά μήκος του άξονά του και μέσα στο τετράγωνο, εφόσον δεν εμποδίζεται από άλλο αυτοκίνητο, και δεν μπορεί να περιστρέφεται.

Η εταιρεία παροχής φυσικού αερίου θέλει να σκάψει κατά μήκος μίας οριζόντιας γραμμής (σειράς) στη θέση R μέσα στο χώρο στάθμευσης, για να περάσει έναν υπόγειο αγωγό. Για το σκοπό αυτό πρέπει να μετακινηθούν τα αυτοκίνητα που βρίσκονται πάνω σε αυτή τη σειρά.

Πρόβλημα:

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του IOI το οποίο: Αφού διαβάσει το μήκος της πλευράς του τετραγώνου, τη θέση όπου θα τοποθετηθεί ο αγωγός, και τις θέσεις των σταθμευμένων αυτοκινήτων, θα υπολογίζει τον ελάχιστο αριθμό κινήσεων που απαιτούνται για να ελευθερωθεί η σειρά του αγωγού. Σε κάθε κίνηση μετακινείται ένα αυτοκίνητο κατά μήκος του άξονά του (εμπρός ή πίσω), μία ή περισσότερες θέσεις.

Αρχεία Εισόδου:

Τα αρχεία εισόδου με όνομα **getout.in** είναι αρχεία κειμένου με την εξής δομή: Στην πρώτη γραμμή έχουν δύο ακέραιους αριθμούς: το μήκος N της πλευράς του τετραγώνου ($4 \leq N \leq 8$) και τη θέση R της σειράς όπου θα τοποθετηθεί ο αγωγός ($1 \leq R \leq N$). Οι επόμενες N γραμμές περιγράφουν τα αυτοκίνητα που είναι σταθμευμένα οριζόντια, κατά μήκος των σειρών του τετραγώνου. Κάθε γραμμή αρχίζει με το πλήθος X_i των αυτοκινήτων που είναι σταθμευμένα κατά μήκος αυτής της σειράς, ακολουθούμενο από X_i ακέραιους αριθμούς: τις συντεταγμένες όπου βρίσκονται τα αυτοκίνητα κατά μήκος της σειράς. (Οι συντεταγμένες είναι μεταξύ 1 και N . Ένα αυτοκίνητο οριζόντια σταθμευμένο με συντεταγμένη 5 βρίσκεται πάνω στις στήλες 5 και 6 του τετραγώνου.) Ομοίως, οι επόμενες N γραμμές περιγράφουν τα αυτοκίνητα που είναι τοποθετημένα κάθετα, κατά μήκος των στηλών του τετραγώνου.



Αρχεία Εξόδου:

Τα αρχεία εξόδου με το όνομα **getout.out** είναι αρχεία κειμένου με μόνο μία γραμμή που περιέχει μόνο έναν αριθμό: το ελάχιστο πλήθος κινήσεων. Αν δεν είναι δυνατόν να ελευθερωθεί η σειρά **R** για να τοποθετηθεί ο αγωγός, το αποτέλεσμα θα είναι **-1**.

Παράδειγμα Αρχείων Εισόδου – Εξόδου

getout.in	getout.out
5 4	4
2 2 4	
0	
1 2	
0	
0	
1 3	
0	
1 4	
1 3	
0	

5			E		
4	Δ		E	Z	
3	Δ	Γ	Γ	Z	
2					
1		A	A	B	B
	1	2	3	4	5

Εξήγηση παραδείγματος:

Για να ελευθερωθεί η σειρά 4 του τετραγώνου, αρκεί να μετακινηθεί το αυτοκίνητο Δ δύο θέσεις κάτω, μετά το Z μία θέση κάτω, μετά το Γ μία θέση αριστερά, και τέλος το E δύο θέσεις κάτω. Χρειάζονται επομένως 4 κινήσεις.

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

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Μέγιστη διαθέσιμη μνήμη: 128 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: lines_man  
*/ για κώδικα σε C  
/*  
USER: username  
LANG: C++  
TASK: getout  

```

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 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++ -O2 -DCONTEST -s -static -lm
- Pascal: gpc -O2 -DCONTEST -s

Σελ. 7 από 8



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) (λη για προγραμματιστές C ή C++, \$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.
- ✓ 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, Α-ΤΕΙ Αθήνας.

