

33^{ος} ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΜΑΤΑ ΤΕΛΙΚΗΣ ΦΑΣΗΣ

Θέμα 1^ο: Ο πόλεμος των τσιφλικάδων

[30 Μονάδες]

Δύο μεγαλογαιοκτήμονες (τσιφλικάδες) είναι η αιτία μεγάλου πονοκέφαλου για το βασιλιά. Τσακώνονται μεταξύ τους για να αποκτήσουν περισσότερα χωράφια και η ιστορία αυτή δε θα έχει καλή έκβαση. Ο βασιλιάς αναλαμβάνει να βρει λύση στη διαμάχη τους και ζητά τη βοήθειά σας.

Στο δρόμο που ενώνει τα δύο τσιφλίκια υπάρχουν N χωράφια. Γνωρίζουμε τα οικονομικά στοιχεία κάθε χωραφιού και τα αναπαριστούμε με ακέραιους αριθμούς $x_1, x_2, \dots, x_N$. Μερικά χωράφια βγάζουν κέρδος, οπότε $x_i > 0$, άλλα έχουν ζημιά, οπότε $x_i < 0$, και κάποια δεν έχουν ούτε κέρδος ούτε ζημιά, οπότε $x_i = 0$. Οι τσιφλικάδες προφανώς προτιμούν να έχουν στην κατοχή τους χωράφια που βγάζουν κέρδος, παρά χωράφια με ζημιά.

Ο βασιλιάς θέλει να δώσει στον ένα τσιφλικά τα L χωράφια που είναι κοντινότερα στο τσιφλίκι του, ξεκινώντας δηλαδή από το ένα άκρο του δρόμου, και στον άλλο τσιφλικά τα R χωράφια που είναι κοντινότερα στο δικό του τσιφλίκι, ξεκινώντας δηλαδή από το άλλο άκρο του δρόμου. Φυσικά, οι τιμές των L και R θα πρέπει να είναι μη αρνητικές και τέτοιες ώστε $L+R \leq N$. Ο βασιλιάς θέλει να μην αδικήσει κανένα τσιφλικά, επομένως το άθροισμα των x_i των L χωραφιών που δίνει στον έναν θα πρέπει να ισούται με το άθροισμα των x_i των R χωραφιών που δίνει στον άλλον. Επίσης θέλει να αφήσει όσο γίνεται λιγότερα χωράφια αδιάθετα.

Ποια είναι η ελάχιστη δυνατή τιμή του $N-(L+R)$ που μπορεί να επιτευχθεί;

Πρόβλημα:

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες της IOI (PASCAL, C, C++, Java) το οποίο θα διαβάσει την τιμή του N και την ακολουθία των αριθμών x_i και θα εκτυπώνει την απάντηση στο παραπάνω ερώτημα.

Αρχεία εισόδου:

Το αρχείο εισόδου με όνομα **landfight.in** είναι αρχείο κειμένου με την εξής δομή. Η πρώτη γραμμή έχει έναν ακέραιο αριθμό N : το πλήθος των χωραφιών. Η επόμενη γραμμή περιέχει N ακέραιους αριθμούς x_i , χωρισμένους ανά δύο με ένα κενό διάστημα: τα οικονομικά στοιχεία των χωραφιών.

Αρχεία εξόδου:

Το αρχείο εξόδου με όνομα **landfight.out** είναι αρχείο κειμένου που περιέχει μία μόνο γραμμή με έναν ακέραιο αριθμό: την απάντηση στο παραπάνω ερώτημα.

Σελίδα 1 από 9



Παραδείγματα αρχείων εισόδου - εξόδου:

1°

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

Εξήγηση: Μπορούμε να δώσουμε $L=3$ χωράφια στον αριστερό τσιφλικά, με άθροισμα $3+2+5=10$, και $R=2$ χωράφια στον δεξιό τσιφλικά, πάλι με άθροισμα $6+4=10$. Έτσι μένουν αδιάθετα δύο χωράφια.

2°

landfight.in	landfight.out
10 5 -2 3 -7 -4 -1 3 -5 6 0	0

Εξήγηση: Μπορούμε να δώσουμε $L=4$ χωράφια στον αριστερό τσιφλικά, με άθροισμα $5-2+3-7=-1$, και $R=6$ χωράφια στον δεξιό τσιφλικά, πάλι με άθροισμα $-4-1+3-5+6+0=-1$. Έτσι δε μένει αδιάθετο κανένα χωράφι.

