



29^{ος} ΠΑΝΕΛΛΗΝΙΟΣ ΔΙΑΓΩΝΙΣΜΟΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΘΕΜΑΤΑ ΤΕΛΙΚΗΣ ΦΑΣΗΣ

Θέμα 1^ο: Ακολουθίες DNA

[30 Μονάδες]

Οι βιολόγοι συσχετίζουν ακολουθίες DNA για να βγάλουν συμπεράσματα για κοινά χαρακτηριστικά. Σε ένα εργαστήριο πιστεύουν ότι ακολουθίες DNA με τα παρακάτω χαρακτηριστικά αναφέρονται σε άτομα με κοινά χαρακτηριστικά προσώπου.

Κάθε ακολουθία αποτελείται από N κεφαλαία γράμματα του λατινικού αλφάβητου. Τα στοιχεία των ακολουθιών είναι αριθμημένα από 1 έως και N . Λέμε ότι δυο ακολουθίες A και B **σχετίζονται** αν ισχύουν τα παρακάτω: υπάρχουν δυο ακέραιοι L και M ($1 \leq L \leq M \leq N$) έτσι ώστε

- $A[i] = B[i]$ για κάθε $i < L$
- $A[i] = B[i]$ για κάθε $i > M$
- $A[i] = B[M+L-i]$ για κάθε $L \leq i \leq M$

Πρόβλημα

Να αναπτύξετε ένα πρόγραμμα σε μια από τις γλώσσες της IOI (Pascal, C, C++, Java) το οποίο θα διαβάζει τα δεδομένα N , $A[i]$ και $B[i]$ και θα βρίσκει αν αυτές οι ακολουθίες σχετίζονται ή όχι.

Αρχεία εισόδου:

Το αρχείο εισόδου με όνομα **dna.in** είναι αρχείο κειμένου που περιέχει τρεις γραμμές. Η πρώτη γραμμή περιέχει έναν μόνο ακέραιο αριθμό N , το πλήθος των στοιχείων των ακολουθιών. Η δεύτερη γραμμή περιέχει τους N χαρακτήρες $A[i]$ (όπου $1 \leq i \leq N$) της πρώτης ακολουθίας. Ομοίως, η τρίτη γραμμή περιέχει τους N χαρακτήρες $B[i]$ της δεύτερης ακολουθίας.

Αρχεία εξόδου:

Το αρχείο εξόδου με όνομα **dna.out** είναι αρχείο κειμένου αποτελούμενο από μία μόνο γραμμή. Αν οι ακολουθίες σχετίζονται, αυτή η γραμμή πρέπει να περιέχει μόνο έναν ακέραιο, την τιμή της διαφοράς $M-L$. Αν υπάρχουν περισσότερες τιμές των L και M που επαληθεύουν τα παραπάνω, πρέπει να βρίσκετε την **ελάχιστη** δυνατή διαφορά $M-L$. Αν οι ακολουθίες δεν σχετίζονται, η γραμμή της εξόδου θα πρέπει να περιέχει τη λέξη «no».



Παραδείγματα αρχείων εισόδου – εξόδου

1°

dna.in	dna.out
12 BCBADABMAADA BCBAAAMBADDA	5

Εξήγηση: Οι τιμές $L=5$ και $M=10$ επαληθεύουν τα παραπάνω και δίνουν την ελάχιστη δυνατή τιμή της διαφοράς $M-L = 10-5 = 5$.

2°

dna.in	dna.out
5 ABACA ABACA	0

Εξήγηση: Οι τιμές $L=1$ και $M=1$ (και εν γένει κάθε $L=M$) επαληθεύουν τα παραπάνω.

3°

dna.in	dna.out
7 ABABCAA DAABBBA	no

Περιορισμοί:

- Για περιπτώσεις ελέγχου συνολικής αξίας 50%, θα είναι:
 $1 \leq N \leq 10.000$
- Για περιπτώσεις ελέγχου συνολικής αξίας 100%, θα είναι:
 $1 \leq N \leq 1.000.000$

Προσοχή! Φροντίστε να διαβάζετε και να επεξεργάζεστε τις ακολουθίες 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: Οδικό δίκτυο

[35 Μονάδες]

Μετά τον Β' παγκόσμιο πόλεμο, οι περισσότεροι δρόμοι που συνέδεαν χωριά έχουν καταστραφεί. Η χώρα έχει N χωριά, αριθμημένα από 1 έως και N , και M βατούς δρόμους διπλής κατεύθυνσης που καθένας συνδέει δύο διαφορετικά χωριά. Δεν υπάρχουν περισσότεροι από ένας δρόμοι που να συνδέουν τα ίδια χωριά. Αυτοί οι δρόμοι δεν εξασφαλίζουν την δυνατότητα οδικής σύνδεσης δυο οποιωνδήποτε χωριών. Υπάρχουν **ομάδες** χωριών, που σε κάθε ομάδα δύο οποιαδήποτε χωριά συνδέονται οδικώς μεταξύ τους. Όμως, δεν υπάρχει οδική σύνδεση ανάμεσα σε χωριά που ανήκουν σε διαφορετικές ομάδες.

