



30^{ος} ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΜΑΤΑ ΤΕΛΙΚΗΣ ΦΑΣΗΣ

Θέμα 1^ο: Rafting

[30 Μονάδες]

Η κατάβαση ποταμών με σχεδία – φουσκωτή βάρκα (rafting) είναι μία πολύ συναρπαστική και διασκεδαστική εμπειρία. Στο rafting το πλήρωμα αποτελείται από 6 έως 12 κωπηλάτες που κατευθύνουν τη φουσκωτή βάρκα μέσα από ποτάμια με βαθμό δυσκολίας από 1^ο–7^ο, ανάλογα με την ορμητικότητα των νερών και το ανάγλυφο της κοίτης. Στην Ελλάδα, δεκάδες ποταμοί μεταξύ των οποίων ο Αχέροντας, ο Βοϊδομάτης, ο Ερύμανθος, και ο Λούσιος τους χειμερινούς μήνες, γίνονται θέρετρα χειμερινών δραστηριοτήτων.

Το αθλητικό rafting συνίσταται στην κατάβαση του ποταμού με συγκεκριμένη διαδρομή, μέσω επιλεγμένων εμποδίων, στον μικρότερο χρόνο. Κάθε φουσκωτή βάρκα κατεβαίνει τη διαδρομή περνώντας υποχρεωτικά από τα επιλεγμένα εμπόδια (έγκυρη κατάβαση) και, με βάση το χρόνο της, οι πίνακες τερματισμού δίνουν τη θέση της στη γενική κατάταξη. Η πρώτη βάρκα για παράδειγμα θα έχει αναγκαστικά θέση 1. Η δεύτερη 1 ή 2 και η Ν-οστή οποιαδήποτε θέση από 1 έως Ν.

Πρόβλημα:

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες του IOI (PASCAL, C, C++, Java) το οποίο αφού διαβάσει τη θέση που παίρνει η κάθε σχεδία στη γενική κατάταξη μέχρι εκείνη τη στιγμή, θα υπολογίζει την τελική θέση κάθε σχεδίας μετά το τέλος του αγώνα.

Αρχεία Εισόδου:

Τα αρχεία εισόδου με όνομα **rafting.in** είναι αρχεία κειμένου με την εξής δομή: Η πρώτη γραμμή έχει έναν ακέραιο αριθμό **N**, το πλήθος των σχεδιών που λαμβάνουν μέρος στον αγώνα. Η δεύτερη γραμμή περιέχει Ν ακέραιους αριθμούς χωρισμένους ανά δύο με ένα κενό διάστημα: τις θέσεις στη γενική κατάταξη που έχει μέχρι εκείνη τη στιγμή η αντίστοιχη σχεδία.

Αρχεία Εξόδου:

Τα αρχεία εξόδου με όνομα **rafting.out** είναι αρχεία κειμένου με την εξής δομή: Έχουν μία γραμμή που περιέχει Ν ακέραιους αριθμούς, όσες και οι σχεδίες, χωρισμένους ανά δύο με ένα κενό διάστημα. Κάθε αριθμός πρέπει να είναι ο αύξων αριθμός της σχεδίας που θα καταλάβει την αντίστοιχη θέση στην τελική κατάταξη.

Σελίδα 1 από 8



Παραδείγματα Αρχείων Εισόδου - Εξόδου:

1°

rafting.in	rafting.out
10 1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10

2°

rafting.in	rafting.out
10 1 1 1 1 1 1 1 1 1 1	10 9 8 7 6 5 4 3 2 1

3°

rafting.in	rafting.out
7 1 1 3 2 3 1 5	6 2 4 5 7 1 3

Εξήγηση: Στο 1° παράδειγμα, κάθε σχεδία που τερματίζει κάνει χειρότερο χρόνο από τις προηγούμενες και κατατάσσεται τελευταία, ενώ στο 2° κάθε σχεδία κάνει καλύτερο χρόνο από τις προηγούμενες και κατατάσσεται πρώτη.