3°

landfight.in	landfight.out
5 1 2 3 4 5	5

Εξήγηση: Δεν υπάρχει κανένας τρόπος να δώσουμε L χωράφια στον έναν και R χωράφια στον άλλο, έτσι ώστε τα αθροίσματα των x_i να είναι ίσα. Επομένως, και τα 5 χωράφια θα μείνουν αδιάθετα.

Περιορισμοί:

- $1 \leq N \leq 1.000.000$
- Το άθροισμα των απόλυτων τιμών όλων των x_i δε θα υπερβαίνει το 10^9
- Για περιπτώσεις ελέγχου συνολικής αξίας 20%, θα είναι $1 \leq N \leq 1000$
- Για περιπτώσεις ελέγχου συνολικής αξίας 50%, θα είναι $1 \leq N \leq 30.000$
- Για περιπτώσεις ελέγχου συνολικής αξίας 70%, θα είναι $x_i > 0$

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: Γυρίζοντας στο σπίτι

[30 Μονάδες]

Είναι άλλη μια συνηθισμένη μέρα μέσα στην καραντίνα. Ως σωστοί προγραμματιστές έχετε φάει όλη σας τη μέρα να ψάχνετε ένα ύπουλο bug και αποφασίζετε να πάτε έναν απογευματινό περίπατο για να σκεφτείτε πιο καθαρά. Απορροφημένοι στην σκέψη σας, όμως, χάνετε την αίσθηση του χρόνου. Η ώρα είναι 9 και η απαγόρευση κυκλοφορίας μόλις αρχίζει. Γνωρίζετε ότι στις 9 ακριβώς ένας αστυνόμος ξεκινάει την περιπολία του από το αστυνομικό τμήμα του δήμου και, αν σας δει, θα σας επιβάλει ένα βαρύ πρόστιμο. Θέλετε να γυρίσετε στο σπίτι γλιτώνοντας το πρόστιμο. Είναι η ευκαιρία σας να αξιοποιήσετε τις αλγοριθμικές γνώσεις που έχετε αποκτήσει όλον αυτόν τον καιρό μέσα στην καραντίνα!

Ο χάρτης του δήμου σας αναπαρίσταται με ένα πλέγμα διαστάσεων $N \times M$. Κάθε τετράγωνο του πλέγματος μπορεί να είναι δρόμος ή κτίριο. Επιπλέον είναι γνωστή η αρχική σας θέση, η θέση του αστυνομικού τμήματος και η θέση του σπιτιού σας.

Πρέπει να σκεφτείτε μια προκαθορισμένη διαδρομή που να ξεκινάει από την αρχική σας θέση, να καταλήγει στο σπίτι σας και να αποτελείται από μια ακολουθία κινήσεων. Σε κάθε κίνηση μπορείτε να προχωρήσετε μόνο (οριζόντια ή κατακόρυφα) σε κάποιο γειτονικό τετράγωνο που δεν είναι κτίριο. Ταυτόχρονα, ο αστυνόμος μπορεί να κινηθεί με παρόμοιο τρόπο ή να μείνει ακίνητος. Η κίνησή σας μαζί με την αντίδραση του αστυνομικού ολοκληρώνονται σε ένα λεπτό.

Μετά από κάθε λεπτό μπορούν τα συμβούν τα εξής:

- Αν είστε στο οπτικό πεδίο του αστυνομικού, δηλαδή στην ίδια γραμμή (οριζόντια) ή στήλη (κατακόρυφα) χωρίς κάποιο κτίριο ανάμεσά σας, τότε παίρνετε πρόστιμο.
- Αν δεν έχετε πάρει πρόστιμο και είστε στο σπίτι σας, τότε μπαίνετε μέσα ασφαλείς.

Πρόβλημα

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες της IOI (Pascal, C, C++, Java) το οποίο θα διαβάσει την περιγραφή του δήμου σας και θα αποφασίζει αν υπάρχει προκαθορισμένη διαδρομή η οποία να οδηγεί στο σπίτι σας με ασφάλεια, ανεξάρτητα από το πώς κινείται ο αστυνόμος.

Αρχεία εισόδου:

Το αρχείο εισόδου με όνομα **wayhome.in** είναι αρχείο κειμένου με την εξής δομή. Η πρώτη γραμμή περιέχει έναν ακέραιο αριθμό **T**, τον αριθμό των σεναρίων του προβλήματος. Στη συνέχεια ακολουθούν **T** σενάκια, κάθε ένα με την παρακάτω δομή.