Το κράτος σχεδιάζει την ανασυγκρότηση του οδικού δικτύου και έχει την δυνατότητα κατασκευής το πολύ K νέων δρόμων, καθένας από τους οποίους θα συνδέει δυο χωριά για τα οποία δεν υπάρχει οδική σύνδεση. Μετά την κατασκευή αυτών των νέων δρόμων, αν έχει γίνει σωστή σχεδίαση, το πλήθος των ομάδων χωριών θα μειωθεί γιατί περισσότερα χωριά θα συνδέονται οδικώς μεταξύ τους. Βρείτε ποιο είναι το ελάχιστο δυνατό πλήθος ομάδων που μπορεί να μείνει μετά την κατασκευή των νέων δρόμων.

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 ελάχιστο δυνατό πλήθος ομάδων χωριών που μένουν, μετά την κατασκευή των νέων δρόμων.

Αρχεία εισόδου:

Το αρχείο εισόδου με όνομα **villages.in** είναι αρχείο κειμένου. Η πρώτη γραμμή του περιέχει τρεις ακέραιους αριθμούς N , M , K , χωρισμένους ανά δύο με ένα κενό διάστημα: το πλήθος N των χωριών, το πλήθος M των υπάρχοντων δρόμων, και το πλήθος K των νέων δρόμων που πρόκειται να κατασκευαστούν. Κάθε μία από τις επόμενες M γραμμές περιέχει δύο ακέραιους αριθμούς A και B , χωρισμένους με ένα κενό διάστημα, όπου $1 \leq A, B \leq N$ και $A \neq B$. Αυτό σημαίνει ότι υπάρχει ένας βατός δρόμος που συνδέει τα χωριά A και B .

Αρχεία εξόδου:

Το αρχείο εξόδου με όνομα **villages.out** είναι αρχείο κειμένου αποτελούμενο από μία μόνο γραμμή που θα περιέχει έναν μόνο

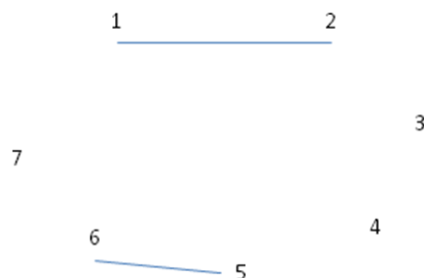


ακέραιο αριθμό: το ελάχιστο δυνατό πλήθος ομάδων που απομένουν, μετά την κατασκευή των νέων δρόμων.

Παραδείγματα αρχείων εισόδου - εξόδου

1°

villages.in	villages.out
7 2 2 1 2 6 5	3



Εξήγηση: Αρχικά υπάρχουν πέντε ομάδες χωριών: {1, 2}, {3}, {4}, {5, 6} και {7}. Αν κατασκευαστούν δύο νέοι δρόμοι, π.χ. μεταξύ των χωριών (3, 7) και των χωριών (3, 4), τότε θα προκύψουν μόνο τρεις ομάδες: {1, 2}, {3, 4, 7} και {5, 6}. Αυτός είναι και ο ελάχιστος αριθμός ομάδων που μπορούν να προκύψουν μετά την κατασκευή δύο δρόμων.

2°

villages.in	villages.out
4 2 3 1 2 4 3	1

3°

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

Εξήγηση: Στο 2° παράδειγμα, αρκεί να κατασκευαστεί ένας νέος δρόμος για να μείνει μόνο μία ομάδα που να περιέχει όλα τα χωριά. Στο 3° παράδειγμα, υπάρχει εξ αρχής μόνο μία ομάδα χωριών και αυτή θα παραμείνει, παρόλο που το κράτος δεν μπορεί να κατασκευάσει κανέναν δρόμο ($K=0$).

Περιορισμοί:

- Για περιπτώσεις ελέγχου συνολικής αξίας 50%, θα είναι:
 $1 \leq N \leq 10.000$, $1 \leq M \leq 20.000$, $0 \leq K \leq 10.000$
- Για περιπτώσεις ελέγχου συνολικής αξίας 100%, θα είναι:
 $1 \leq N \leq 1.000.000$, $1 \leq M \leq 2.000.000$, $0 \leq K \leq 1.000.000$

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

Μέγιστος χρόνος εκτέλεσης: 2 sec.

Μέγιστη διαθέσιμη μνήμη: 128 MB.