Περιορισμοί:

- Για περιπτώσεις ελέγχου συνολικής αξίας 50%, θα είναι:
 $1 \leq N \leq 10.000$
- Για περιπτώσεις ελέγχου συνολικής αξίας 100%, θα είναι:
 $1 \leq N \leq 500.000$

Προσοχή! Φροντίστε να διαβάσετε 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: Παιχνίδι με λίστα**[35 Μονάδες]**

Δίνεται μία λίστα που αποτελείται από **N** φυσικούς αριθμούς. Δύο παίκτες παίζουν το εξής παιχνίδι επαναλαμβάνοντας, πρώτα ο ένας και μετά ο άλλος, την εξής κίνηση: επιλέγουν έναν από τους αριθμούς που βρίσκονται στα άκρα της λίστας (αριστερά ή δεξιά), τον αφαιρούν από τη λίστα και τον προσθέτουν στο σκορ τους. Το παιχνίδι τελειώνει όταν εξαντληθούν οι αριθμοί της λίστας. Κάθε παίκτης προσπαθεί να μεγιστοποιήσει το τελικό του σκορ.

Βρείτε ποιο είναι το μέγιστο σκορ που μπορεί να έχει στο τέλος του παιχνιδιού ο πρώτος παίκτης, αν και αυτός και ο αντίπαλός του παίζουν με βέλτιστο τρόπο.

Πρόβλημα

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες της IOI (Pascal, C, C++, Java) το οποίο θα διαβάζει τα δεδομένα και θα βρίσκει το μέγιστο δυνατό σκορ του πρώτου παίκτη αν και οι δύο παίζουν με βέλτιστο τρόπο.

Αρχεία εισόδου:

Το αρχείο εισόδου με όνομα **listgame.in** είναι αρχείο κειμένου με ακριβώς δύο γραμμές. Η πρώτη γραμμή του περιέχει έναν ακέραιο αριθμό **N**, το πλήθος των αριθμών της λίστας. Η δεύτερη γραμμή περιέχει ακριβώς **N** ακέραιους αριθμούς, χωρισμένους ανά δύο με ένα κενό διάστημα: τους **N** αριθμούς της λίστας.

Αρχεία εξόδου:

Το αρχείο εξόδου με όνομα **listgame.out** είναι αρχείο κειμένου αποτελούμενο από μία γραμμή που θα περιέχει έναν ακέραιο αριθμό: το μέγιστο δυνατό σκορ του πρώτου παίκτη αν και οι δύο παίζουν με βέλτιστο τρόπο.

Παραδείγματα αρχείων εισόδου – εξόδου**1^ο**

listgame.in	listgame.out
10 6 3 9 2 10 4 9 2 3 7	38

Εξήγηση: Οι δύο παίκτες μπορούν να επιλέξουν να κάνουν τις παρακάτω κινήσεις, κατά σειρά. Οι κινήσεις αυτές είναι βέλτιστες και για τους δύο παίκτες.



Κίνηση	1 ^{ος} παίκτης	2 ^{ος} παίκτης	η λίστα μετά την κίνηση
0			6 3 9 2 10 4 9 2 3 7
1	7		6 3 9 2 10 4 9 2 3
2		6	3 9 2 10 4 9 2 3
3	3		3 9 2 10 4 9 2
4		3	9 2 10 4 9 2
5	9		2 10 4 9 2
6		2	2 10 4 9
7	9		2 10 4
8		4	2 10
9	10		2
10		2	
σκορ:	38	17	

2^ο

listgame.in	listgame.out
5 10 2 1 4 3	14

3^ο

listgame.in	listgame.out
6 4 8 0 1 5 6	15

Περιορισμοί:

- Το άθροισμα όλων των αριθμών της λίστας δε θα ξεπερνά το 10^9 .
- Για περιπτώσεις ελέγχου συνολικής αξίας 25%, θα είναι: $1 \leq N \leq 20$
- Για περιπτώσεις ελέγχου συνολικής αξίας 100%, θα είναι: $1 \leq N \leq 10.000$

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

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Μέγιστη διαθέσιμη μνήμη: 64 MB.