Η πρώτη γραμμή περιέχει δύο ακέραιους αριθμούς **N**, **M**, τις διαστάσεις του δόμου. Κάθε μία από τις επόμενες **N** γραμμές περιέχει **M** χαρακτήρες, που περιγράφουν τα αντίστοιχα τετράγωνα. Κάθε χαρακτήρας μπορεί να είναι είτε **.** (δρόμος), **#** (κτίριο), **Y** (η θέση σας), **P** (η θέση του αστυνόμου), ή **H** (η θέση του σπιτιού). Θεωρήστε δεδομένο ότι κάθε χαρακτήρας από τους **Y**, **P** και **H** θα εμφανίζεται ακριβώς μία φορά και θα μπορεί να χρησιμοποιηθεί ως δρόμος για τις μετακινήσεις.

Αρχεία εξόδου:

Το αρχείο εξόδου με όνομα **wayhome.out** είναι αρχείο κειμένου με την εξής δομή. Αποτελείται από **T** γραμμές, κάθε μία από τις οποίες αντιστοιχεί στην απάντηση ενός σεναρίου και περιέχει ακριβώς μια συμβολοσειρά: **YES** αν υπάρχει ασφαλής διαδρομή προς το σπίτι και **NO** διαφορετικά.

Παράδειγμα αρχείου εισόδου – εξόδου

wayhome.in	wayhome.out
3	YES
5 7	NO
Y.....P	NO
..#.....	
..#####	
.....	
...H...	
5 7	
Y....P.	
..#.....	
..#####	
.....	
...H...	
2 3	
.YH	
P##	

Εξήγηση 1^{ου} σεναρίου: Μια ασφαλής διαδρομή προς το σπίτι είναι η εξής: κάτω, κάτω, κάτω, δεξιά, δεξιά, δεξιά, κάτω. Προσέξτε ότι ο αστυνόμος δεν μπορεί να σας δώσει πρόστιμο στην αρχή αφού η απαγόρευση της κυκλοφορίας αρχίζει από το πρώτο λεπτό μετά τις 9.

Εξήγηση 3^{ου} σεναρίου: Προσέξτε ότι για να μπείτε ασφαλείς στο σπίτι δεν πρέπει να σας βλέπει ο αστυνόμος.

Περιορισμοί:

- $1 \leq T \leq 12$
- $1 \leq N, M \leq 700$



- Για περιπτώσεις ελέγχου συνολικής αξίας 17%, ο χάρτης δεν θα έχει καθόλου κτίρια.
- Για περιπτώσεις ελέγχου συνολικής αξίας 42%, θα είναι $1 \leq N, M \leq 200$.

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 3^ο: Ταξίδι στη Νέα Υόρκη

[40 Μονάδες]

Μετά την άρση των περιοριστικών μέτρων κατά του COVID-19, ο Μίλτος το δελφίνι αποφάσισε να κάνει δώρο στον εαυτό του ένα ταξίδι αναψυχής στην πανέμορφη Νέα Υόρκη και να θαυμάσει μερικούς από τους πιο εντυπωσιακούς και ιστορικούς ουρανοξύστες του κόσμου. Η ώρα φτάνει για το ταξίδι και οι μόνες αποσκευές του θα είναι το μαγικό του και η φωτογραφική του μηχανή.

Η Νέα Υόρκη είναι διάσημη για την «κορυφογραμμή των ουρανοξυστών» (Manhattan skyline). Σε αυτό το πρόβλημα, θα θεωρήσουμε ότι η κορυφογραμμή αποτελείται από N ουρανοξύστες που βρίσκονται ο ένας δίπλα στον άλλον σε μια γραμμή και καθένας έχει διαφορετικό ύψος H_i και πλάτος W_i .

Ο Μίλτος θέλει να φωτογραφήσει την κορυφογραμμή έτσι ώστε να απαθανατιστούν όλοι οι ουρανοξύστες (καθένας ολόκληρος) σε μία ή περισσότερες φωτογραφίες. Κάθε φωτογραφία που θα βγάλει μπορεί να καλύψει ένα τμήμα της κορυφογραμμής πλάτους L και, προφανώς, κάθε φωτογραφία πρέπει να χωράει τον ψηλότερο ουρανοξύστη στο αντίστοιχο τμήμα. Κατά την εκτύπωση των φωτογραφιών, ο Μίλτος δεν μπορεί να αλλάξει το πλάτος, μπορεί όμως να επιλέξει χαρτί κατάλληλου ύψους ώστε να χωράει ακριβώς ο ψηλότερος ουρανοξύστης.