Θέμα 3^ο: Πάω διακοπές

[35 Μονάδες]

Η Κατερίνα δουλεύει πολύ και χρειάζεται επειγόντως διακοπές. Θέλει να φύγει ταξίδι στα νησιά και να έχει όσο γίνεται περισσότερες συνεχόμενες μέρες για να ξεκουραστεί, μπορεί όμως να κάνει μόνο ένα ταξίδι. Έστω ότι οι μέρες είναι αριθμημένες από 1 έως και N . Η Κατερίνα έχει στο πρόγραμμά της κάποιες υποχρεώσεις (συνολικά το πλήθος τους είναι M) τις οποίες έχει ήδη αναλάβει και που αντιστοιχούν σε κάποιες συγκεκριμένες μέρες: η i -οστή υποχρέωση είναι προγραμματισμένη για τη μέρα D_i .

Η Κατερίνα είναι διατεθειμένη να ακυρώσει το πολύ K από τις υποχρεώσεις της, προκειμένου να κάνει περισσότερες διακοπές. Βοηθήστε τη να βρει πόσες μέρες διακοπών μπορεί να κάνει.

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 πλήθος συνεχόμενων ημερών διακοπών που μπορεί να κάνει η Κατερίνα.

Αρχεία εισόδου:

Το αρχείο εισόδου με όνομα **maketime.in** είναι αρχείο κειμένου που η πρώτη γραμμή του περιέχει τρεις ακέραιους αριθμούς N , M και K , χωρισμένους ανά δύο με ένα κενό διάστημα: το συνολικό πλήθος N των ημερών, το πλήθος M των υποχρεώσεων που έχει προγραμματίσει η Κατερίνα, και το μέγιστο αριθμό K των υποχρεώσεων που είναι διατεθειμένη να ακυρώσει. Η δεύτερη γραμμή περιέχει ακριβώς M ακέραιους αριθμούς D_i (όπου $1 \leq i \leq M$ και $1 \leq D_i \leq N$): τις ημέρες που είναι προγραμματισμένες οι M υποχρεώσεις της Κατερίνας.

Αρχεία εξόδου:

Το αρχείο εξόδου με όνομα **maketime.out** είναι αρχείο κειμένου αποτελούμενο από μία γραμμή που θα περιέχει έναν ακέραιο αριθμό: το μέγιστο πλήθος ημερών που η Κατερίνα μπορεί να κάνει διακοπές.



Παραδείγματα αρχείων εισόδου – εξόδου

1°

maketime.in	maketime.out
10 5 2 6 9 3 2 7	5

Εξήγηση: Η Κατερίνα μπορεί να ακυρώσει τις υποχρεώσεις που έχει προγραμματίσει για τις μέρες 2 και 3. Έτσι, μπορεί να κάνει διακοπές από την 1^η μέρα έως και την 5^η, να έχει δηλαδή συνολικά 5 μέρες διακοπών.

2°

maketime.in	maketime.out
12 4 1 4 10 4 8	5

3°

maketime.in	maketime.out
7 2 0 3 4	3

Εξήγηση: Στο 2° παράδειγμα, η Κατερίνα μπορεί να ακυρώσει την υποχρέωση της μέρας 8 και να κάνει 5 μέρες διακοπών (από την 5^η μέρα έως και την 9^η). Προσέξτε ότι έχει προγραμματίσει δύο υποχρεώσεις για την 4^η μέρα και δεν είναι διατεθειμένη να τις ακυρώσει και τις δύο ($K=1$). Στο 3° παράδειγμα, η Κατερίνα δεν είναι διατεθειμένη να ακυρώσει καμία υποχρέωση και μπορεί να κάνει 3 μέρες διακοπών, από την 5^η μέρα έως και την 7^η.

Περιορισμοί:

- Για περιπτώσεις ελέγχου συνολικής αξίας 20%, θα είναι:
 $1 \leq N \leq 100, 1 \leq M \leq 100, 0 \leq K \leq M$
- Για περιπτώσεις ελέγχου συνολικής αξίας 50%, θα είναι:
 $1 \leq N \leq 10.000, 1 \leq M \leq 20.000, 0 \leq K \leq M$
- Για περιπτώσεις ελέγχου συνολικής αξίας 100%, θα είναι:
 $1 \leq N \leq 1.000.000, 1 \leq M \leq 2.000.000, 0 \leq K \leq M$

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

Μέγιστος χρόνος εκτέλεσης: 1 sec.

Μέγιστη διαθέσιμη μνήμη: 64 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: maketime  
*/  
για κώδικα σε C  
  
/*  
USER: username  
LANG: C++  
TASK: maketime  
*/  
για κώδικα σε C++  
  
(*  
USER: username  
LANG: PASCAL  
TASK: maketime  
*)  
για κώδικα σε PASCAL  
  
/*  
USER: username  
LANG: Java  
TASK: maketime  
*/  
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. `starwars`). 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.