Θέμα 3^ο: Ο Χρήστος και τα τούβλα

[35 Μονάδες]

Ο Χρήστος είναι Πολιτικός Μηχανικός. Η πτυχιακή του εργασία αφορά την αστοχία των υλικών στα κτίρια. Συγκεκριμένα, ο Χρήστος προσομοιώνει την κατασκευή ενός τοίχου σαν ένα πλέγμα διαστάσεων $N \times M$, του οποίου κάθε τετραγωνάκι μπορεί να περιέχει ένα τούβλο (που συμβολίζεται με "#") ή να είναι κενό (που συμβολίζεται με "."). Στην κορυφή του πλέγματος, την οποία μπορείτε να φαντάζεστε σαν το ταβάνι, υπάρχει μια στρώση μπετόν και επομένως όσα τούβλα βρίσκονται στην πιο πάνω γραμμή του πλέγματος είναι κολλημένα στο ταβάνι και δεν πέφτουν. Όλα τα τούβλα έχουν μπετόν σε κάθε πλευρά τους και άρα όσα ακουμπάνε σε τούβλα της πιο πάνω γραμμής επίσης δεν πέφτουν, όσα ακουμπάνε σε αυτά επίσης δεν πέφτουν, κ.ο.κ. Θεωρούμε πως ένα τούβλο μπορεί να ακουμπάει σε άλλα τούβλα που βρίσκονται στα τέσσερα γειτονικά τετράγωνα του πλέγματος: αριστερά, δεξιά, πάνω και κάτω.

Ο Χρήστος θέλει να μελετήσει τη συμπεριφορά του τοίχου αν κάποια τούβλα "αστοχήσουν". Συγκεκριμένα, θέλει να κάνει το εξής πείραμα: Επιλέγει ένα σύνολο από τούβλα-στόχους και πηγαίνει και τα σπάει, ένα-ένα. Κάθε φορά που σπάει ένα τούβλο-στόχος, είναι πιθανό ένα κομμάτι της κατασκευής να καταρρεύσει, αφού πλέον δεν θα συνδέεται με το υπόλοιπο στέρεο κομμάτι. Ο Χρήστος θέλει να ξέρει κάθε φορά που σπάει ένα τούβλο-στόχο πόσο μεγάλο θα είναι το κομμάτι που καταρρέει, δηλαδή πόσα τούβλα περιέχει.

Πρόβλημα

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες της IOI (Pascal, C, C++, Java) το οποίο θα διαβάζει την αρχική κατάσταση του πλέγματος και το σύνολο των τούβλων-στόχων και θα βρίσκει το πλήθος των τούβλων που καταρρέουν μετά από κάθε σπάσιμο.

Αρχεία εισόδου:

Το αρχείο εισόδου με όνομα **brickbreaker.in** είναι αρχείο κειμένου. Η πρώτη γραμμή περιέχει δύο ακραίους N και M χωρισμένους μεταξύ τους με ένα κενό διάστημα: τον αριθμό των γραμμών και των στηλών του πλέγματος αντίστοιχα. Οι επόμενες N γραμμές περιέχουν την περιγραφή του πλέγματος: κάθε μία από αυτές θα περιέχει ακριβώς M χαρακτήρες, ο καθένας εκ των οποίων θα είναι "#" ή "." (τελεία) ανάλογα αν το αντίστοιχο τετράγωνο περιέχει τούβλο ή είναι κενό. Η επόμενη γραμμή θα περιέχει έναν ακέραιο αριθμό Q : το πλήθος των τούβλων-στόχων που θα σπάσει ο Χρήστος. Κάθε μία από τις επόμενες Q γραμμές περιέχει δύο ακραίους αριθμούς R_i και C_i : τις συντεταγμένες στο πλέγμα (γραμμή και στήλη) του αντίστοιχου

Σελίδα 5 από 8



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: rafting
*/                                για κώδικα σε C

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

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

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

2. Έλεγχος τιμών δεν απαιτείται. Οι τιμές των αρχείων ελέγχου είναι πάντα έγκυρες.

Σελίδα 7 από 8



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`
 - 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. `rafting`). 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.