Βοηθήστε το Μίλτο να εξοικονομήσει όσο περισσότερο χαρτί γίνεται στην εκτύπωση. Βοηθήστε τον δηλαδή να βρει με ποιον τρόπο πρέπει να τραβήξει τις φωτογραφίες για να ελαχιστοποιήσει το συνολικό άθροισμα των υψών των ψηλότερων ουρανοξυστών κάθε φωτογραφίας.

Πρόβλημα

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες της IOI (Pascal, C, C++, Java) το οποίο αφού διαβάζει τις τιμές των N και L και τα ύψη και πλάτη των N ουρανοξυστών, θα απαντάει στο παραπάνω ερώτημα.

Αρχεία εισόδου:

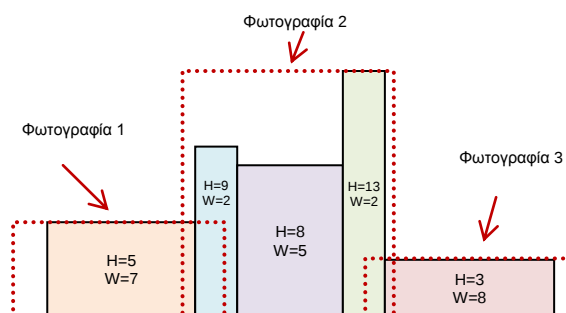
Το αρχείο εισόδου με όνομα **nytrip.in** είναι αρχείο κειμένου. Η πρώτη γραμμή περιέχει δύο φυσικούς αριθμούς χωρισμένους με ένα κενό διάστημα: το πλήθος N των ουρανοξυστών και το πλάτος L που καλύπτει κάθε φωτογραφία. Κάθε μία από τις επόμενες N γραμμές περιέχει δύο φυσικούς αριθμούς H_i και W_i , χωρισμένους με ένα κενό διάστημα: το ύψος και το πλάτος του i -οστού ουρανοξύστη.

Αρχεία εξόδου:

Το αρχείο εξόδου με όνομα **nytrip.out** είναι αρχείο κειμένου το οποίο περιέχει ακριβώς έναν αριθμό: το ελάχιστο δυνατό άθροισμα των υψών των ψηλότερων ουρανοξυστών κάθε φωτογραφίας.

Παράδειγμα αρχείων εισόδου – εξόδου

nytrip.in	nytrip.out
5 10 5 7 9 2 8 5 13 2 3 8	21



Εξήγηση: Έχουμε $N = 5$ ουρανοξύστες και κάθε φωτογραφία καλύπτει πλάτος $L = 10$. Το ελάχιστο δυνατό άθροισμα των υψών των ψηλότερων ουρανοξυστών από κάθε φωτογραφία το επιτυγχάνουμε τραβώντας 3 φωτογραφίες. Η πρώτη φωτογραφία θα περιέχει μόνο τον πρώτο ουρανοξύστη (5, 7). Η δεύτερη θα περιέχει τους επόμενους τρεις: (9, 2), (8, 5), (13, 2). Η τρίτη θα περιέχει μόνο τον τελευταίο (3, 8). Το άθροισμα των υψών των ψηλότερων ουρανοξυστών σε κάθε φωτογραφία είναι $5+13+3=21$.

Περιορισμοί:

- $1 \leq N \leq 1.000.000$, $1 \leq L \leq 1.000.000.000$
- $1 \leq H_i \leq 1.000.000$
- $1 \leq W_i \leq 1.000.000$ και $W_i \leq L$
- Για περιπτώσεις ελέγχου συνολικής αξίας 11%, θα είναι $1 \leq N \leq 25$
- Για περιπτώσεις ελέγχου συνολικής αξίας 23%, θα είναι $1 \leq N \leq 1000$
- Για περιπτώσεις ελέγχου συνολικής αξίας 41%, θα είναι $1 \leq N \leq 30.000$

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.



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: landfight  
*/  
για κώδικα σε C  
  
/*  
USER: username  
LANG: C++  
TASK: landfight  
*/  
για κώδικα σε C++  
  
(*  
USER: username  
LANG: PASCAL  
TASK: landfight  
*)  
για κώδικα σε PASCAL  
  
/*  
USER: username  
LANG: Java  
TASK: landfight  
*/  
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 8 από 9



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. `landfight`). 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.

